RESEARCH ARTICLE

Spoken language identification based on the enhanced self-adjusting extreme learning machine approach

Musatafa Abbas Abbood Albadr[1]*, Sabrina Tiun[1], Fahad Taha AL-Dhief[2], Mahmoud A. M. Sammour[3]

1 CAIT, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia, **2** Faculty of Electrical Engineering, Department of Communication Engineering, Universiti Teknologi Malaysia, UTM Johor Bahru, Johor, Malaysia, **3** Faculty of Information and Communication Technology Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

* mustafa_abbas1988@yahoo.com

Abstract

Spoken Language Identification (LID) is the process of determining and classifying natural language from a given content and dataset. Typically, data must be processed to extract useful features to perform LID. The extracting features for LID, based on literature, is a mature process where the standard features for LID have already been developed using Mel-Frequency Cepstral Coefficients (MFCC), Shifted Delta Cepstral (SDC), the Gaussian Mixture Model (GMM) and ending with the i-vector based framework. However, the process of learning based on extract features remains to be improved (i.e. optimised) to capture all embedded knowledge on the extracted features. The Extreme Learning Machine (ELM) is an effective learning model used to perform classification and regression analysis and is extremely useful to train a single hidden layer neural network. Nevertheless, the learning process of this model is not entirely effective (i.e. optimised) due to the random selection of weights within the input hidden layer. In this study, the ELM is selected as a learning model for LID based on standard feature extraction. One of the optimisation approaches of ELM, the Self-Adjusting Extreme Learning Machine (SA-ELM) is selected as the benchmark and improved by altering the selection phase of the optimisation process. The selection process is performed incorporating both the Split-Ratio and K-Tournament methods, the improved SA-ELM is named Enhanced Self-Adjusting Extreme Learning Machine (ESA-ELM). The results are generated based on LID with the datasets created from eight different languages. The results of the study showed excellent superiority relating to the performance of the Enhanced Self-Adjusting Extreme Learning Machine LID (ESA-ELM LID) compared with the SA-ELM LID, with ESA-ELM LID achieving an accuracy of 96.25%, as compared to the accuracy of SA-ELM LID of only 95.00%.

Citation: Albadr MAA, Tiun S, AL-Dhief FT, Sammour MAM (2018) Spoken language identification based on the enhanced self-adjusting extreme learning machine approach. PLoS ONE 13 (4): e0194770. https://doi.org/10.1371/journal.pone.0194770

Editor: Xiangtao Li, Northeast Normal University, CHINA

Received: September 25, 2017

Accepted: March 11, 2018

Published: April 19, 2018

Data Availability Statement: The data has been uploaded to Figshare via the following URL: (https://doi.org/10.6084/m9.figshare.6015173.v1). In addition, the dataset is accessible at one of the author's official website (http://www.ftsm.ukm.my/sabrina/resources.html).

Funding: This project is funded by Malaysian government (to Mustafa Abbas Albadr) under research code FRGS/1/2016/ICT02/UKM/02/14.

Competing interests: The authors have declared that no competing interests exist in this paper.

1. Introduction

Language Identification (LID) is the process of determining and classifying a natural spoken language from given content and datasets [1, 2]. It is undertaken by performing computational linguistics approaches and applying many contexts. These contexts include; text categorisation of a written text [3] or speech recognition of a recorded utterance [4] of a spoken identified language. It is a challenging task because due to the variations in the type of speech input and understanding how humans process and interpret speech in adverse conditions [5].

When using a LID system, several types of information are considered. Furthermore, human understanding has inspired the classification of information, and several studies have applied methods which people have used to differentiate languages, whether consciously or not. A broad classification has been used to separate or split speech features into a low level and a high level.

At the low level, most commonly used features for LID are acoustics, phonetics, phonotactics and prosodic information while at the high level, LID can be established based on the morphology and sentence syntax [6].

The acoustic features usually modelled by MFCCs are the compact representation of the input speech signal fulfilling a compression of the data contained in the audio waveform.

The phonotactic features represent admissible sound patterns formed within a given language. The N-gram language model (LM) is used to model the phonotactic features. The prosodic features refer to the duration, pitch and stress of the speech and reflect elements such as the speaker's emotional state which cannot be characterised by the grammar used. The lexical features address the problems associated with the internal structure of words, and lastly, the syntactic features are the outcome of the analysis performed by the way in which words are linked or connected together to form phrases, clauses and sentences [6].

The conclusions, therefore, when comparing these two broad levels can be as follows. The low-level features are easier to obtain but are very volatile and are easily affected by noise and speaker variations, whereas high-level features contain more information regarding language discrimination. However, high-level features rely on large vocabulary recognisers, and as a result, more training data is needed which ultimately leads to a greater level of complexity in obtaining these features. Therefore, this study has used acoustic features and adopted the concept of feature extraction from [7] whereby the LID system is combined with a sequence of steps commencing from feature extraction, the Gaussian mixture model (GMM), i-vector construction, and recognition (classification), refer "Fig 1".

LID is an important pre-processing technique applied to future multi-lingual speech processing systems, such as audio and video information retrieval, automatic machine translation, multi-lingual speech recognition, intelligent surveillance and so forth. A major problem in LID is how to design a specific and effective language to represent speech utterances. It is challenging due to the significant variations introduced relating to different speech patterns, speakers, channels and background noise [8]. Due to technological advances, data is being generated at an ever-increasing pace, and the size and dimensionality of the data sets continue to grow each day. Therefore, it is important to develop efficient and effective machine learning methods that can be applied to analyze the data and to extract useful knowledge and insights from the information. More recently, Extreme Learning Machines (ELMs) have emerged and have been adopted as a popular framework for machine learning [9–11]. ELMs are a type of feed-forward neural network, characterized by random initialization of their hidden layer weights, combined with a fast training algorithm. The effectiveness (i.e. without blindness) of the random initialization and fast training makes it very appealing for large data analysis.

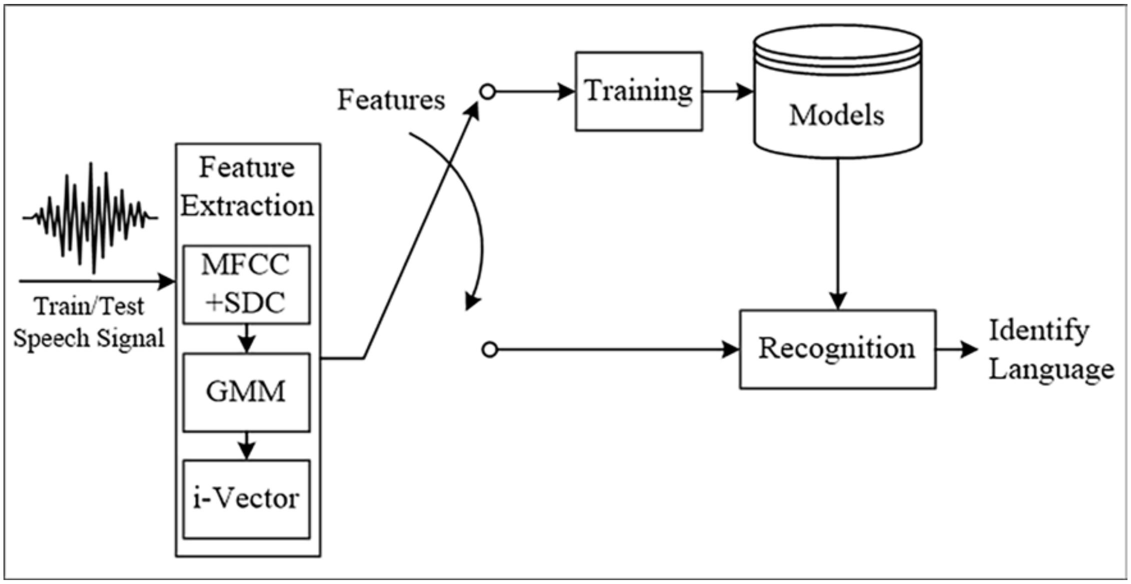

Fig 1. Steps of the language identification system, [6].

https://doi.org/10.1371/journal.pone.0194770.g001

The core classification unit is an important part of any LID system. The role of the classification unit is to map the audio sets and extract features from the i-vector system to enable its corresponding language to be identified. Different classifier types are defined in the literature such as the deep learning classifier, SVM, and ELM. ELM is described by [12], as a kind of feed-forward single hidden layer neural network, whose input weights, and thresholds of hidden layers are randomly generated. Because the output weights of the ELM are calculated utilizing the least-square method, the ELM exhibits high speed for training and testing purposes. However, the random input weights and thresholds of the hidden layers are not the best parameters, given that they cannot promise to achieve the ELM training goals and to meet global minimum requirements. The literature addresses the problem of optimizing the weights of the single-hidden layer feedforward neural networks (SLFN) trained by the ELM using various approaches. Researchers [13, 14] attempted to optimize the weights using meta-heuristic searching methods. Also, [15] aimed to optimize the weights of the ELM using the teaching phase and the learning phase under the ameliorated teaching learning-based optimisation framework. However, studies on the selection approach, to generate fresh solutions and to examine the impact on the performance of the search, are currently limited. This may lead to a slower convergence rate or incomplete optimisation. The purpose of this study is to improve the Extreme Learning Machine (ELM) algorithm by improving the self-adjusting approach and the implementation of Spoken Language Identification (LID). The final aim of the study is to prove the efficiency of the extreme learning machine as a classifier model for LID when improved optimisation is observed. The remainder of the study is organized into the following sections. Section 2 discusses related work; Section 3 describes the proposed method; Section 4 discusses and presents the experiments and results, and finally, Section 5 presents the conclusions and recommendations for future action.

2. Related work

The focus in this section is on machine learning and its applicability on LID as a learning model for classifying languages. The ELM is one type of classification algorithm proposed by

[12] as being an effective approach towards training Single Hidden Layer Neural Network (SLNN) in one iteration. The research conducted by Huang and his team published several improvements to the extreme learning machine such as an online extreme learning machine [16] and a kernel extreme learning machine [17]. This has been proven in a wide range of applications requiring learning; human action recognition [18], Cryptography [19], image segmentation [20], face classification [21, 22], intrusion detection in cloud computing [23], Graph embedding [24], and ELMs for both semi-supervised and unsupervised tasks based on the manifold regularization [25, 26].

During the past years, extreme learning machine (ELM) [27] has been becoming an increasingly significant research topic for machine learning and artificial intelligence, due to its unique characteristics, i.e., extremely fast training, good generalization, and universal approximation/classification capability. ELM is an effective solution for the single hidden layer feedforward networks (SLFNs), and has been demonstrated to have excellent learning accuracy/speed in various applications. Thus, ELM tends to achieve faster and better generalization performance than those of back propagation (BP)-based neural networks (NNs), and SVM [27–29].

One of the important factors motivating researchers to use the extreme learning machine is its superiority over classical support vector machines from several aspects [12]. Firstly, the extreme learning machine has greater capability to avoid overfitting. Secondly, it can function on both binary and multi-type of classifiers, and thirdly, it has a neural network structure and can function as being kernel based, like SVM. All these factors add increased recognition capabilities regarding the efficiency of ELM to achieve effective learning performance.

In the field of language identification, there have been several attempts at building an ELM based language classifier to replace the classical SVM. [30] developed a new variant of an extreme learning machine applied to language identification. The improved algorithm is known as the Regularized Minimum Class Variance Extreme Learning Machine (RMCVELM). The core concept of the algorithm is to minimize the empirical risk, structural risk, and the intra-class variance. The authors evaluated it from the perspective of the execution time and level of accuracy. It outperformed SVM on the execution time and comparable classification accuracy. It is important to point out, that despite the fact of the superiority of the developed classifier, the aspect relating to the optimisation of random weights of the ELM have been ignored, causing non-optimal classification performance.

Another study applying the extreme learning machine was in the field of speaker recognition by [31]. The study used ELM on a speaker with independent text data and comparing the results with SVM. The findings from this study identified that ELM is faster to execute with much higher accuracy, however, this work is not considered as a precise application given it focused on language identification. Furthermore, their model is a binary classification model, whereby the aim of this study is to investigate using ELM in language identification, being a multi-classification problem.

A further study was conducted by [32] to identify emotions of the speaker using DNN as a feature extractor and to use extreme learning machine as a classifier. The findings identify that Kernel ELM (KELM) and ELM combined with DNN achieve the highest accuracy compared to the other baseline approaches. The authors, however, ignore the fact that ELM or KELM needs to be optimised on the input hidden layer weights.

[33] used ELM to examine the problems associated with another classifier on a different type of audio-related classification. The emotion recognition studied was based on the audio of the speaker. The features of the GMM model are used as input to the classifier with the authors emphasizing the high capabilities of GMM based features in providing a discriminative factor for classifying emotions. Unfortunately, however, minimal investigation on the

effect of adding extra features to the classification or attempts to overcome the drawbacks of ELM was carried out.

3. Method

3.1 General overview

The general overview of the proposed method is illustrated in "Fig 2". The diagram shows the various blocks that will be used to create the LID system with optimised machine learning. The following sub-sections will discuss a separate area as shown in the LID system.

3.2 Feature extraction

The standard feature extraction for LID is adopted from [7]. Firstly, segmentation is performed to convert the input signal into frames of 25 ms with 10 ms overlap. Secondly, 7 Mel-Frequency Cepstral Coefficients (MFCCs), including C0, are obtained followed by applying Vocal Tract Length Normalization (VTLN). Next, cepstral mean and variance normalization is performed along with RASTA filtering, and this is then followed by calculating the Shifted Delta Cepstral (SDC) features in a 7-1-3-7 configuration. The results are 56-dimensional vectors consisting of both the MFCCs and the SDC. Also, GMM containing 2048 Gaussian

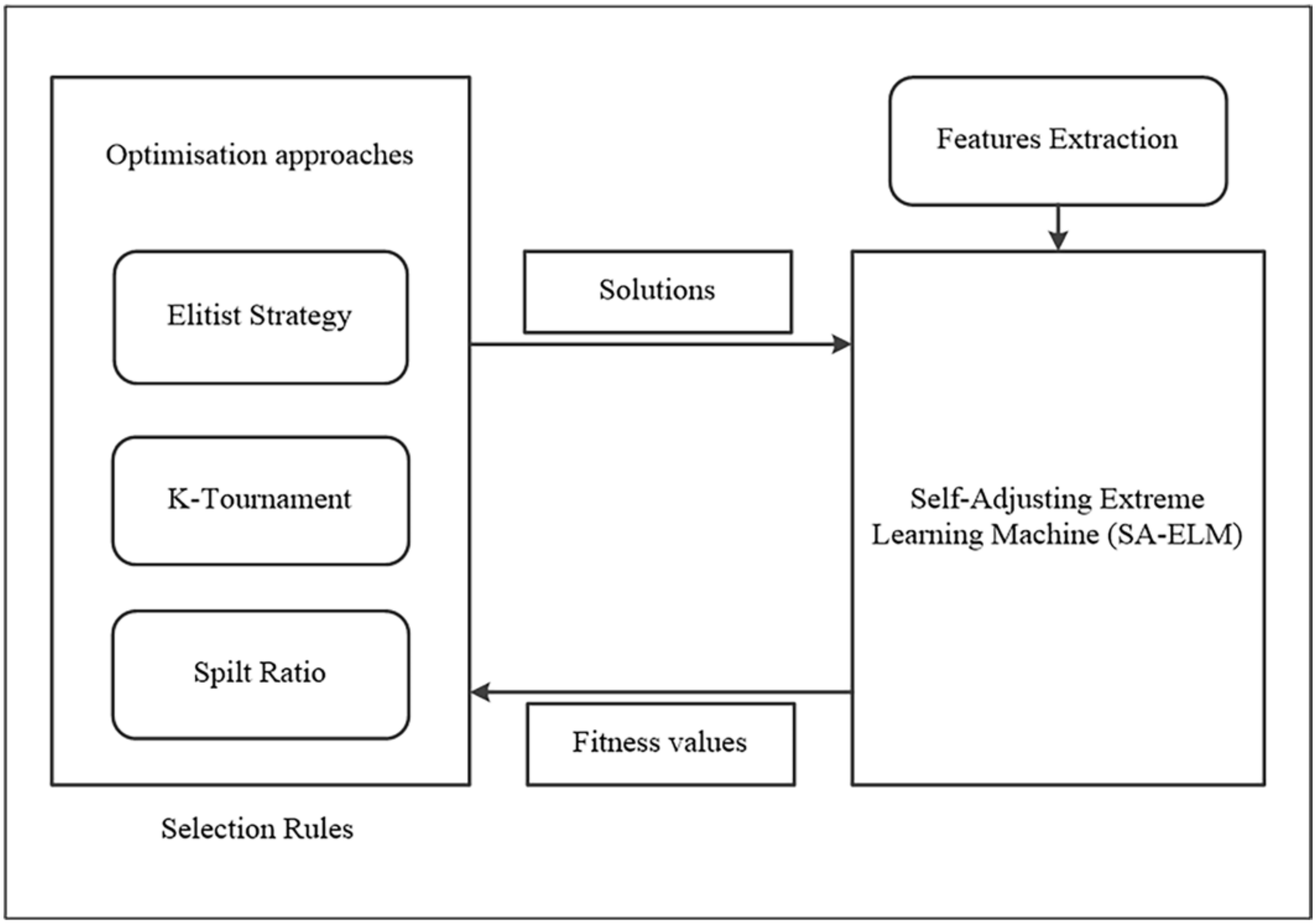

Fig 2. An illustrative block diagram of the optimised LID system.

https://doi.org/10.1371/journal.pone.0194770.g002

components with diagonal covariance matrices was used with the dimensionality of the i-vectors set to 600.

3.3 Basic extreme learning machine (ELM)

The original ELM algorithm for training SLFN is proposed by [12]. The main concepts or ideas behind ELM are the hidden layer weights, where the biases are generated randomly. The output weights are then calculated using the least-squares solution which is defined by the outputs of the hidden layer and targets. An overview of the ELM structure and the training algorithm is shown in "Fig 3". The next section which provides a brief description of the ELM.

Where

N = represents a set of distinct samples ($X_{i,}$ t_i), where X_i = [$x_{i1,}$ $x_{i2...}$ x_{in}] ^{T} ∈ R^n and t_i = [$t_{i1,}$ $t_{i2...}$ t_{im}] ^{T} ∈ R^m.

L = indicates to the hidden layer nodes.

g(x) = represents the activation function, which is a mathematical model as described and applied using Eq (1)

$$\sum_{i=1}^{L} \beta_i g_i(X_j) = \sum_{i=1}^{L} \beta_i g_i(W_i.X_j + b_i) = o_j \tag{1}$$

J = 1. . . N.

Where:

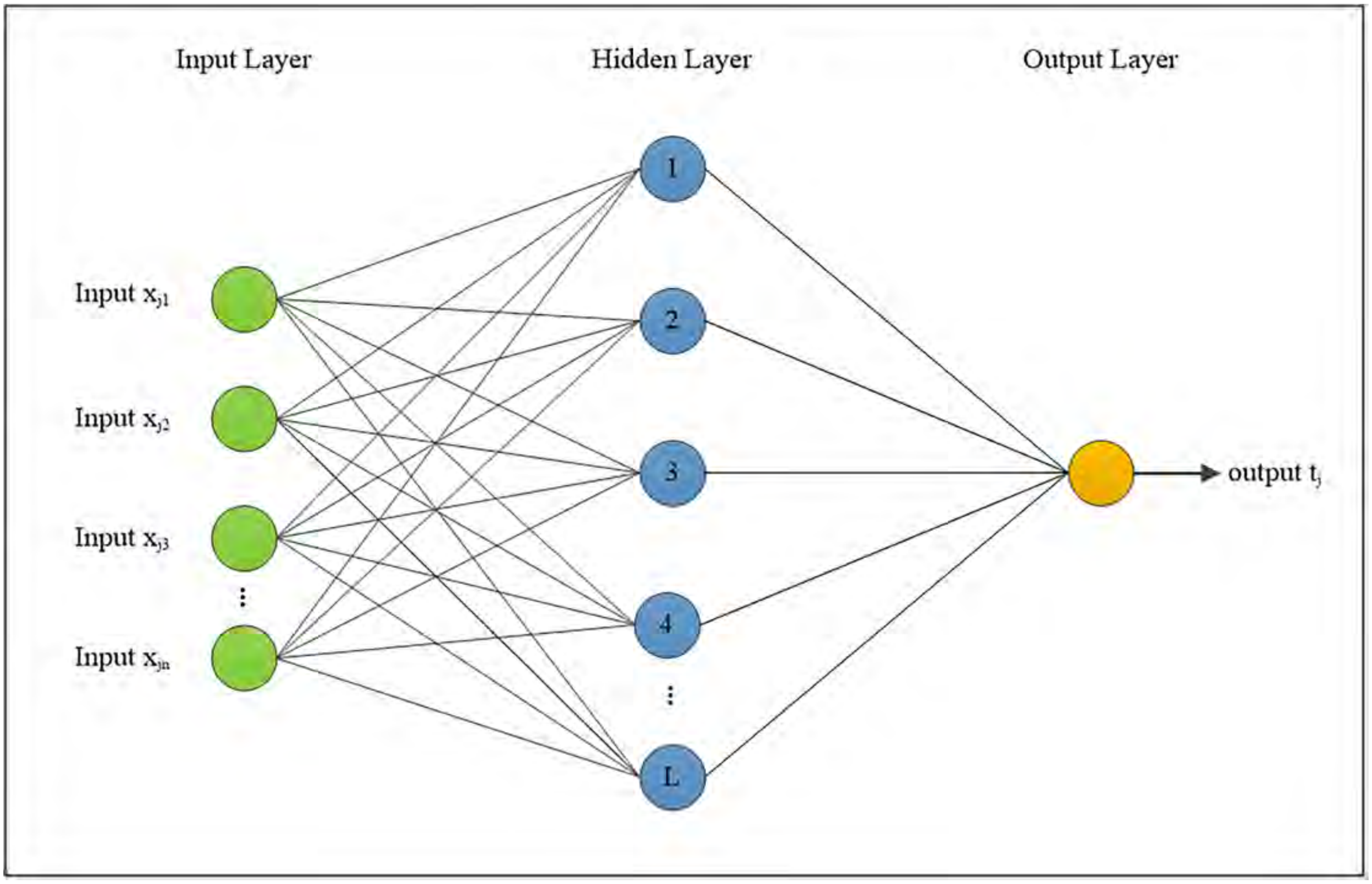

Fig 3. Diagram of the extreme learning machine [34].

https://doi.org/10.1371/journal.pone.0194770.g003

$W_i = [W_{i1}, W_{i2\ldots} W_{in}]^T$ is the weight vector that provides the connection between the *ith* input nodes and the hidden node.

$\beta_i = [\beta_{i1}, \beta_{i2}, \ldots\ldots, \beta_{im}]^{\mathrm{T}}$ = the weight vector that provides the connection between the *ith* output nodes and the hidden node.

b_i = the threshold of the *ith* hidden node.

$W_i.\ X_j$ = the inner product of W_i and X_j. However, the output nodes are chosen linearly.

L = hidden nodes, and the standard of SLFNs in the activation function *g(x)* could be the samples of *N* without error.

That is:

$\sum_{j=1}^{L} \|o_j - t_j\| = 0$, i.e., there exist β_i, W_i, and b_i such that in Eq (2):

$$\sum_{i=1}^{L} \beta_i g_i (W_i.X_j + b_i) = t_j, \ \ j = 1, \ldots., N. \tag{2}$$

From the above equations for *N*, this can be written as follows:

$$H\beta = T \tag{3}$$

Where:

$$H(\ W_1 \quad \ldots \quad W_{L,} \quad b_1 \quad \ldots \quad b_{L,} \quad X_1 \quad \ldots \quad X_N\ )$$

$$= \begin{bmatrix} g(W_1.X_1 + b_1) & \cdots & g(W_L.X_1 + b_L) \\ \vdots & \ldots & \vdots \\ g(W_1.X_N + b_1) & \cdots & g(W_L.X_N + b_L) \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \beta_L^T \end{bmatrix}_{L*m} \quad \text{and} \quad T = \begin{bmatrix} t_1^T \\ t_N^T \end{bmatrix}_{N*m}$$

The authors in Huang et al. (2006) named the variables, where *H* refers to the output matrix of the hidden layer in the neural network; in *H* the *ith* column refers to the *ith* hidden layer nodes on the input nodes. If the desired number of the hidden nodes is $L \leq N$, this therefore means the activation function *g* is infinitely differentiable. Eq (3) then turns into a linear system. Furthermore, the output weights β can be determined analytically by discovering a least square solution in the following way:

$$\beta = H^{\dagger}\mathrm{T}$$

Where $H^{\dagger}$ is represents the Moore–Penrose generalised inverse for H. Thus, the output weights are calculated via a mathematical transformation. This makes sure that the lengthy training phrase when network parameters are iteratively adjusted with some suitable learning parameters (like iterations and learning rate) is done away with.

The authors in [12] named the variables, where *H* refers to the output matrix of the hidden layer in the neural network; in *H* the *ith* column refers to the *ith* hidden layer nodes on the input nodes. If the desired number of the hidden nodes is $L \leq N$, this therefore means the activation function *g* is infinitely differentiable.

The weakness of ELM is that it should have a particular approach for determining the weights of the input-hidden layer weights and therefore, is subject to local minima. In other words, based on given training data, there is no way to assure that the trained ELM is the most appropriate in performing the classification. To resolve the weakness, an optimised approach must be integrated with the ELM to identify the optimal weights that assure the best performance of ELM. In the next subsection, ATLBO is presented and adopted as an optimisation approach for this very purpose.

3.4 Ameliorated teaching-learning-based optimisation (ATLBO)

Teaching Learning Based Optimisation (TLBO) is one of many optimisation approaches proposed by Rao et al. The algorithm has attracted many researchers' due to its simple structure, fewer parameters and high execution speed. After developing TLBO, [35], further improvement of the algorithm was made to execute faster and to avoid selfish behaviour and presented this improvement in ATLBO.

The set of equations of ATLBO can be divided into two phases; the 'Teaching' phase, and the 'Learning' phase. The 'Teaching' phase means learning from the teacher, while the 'Learning' phase means learning through the interaction between learners. In the teaching phase, each solution is updated based on Eqs (4–6):

$$X_{new,i} = \omega_i X_{old,i} + \emptyset_i (M_{new} - T_F M_i) \tag{4}$$

$$\omega_i = 1/(1 + \exp(-fit(i)/ap)^{iter}) \tag{5}$$

$$\emptyset_i = 1/(1 + \exp(-fit(i)/ap) \times iter) \tag{6}$$

Let *Mi* = the mean, and *Ti* = Teacher (best learner) at any iteration *i*. *Ti* will try to move the mean *Mi* towards its own level, so now the new mean will be T_i and designated as M_{new}. The solution is updated according to the difference between the existing and the new mean as depicted in Eq (4).

Where

ω_i = the inertia weight, which controls the effect of the former solution.

$\emptyset_i$ = the acceleration coefficient, which defines the maximum step size.

T_F = the teaching factor that decides the value of the mean to be changed, the value of T_F can be either 1 or 2.

fit(*i*) = the fitness of the *ith* learner.

ap = the maximum fitness in the first iteration.

iter = the current iteration.

While in the learning phase each solution is updated using Eqs (7–9).

$$X_{new,i} = \begin{cases} X_{old,i} + \varphi_i (X_j - X_i) & if\ f(X_i) \le f(X_j) \\ X_{old,i} + \psi_i (X_{best} - X_i) & if\ f(X_i) > f(X_j) \end{cases} \tag{7}$$

$$\varphi_i = 1 - \exp(fit(X_j) - fit(X_i)) \tag{8}$$

$$\psi_i = 1 - \exp(fit(X_{best}) - fit(X_i)) \tag{9}$$

where

X_{best} = the best learner in a class.

φ_i and ψ_i = the acceleration coefficients that decide the step size depending on the differences between two learners.

3.5 Self-adjusting extreme learning machine (SA-ELM)

[15] proposed SA-ELM using the concept of an of Teaching-Learning-Based Optimisation algorithm (TLBO) consisting of two phases for adjusting the input weight and bias of hidden

nodes. The first phase being the 'teaching phase' and the second phase being the 'learning phase'.

The SA-ELM is described in detail as follows. The values of the input weights and thresholds of the hidden nodes are defined randomly in the teaching phase of SA-ELM, and learners' indicating the marks of all course types as shown below.

$$\mathrm{Vis}\theta = \{\, w_{11,} \quad w_{12,} \quad \ldots \quad w_{1n,} \quad w_{21,} \quad w_{22,} \quad \ldots \quad w_{2n,} \quad w_{m1,} \quad w_{m2,} \quad \ldots \quad w_{mn,} \quad b_{1,} \quad \ldots \quad b_m \}.$$

where,

W_{ij} is the weight's value connecting between the *jth* input node and the *ith* hidden node, $W_{ij} \in [-1, 1]$;

b_i is the bias of the *ith* hidden node, $b_i \in [0, 1]$;

n is the number of input nodes; and

m is the number of hidden nodes.

$(n + 1) \times m$ represents the dimension of the learners' mark, which means the $(n + 1) \times m$ parameters need to be optimised. Therefore, the fitness function in the SA-ELM is set using the following Equation

$$f(\theta) = \sqrt{\frac{\sum_j^N \| \sum_k^m \rho_k g(w_k x_j + b_k) - y_j \|_2^2}{N}} \tag{10}$$

where,

ρ is the output weight matrix;

y_j is the true value; and

N is the number of training samples.

The initial or first step calculates each target function fitness value. Following this, the learner having the minimum fitness value is selected as a teacher. The learner's new mark fundamentally relied on the previous mark $\theta_{old,i}$ and the difference between the former mark and the teacher ($\theta_{best} - \theta_{old,i}$). The mechanism to update the structure of the parameters in the SA-ELM are calculated using the following Equations:

$$\theta_{new,i} = \omega_i \theta_{old,i} + \emptyset_i(\theta_{best} - \theta_{old,i}) \tag{11}$$

$$\omega_i = 1/(1 + \exp(-f(i)/a)^{iter}) \tag{12}$$

$$\emptyset_i = 1/(1 + \exp(-f(i)/a) \times iter) \tag{13}$$

where,

ω_i is the inertia weight, which controls the effect of the former mark.

$\emptyset_i$ represents the acceleration coefficient, which defines the maximum step size.

In Eqs (12) and (13), 'a' represents the maximum target function fitness value in the first iteration, and *iter* represents the present iteration.

Through communicating with each other, the learners increased their marks in the SA-ELM 'learning phase'. In this step, the structure of the updated parameters used the Elitist strategy. The following Equations are used to calculate the update in the ith learner's marks, in the ith iteration.

$$\theta_{new,i} = \begin{cases} \theta_{old,i} + \alpha_i(\theta_j - \theta_i) & if \ f(\theta_i) \leq f(\theta_j) \\ \theta_{old,i} + \beta_i(\theta_{best} - \theta_i) & if \ f(\theta_i) > f(\theta_j) \end{cases} \quad (14)$$

$$\alpha_i = 1 - \exp(f(\theta_j) - f(\theta_i)) \quad (15)$$

$$\beta_i = 1 - \exp(f(\theta_{best}) - f(\theta_i)) \quad (16)$$

where,

θ_{best} in Eq (14), represents the best learner; α_i and β_i are acceleration coefficients which decide the step size depending on the differences between two learners.

3.6 Optimization approach

This section provides an explanation of the optimisation approach of the LID learning model. As previously mentioned, the ELM requires optimisation of the input hidden layer weights. The baseline approach adopts ATLBO for performing the optimisation. However, ATLBO uses only one criterion for selection. Therefore, an enhanced ATLBO or EATLBO will seek to optimize the ATLBO which is discussed further in the next sub-section along with the ESA-ELM which is based on EATLBO.

3.6.1 Enhanced ATLBO (EATLBO). The SA-ELM benchmark is based on ATLBO optimisation. The ATLBO process is divided into two parts. The first part consists of the 'Teacher Phase' and the second part consists of the 'Learner Phase'. The 'Teacher Phase' is best described as, learning from the teacher and the 'Learner Phase' described as learning through the interaction between the learners. A good teacher is one who brings his or her learners up to his or her level regarding knowledge. But in practice, this is not always possible, and the teacher can only move the mean or average of a class up to some extent depending on the capability of the class. This follows a random process depending on many factors. In the 'Learner Phase', the Learners can increase their knowledge using two different methods. The first method is through obtaining input from the teacher, and the second method is through the interaction between them. A learner interacts randomly with other learners assisted through group discussions, presentations, formal communications, etc. A learner can learn something new if the other learner whom they are interacting with, has greater knowledge. ATLBO is based on the Elitist strategy criterion to select the best solutions in each iteration, but, this approach suffered from two problems. The first problem is that if the best solution falls into some local optima, then all other solutions will be driven towards the wrong solution and the algorithm will provide the incorrect answer. Secondly, since all solutions will follow the best solution, if there is a better solution than the one found, it may not be possible to discover. Therefore, the enhancement of ATLBO in this study, two additional criteria are incorporated, Split Ratio and K-Tournament method.

The purpose of using the k-Tournament method is to choose several solutions randomly, followed by selecting from the selected solutions the most appropriate (or best) solutions to transfer to the following generation. The Split ratio method determines how many of the identified best solutions will be transferred to the next generation, and then, from the remaining solutions, randomly selecting solutions to transfer to the next generation. Through applying this method, the search space is expanded, and the right answer is more likely to be found.

K-random samples are selected from the population to illustrate how K-Tournament works. The best solution is then selected from among the random tournament. Next, the k-

Tournament is repeated until the required number of solutions is reached and then moved to the next generation. Similarly, the split ratio is applied based on a 25% - 75% ratio. This means that the algorithm will select the best 25 solutions in a deterministic manner, and then moved to the next generation while the next 75% are randomly chosen from the entire population.

3.6.2 Enhanced self-adjusting extreme learning machine (ESA-ELM). The ESA-ELM is recommended based on the concept of the Enhanced Teaching-Learning-Based Optimisation algorithm (called EATLBO). This uses the Split Ratio instead of the Elitist strategy, whose input weight values and the bias of hidden nodes are adjusted via the teaching phase and learning phase of the EATLBO. The ESA-ELM is described along with the notation of the ESA-ELM and presented in Table 1.

The values of the input weights and thresholds of the hidden nodes are defined randomly in the teaching phase of the ESA-ELM and represented as learners' marks for all courses types,

$$\mathrm{Vis}X = \{\, w_{11,} \quad w_{12,} \quad \ldots \quad w_{1n,} \quad w_{21,} \quad w_{22,} \quad \ldots \quad w_{2n,} \quad w_{m1,} \quad w_{m2,} \quad \ldots \quad w_{mn,} \quad b_{1,} \quad \ldots \quad b_{m} \}.$$

where:

W_{ij} is the weight's value connecting between the *jth* input node and the *ith* hidden node, $W_{ij} \in [-1, 1]$;

$\mathrm{b_i}$ is the bias of the *ith* hidden node, $\mathrm{b_i} \in [0, 1]$;

n is the number of input nodes; and

m is the number of hidden nodes.

$(n + 1) \times m$ represents the dimension of learners' mark, which therefore requires the $(n +1) \times m$ parameters to be optimised. Therefore, the fitness function in the ESA-ELM set is calculated using the following Equation

$$f(X) = \sqrt{\frac{\sum_j^N || \sum_k^m \rho_k g(w_k x_j + b_k) - y_j ||_2^2}{N}} \tag{17}$$

where,

ρ is the output weight matrix;

$\mathrm{y_j}$ is the true value; and

N is the number of training samples.

In the first step, the target function fitness value is calculated. Then, the learner having the minimum or lowest fitness value is chosen as a teacher. The learner's new mark fundamentally relied on the previous mark $\mathrm{X_{old,i}}$ and the differences between the former mark and the teacher

Table 1. Notation table for ESA-ELM.

Notations	Implications
$\mathbf{X}$	Input-weight and bias assemble
ρ	The output weight matrix
$\boldsymbol{X_{old,i}}$	The previous ith solution
$\boldsymbol{X_{new,i}}$	The new ith solution
$\boldsymbol{X_{best}}$	The best solution
$\boldsymbol{\omega_i}$	The inertia weight
$\emptyset_{\boldsymbol{i}}$	The acceleration coefficient
iter	The current iteration
a	The maximum fitness value in the first iteration
$\boldsymbol{\alpha_i}$	The acceleration coefficients
$\boldsymbol{\beta_i}$	The acceleration coefficients

https://doi.org/10.1371/journal.pone.0194770.t001

(X_{best} – $X_{old,i}$). The mechanism to update the structure of the parameters in the ESA-ELM is calculated using the following Equations

$$X_{new,i} = \omega_i X_{old,i} + \emptyset_i (X_{best} - X_{old,i}) \tag{18}$$

$$\omega_i = 1/(1 + \exp(-f(i)/a)^{iter}) \tag{19}$$

$$\emptyset_i = 1/(1 + \exp(-f(i)/a) \times iter) \tag{20}$$

where,

ω_i is the inertia weight, which controls the effect of the former mark; and

$\emptyset_i$ representing the acceleration coefficient, defining the maximum step size.

In Eqs (19) and (20), 'a' represents the maximum target function fitness value in the first iteration, and *iter* represents the present iteration.

Through communicating with others, the learners improved their marks in the 'Learning Phase' of ESA-ELM. In this step, the mechanism to update the structure of the parameters, adopted the Split Ratio method to calculate the ith learner's marks in the ith iteration, as shown in the following Equations

$$X_{new,i} = \begin{cases} X_{old,i} + \beta_i (X_j - X_i) & if\ f(X_i) \le f(X_j) \\ X_{old,i} + \alpha_i (X_{best} - X_i) & if\ f(X_i) > f(X_j) \end{cases} \tag{21}$$

$$\alpha_i = 1 - \exp(f(X_j) - f(X_i)) \tag{22}$$

$$\beta_i = 1 - \exp(f(X_{best}) - f(X_i)) \tag{23}$$

Where, Eq (21), X_{best} represents the best learner; α_i and β_i are the acceleration coefficients which decide the step size depending on the differences between two learners. The learning algorithm of the ESA-ELM performed using the following steps:

- Step (1): Generate the input weights and the bias of the hidden layer (i.e. a number of students) randomly which sets the population number and target function.
- Step (2): 'Teaching phase', calculates the fitness value, thereby updating the structure parameters applying Eq (18).
- Step (3): 'Learning phase', adopts the Split Ratio method to update the parameters using Eq (21).

According to explanations noted above, regarding the ESA-ELM, this can be described further with the aid of a flowchart illustrating the ESA-ELM algorithm and steps. "Fig 4". represents the flowchart of the ESA-ELM algorithm.

4. Experiments and results

4.1 Raw dataset preparation

Eight different spoken languages were selected and tested for recognition purposes. The languages were; 1) Arabic, 2) English, 3) Malay, 4) French, 5) Spanish, 6) German, 7) Persian, and 8) Urdu with audio files recorded from broadcasting media channels in those respective countries. The following media broadcasting channels were:

1. Arabic: Syrian broadcast TV;

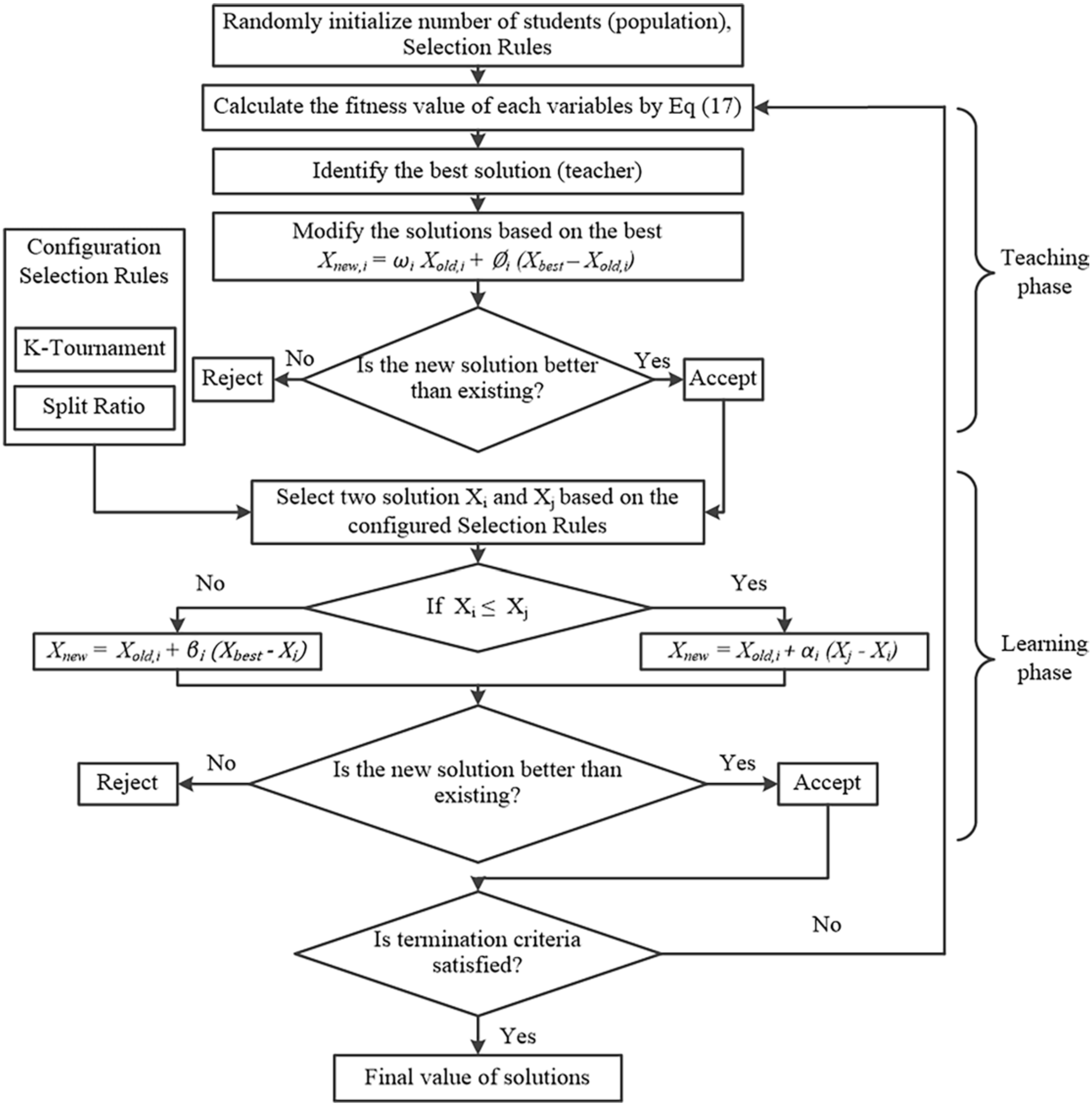
Randomly initialize number of students (population), Selection Rules
Calculate the fitness value of each variables by Eq (17)
Identify the best solution (teacher)
Modify the solutions based on the best
$X_{new,i} = \omega_i X_{old,i} + \emptyset_i (X_{best} - X_{old,i})$
Configuration Selection Rules
K-Tournament
Split Ratio
Reject
No
Is the new solution better than existing?
Yes
Accept
Teaching phase
Select two solution X_i and X_j based on the configured Selection Rules
No
If $X_i \leq X_j$
Yes
$X_{new} = X_{old,i} + \beta_i (X_{best} - X_i)$
$X_{new} = X_{old,i} + \alpha_i (X_j - X_i)$
Learning phase
Reject
No
Is the new solution better than existing?
Yes
Accept
Is termination criteria satisfied?
No
Yes
Final value of solutions

Fig 4. Flowchart illustrating the ESA-ELM algorithm.

https://doi.org/10.1371/journal.pone.0194770.g004

2. English (British): British Broadcasting Corporation (BBC);
3. Malay: TV9, TV2, TV3;
4. French: TF1 HD;

Table 2. Dataset dimension.

Number of records	Number of classes	Number of features
120	8	600

https://doi.org/10.1371/journal.pone.0194770.t002

5. Spanish: Real Madrid TV HD, La1, La2;
6. German: Zweites Deutsches Fernsehen (ZTV);
7. Persian: Islamic Republic of Iran News Network (IRINN); and
8. Urdu: GEO Kahani.

Each language consisted of 15 utterances, with the duration of each utterance recorded being 30 Sec. 67% of the datasets were used for training, and 33% of the datasets were used for testing purposes. The audio files were recorded from respective channels as mentioned, with each dataset representing a different language to test the robustness of the algorithm.

All utterances were recorded using an mp3 format with a dual channel, using MATLAB as an array consisting of two similar columns although, only one column was used. The utterance term was the equivalent to one vector of the sampled data from the audio file. Each utterance was 30 seconds in length and required to be sampled and quantised:

1. Sampling rate: (44100 Hz), the largest frequency was (22050 Hz) referencing the Nyquist frequency. The 30 seconds' length was approximately (30 * 44100 = 1323000).
2. Quantisation: representing real-valued numbers as integers using a 16-bit range (with values from -32768 to 32767).

The dataset that has used is described in the following below:

a. Dataset name (with extension): iVectors.mat.

b. Dataset dimensions as presented in Table 2:

c. Class description as provided in Table 3:

d. Features description as depicted in Table 4:

e. Class-label-column number: Last column (601)

4.2 Evaluation scenario

This section discusses the evaluation measures of the EATLBO and ESA-ELM. Firstly, the EATLBO was compared with the original ATLBO for several standard mathematical functions

Table 3. Class description.

Number	Meaning	Number of records
1	Arabic	15
2	English	15
3	Malay	15
4	French	15
5	Spanish	15
6	German	15
7	Persian	15
8	Urdu	15

https://doi.org/10.1371/journal.pone.0194770.t003

Table 4. Features description.

Number	Name	Type
1 → 600	i-vector values	Single

https://doi.org/10.1371/journal.pone.0194770.t004

relating to the optimisation surface. Secondly, the ESA-ELM was evaluated on several different parameters of the learning model.

4.2.1 Evaluation of common mathematical functions. Five experiments applying five different objective functions were conducted for ATLBO and the EATLBO (k-Tournament and Split Ratio), with the number of iterations equivalent to 1000. The purpose of using five different objective functions was to evaluate the performance of choosing the optimal (i.e. best) fitness value for the ATLBO and the EATLBO (k-Tournament and Split Ratio) in all iterations. Table 5, represents the fitness values obtained from the ATLBO and the EATLBO (k-Tournament and Split Ratio).

Comparing EATLBO and ATLBO it can conclude that the former has outperformed the latter. However EATLBO in this comparison is based on K-Tournament which might not be the best. Thus another selection criteria will be investigate. Therefore, another method called the Split Ratio method was also used. The results as shown in Table 1, illustrate the EATLBO (split Ratio) providing a fitness value closer to the optimal value, meaning that the performance of the EATLBO (split Ratio) was better compared to both the EATLBO (K-Tournament) and the ATLBO.

4.2.2 Evaluation on different learning model parameters. Several classification experiments were conducted on the formulated datasets with both the SA-ELM benchmark and the ESA-ELM (Split Ratio) method, varying the number of hidden neurones in the range [650–900] with an increment or step of 25. Therefore, the number of all experiments for the SA-ELM benchmark was 11, and similar for ESA-ELM (Split Ratio) and the number of iterations for each test was equal to 500 iterations. The Split ratio method was selected to generate the remaining results due to its advantages over using the K-tournament method.

The evaluation performed in this study is based on [36] which presents different measures applied for the evaluation. This article was selected because it addresses the problem of classifier evaluation, and provides effective measures. Supervised Machine Learning (SML) has several ways to evaluate the performance of learning algorithms and produced classifiers. Measures relating to the quality of the classification are created from a confusion matrix which records recognised examples for each class based on their correction rate.

In this study, several evaluation measures were used to evaluate the SA-ELM (benchmark) and the ESA-ELM (split ratio) based on the ground truth. Furthermore, the evaluation measures have been adopted to compare the benchmark with the ESA-ELM (split ratio) regarding true positive, true negative, false positive, false negative, accuracy, precision, recall, F-measure

Table 5. Testing results of optimizing benchmark mathematical functions.

Function number	Function Name	Number of variables	EATLBO (Split Ratio) fitness	EATLBO (K-Tournament) fitness	ATLBO fitness	Optimum value
1	Ackley’s	10	0	0	0.3445	0
2	Alpine #2	10	-26454	-370.3588	-119.7927	-30491
3	Styblinski Tang	10	-320.8240	-262.8895	-236.5990	-391.6620
4	Egg-Holder	2	-838.5126	-759.1134	-759.1971	-959.6407
5	Deb’s No.01	10	-1	-0.8806	-0.8180	-1

https://doi.org/10.1371/journal.pone.0194770.t005

and G-mean. The evaluation measures used in this study are depicted in Eqs (24–28)

$$\text{accuracy} = \frac{tp + tn}{tp + tn + fn + fp} \tag{24}$$

$$\text{precision} = \frac{tp}{tp + fp} \tag{25}$$

$$\text{recall} = \frac{tp}{tp + fn} \tag{26}$$

$$\text{F} - \text{Measure} = \frac{(2 \times precision \times recall)}{(precision + recall)} \tag{27}$$

$$\text{G} - \text{Mean} = \sqrt[2]{\frac{tp}{p} \times \frac{tn}{n}} \tag{28}$$

where:

tp = true positive, *tn* = true negative, *fp* = false positive, and *fn* = false negative.

The following figures demonstrate the results between the SA-ELM and the ESA-ELM (Split Ratio) for all experiments conducted. The accuracy of the ESA-ELM in the range [650–900] of hidden neurones was higher than the SA-ELM benchmark. This means that the ESA-ELM performance results are much better than the SA-ELM benchmark in all iterations. "Figs 5–9" illustrate the comparative results between the SA-ELM benchmark and ESA-ELM regarding accuracy, precision, recall, F-measure and G-mean. An important observation here

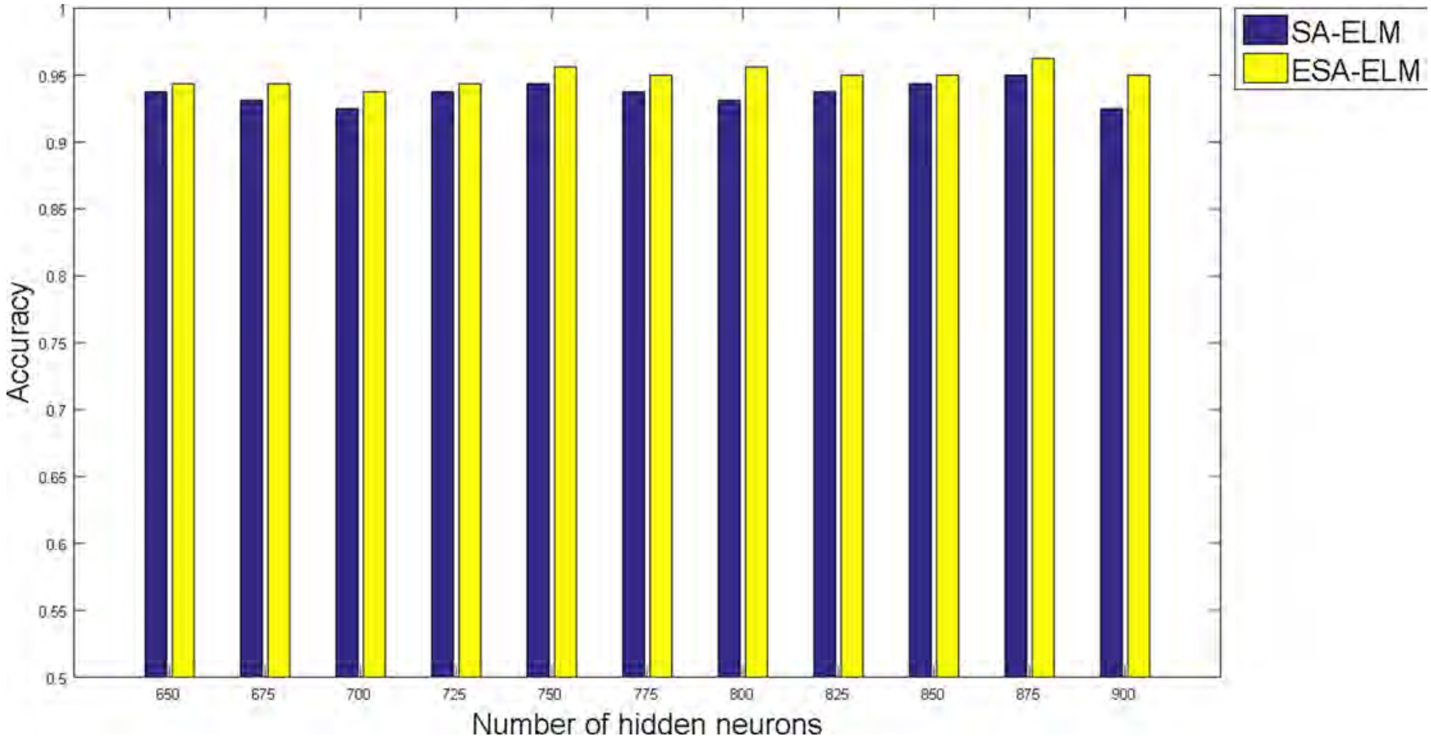

Fig 5. Accuracy measurement of the ESA-ELM and the SA-ELM benchmark.

https://doi.org/10.1371/journal.pone.0194770.g005

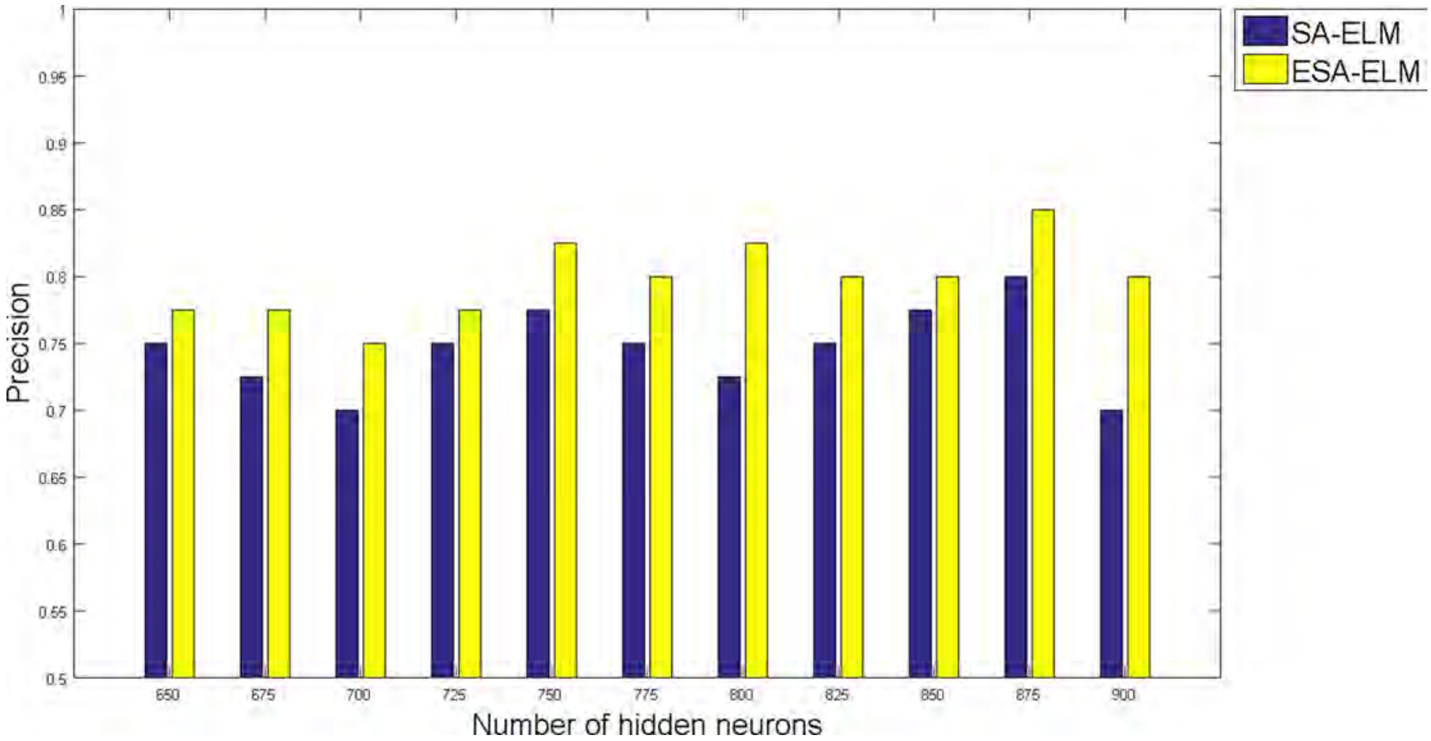

Fig 6. Precision measurement of the ESA-ELM and the SA-ELM benchmark.

https://doi.org/10.1371/journal.pone.0194770.g006

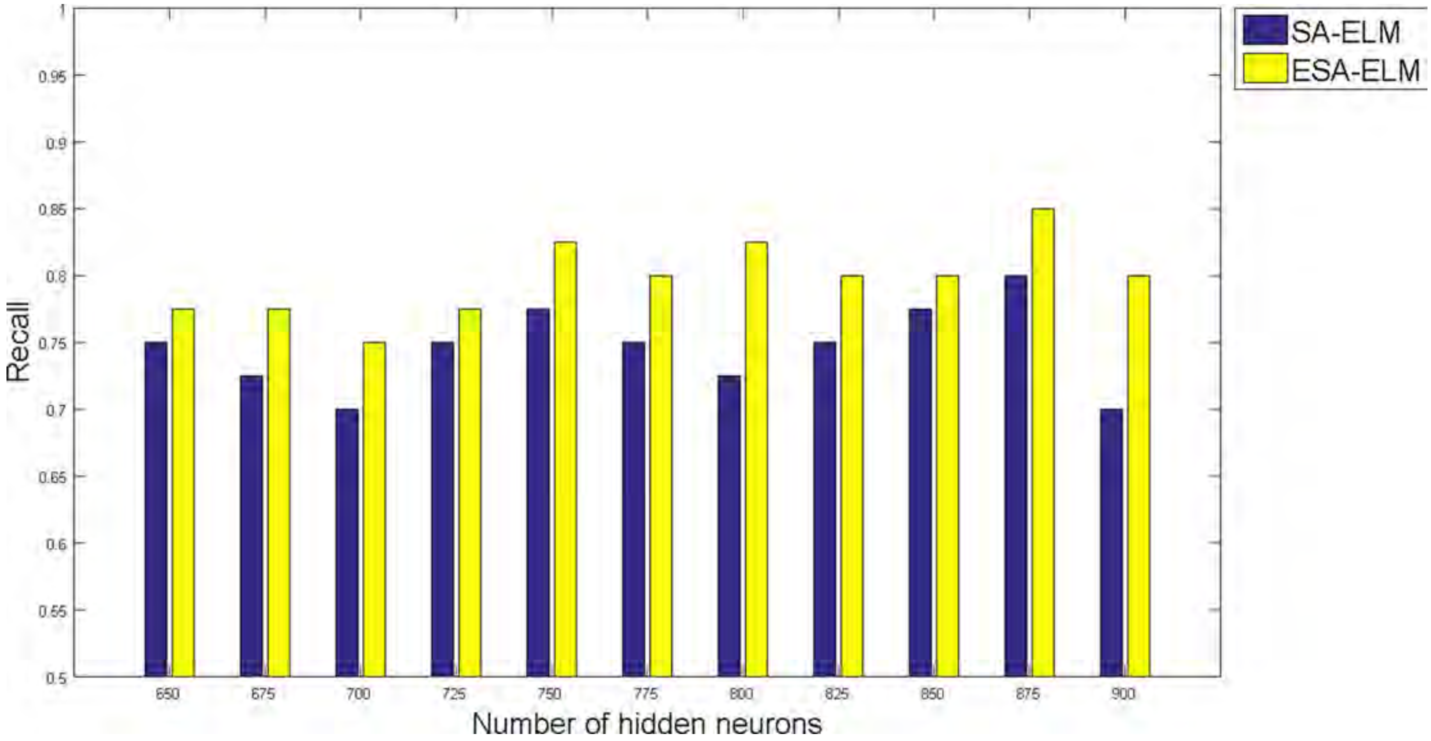

Fig 7. Recall measurement of the ESA-ELM and the SA-ELM benchmark.

https://doi.org/10.1371/journal.pone.0194770.g007

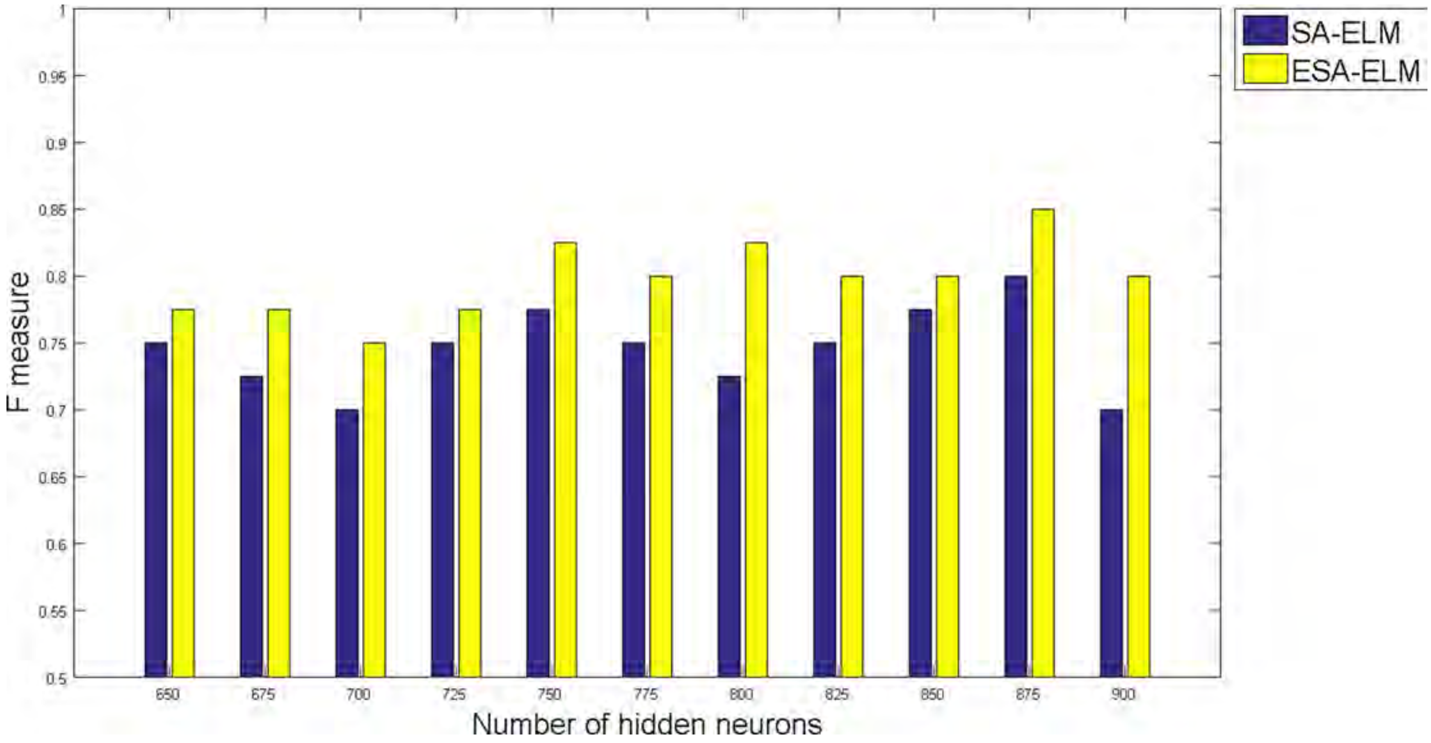

Fig 8. F-measure measurement of the ESA-ELM and the SA-ELM benchmark.

https://doi.org/10.1371/journal.pone.0194770.g008

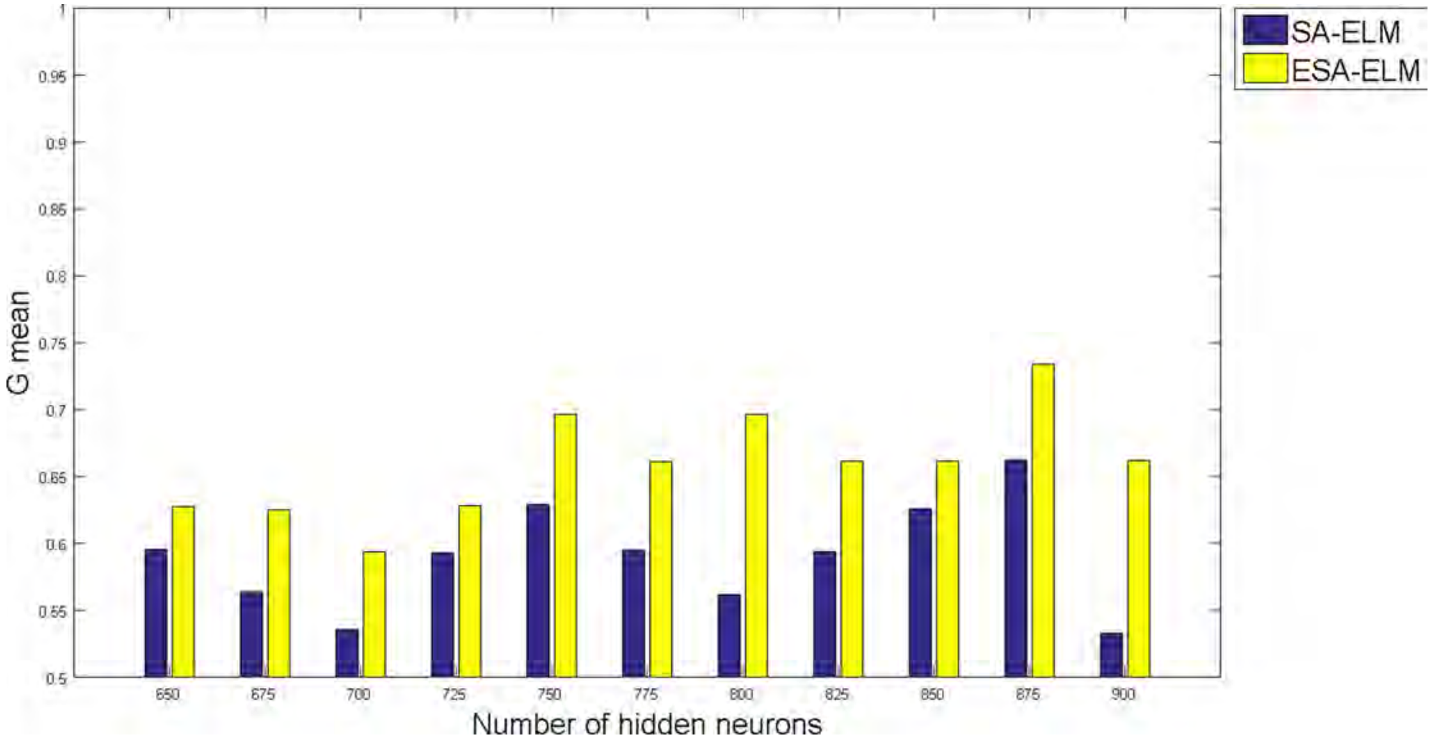

Fig 9. G-mean measurement of the ESA-ELM and the SA-LEM benchmark.

https://doi.org/10.1371/journal.pone.0194770.g009

Table 6. SA-ELM Evaluation measures through all the experiments.

	Accuracy	Precision	Recall	F-measure	G-mean
SA-ELM 650 hidden neurons	93.75	75.00	75.00	75.00	59.54
SA-ELM 675 hidden neurons	93.13	72.50	72.50	72.50	56.37
SA-ELM 700 hidden neurons	92.50	70.00	70.00	70.00	53.56
SA-ELM 725 hidden neurons	93.75	75.00	75.00	75.00	59.32
SA-ELM 750 hidden neurons	94.37	77.50	77.50	77.50	62.90
SA-ELM 775 hidden neurons	93.75	75.00	75.00	75.00	59.50
SA-ELM 800 hidden neurons	93.13	72.50	72.50	72.50	56.16
SA-ELM 825 hidden neurons	93.75	75.00	75.00	75.00	59.37
SA-ELM 850 hidden neurons	94.37	77.50	77.50	77.50	62.58
SA-ELM 875 hidden neurons	95.00	80.00	80.00	80.00	66.25
SA-ELM 900 hidden neurons	92.50	70.00	70.00	70.00	53.29

https://doi.org/10.1371/journal.pone.0194770.t006

is that the highest accuracy was achieved for 875 neurones, refer "Fig 5". The achieved accuracy was 96.25% for ESA-ELM and slightly lower, 95.00% for SA-ELM. Also, the obtained results of other measures for SA-ELM were; Recall 80.00%, Precision 80.00%, F-measure 80.00%, and G-mean 66.25%. While for ESA-ELM the results were; Recall 85.00%, Precision 85.00%, F-measure 85.00% and G-mean 73.41%. Tables 6 and 7 provides all the results of the Evaluation Measures through all the experiments for the SA-ELM and ESA-ELM as the following below:

As mentioned above, the highest accuracy have achieved with 875 hidden neurons therefore, "Figs 10–14" show the comparative results between the SA-ELM benchmark and ESA-ELM regarding accuracy, precision, recall, F-measure and, G-mean for each language separately with 875 hidden neurons.

Moreover, "Figs 15–19" illustrate the comparative results between the ESA-ELM and additional approach under name Elitist Genetic Algorithm Based ELM (EGA-ELM) regarding accuracy, precision, recall, F-measure, and G-mean.

5. Conclusion

This study enhances the existing learning model based on the ELM named as SA-ELM. The context regarding the development was to improve LID accuracy. The improvement of SA-ELM was based the optimisation approach, namely, ATLBO. ATLBO was enhanced through incorporating additional selection criteria for the searching process. The improvement was

Table 7. ESA-ELM evaluation measures through all the experiments.

	Accuracy	Precision	Recall	F-measure	G-mean
ESA-ELM 650 hidden neurons	94.37	77.50	77.50	77.50	62.76
ESA-ELM 675 hidden neurons	94.37	77.50	77.50	77.50	62.50
ESA-ELM 700 hidden neurons	93.75	75.00	75.00	75.00	59.38
ESA-ELM 725 hidden neurons	94.37	77.50	77.50	77.50	62.81
ESA-ELM 750 hidden neurons	95.63	82.50	82.50	82.50	69.64
ESA-ELM 775 hidden neurons	95.00	80.00	80.00	80.00	66.11
ESA-ELM 800 hidden neurons	95.63	82.50	82.50	82.50	69.64
ESA-ELM 825 hidden neurons	95.00	80.00	80.00	80.00	66.16
ESA-ELM 850 hidden neurons	95.00	80.00	80.00	80.00	66.16
ESA-ELM 875 hidden neurons	96.25	85.00	85.00	85.00	73.41
ESA-ELM 900 hidden neurons	95.00	80.00	80.00	80.00	66.20

https://doi.org/10.1371/journal.pone.0194770.t007

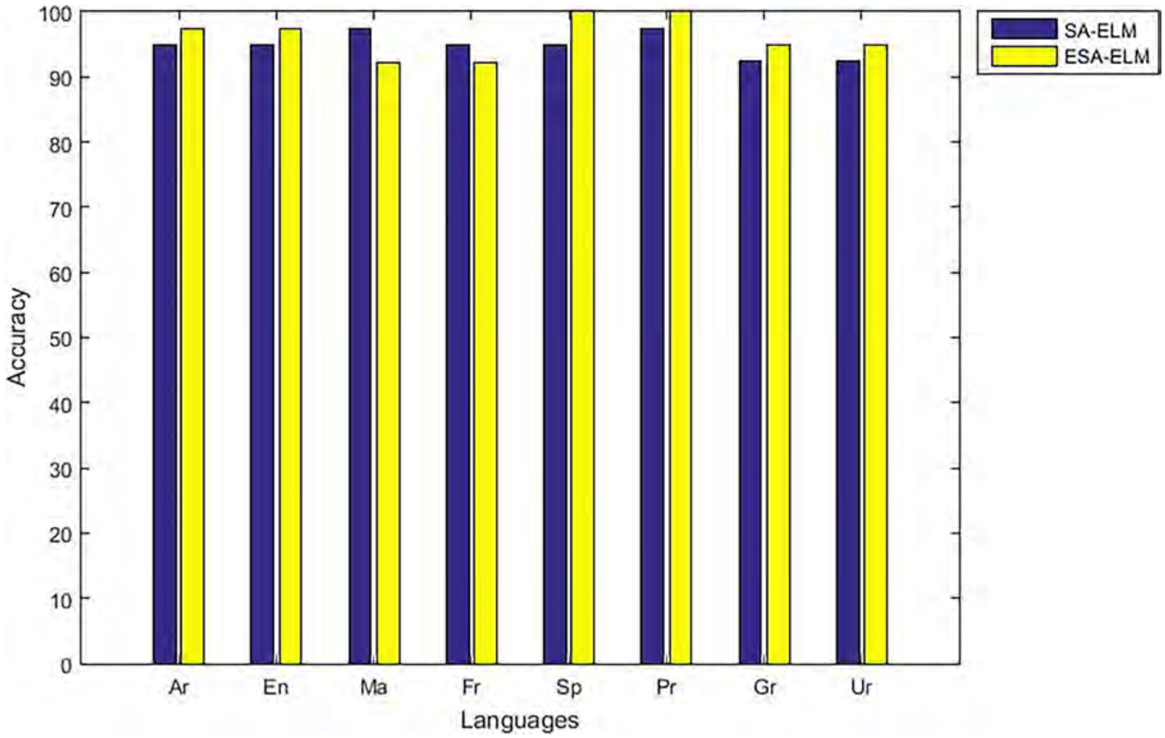

Fig 10. Accuracy measurement of the ESA-ELM and the SA-ELM for each language separately.

https://doi.org/10.1371/journal.pone.0194770.g010

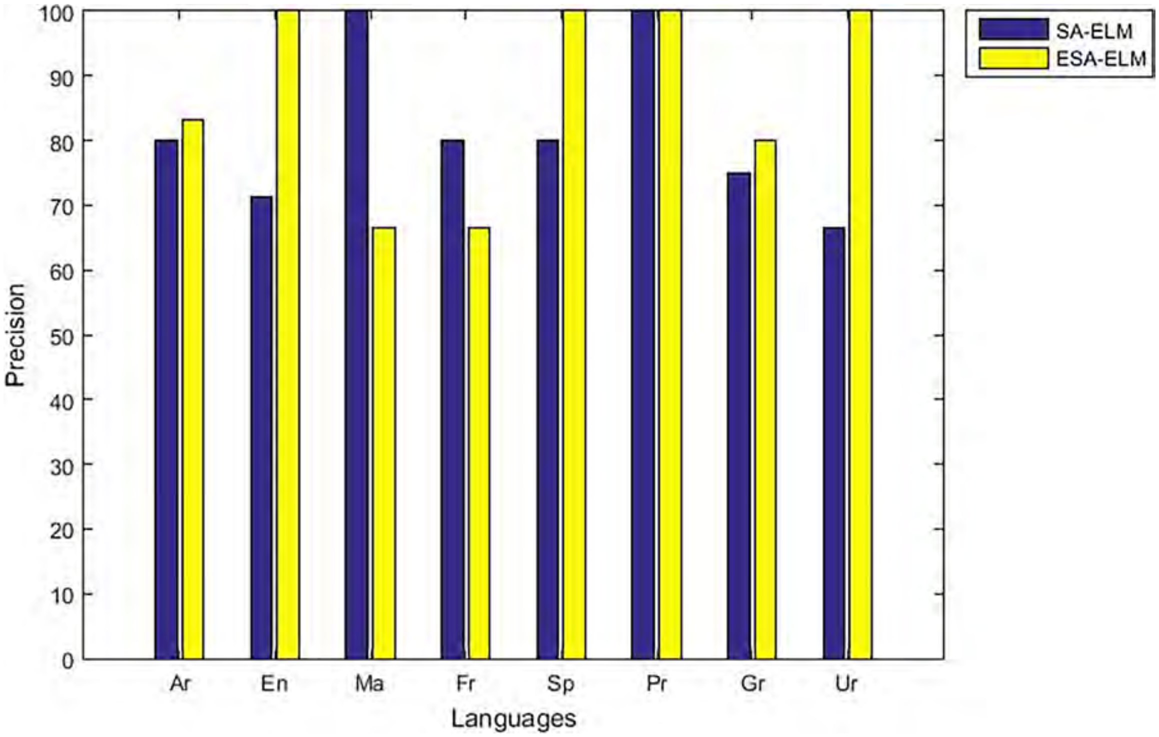

Fig 11. Precision measurement of the ESA-ELM and the SA-ELM for each language separately.

https://doi.org/10.1371/journal.pone.0194770.g011

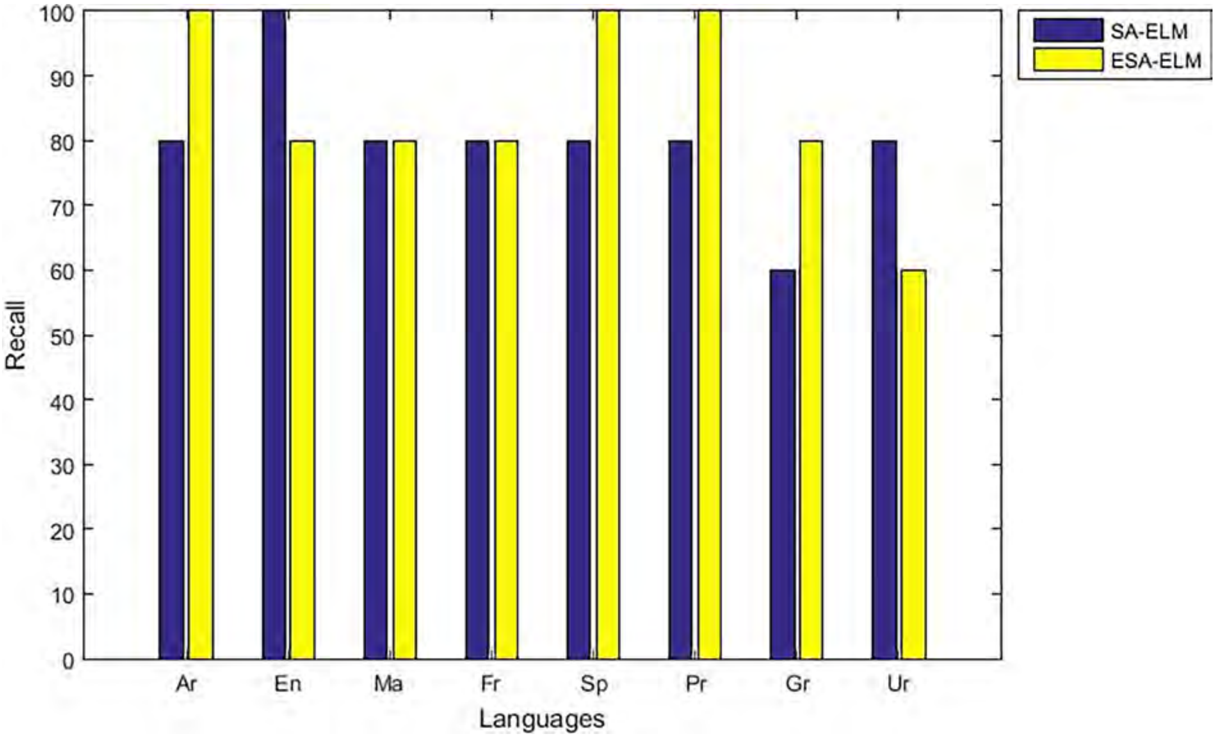

Fig 12. Recall measurement of the ESA-ELM and the SA-ELM for each language separately.

https://doi.org/10.1371/journal.pone.0194770.g012

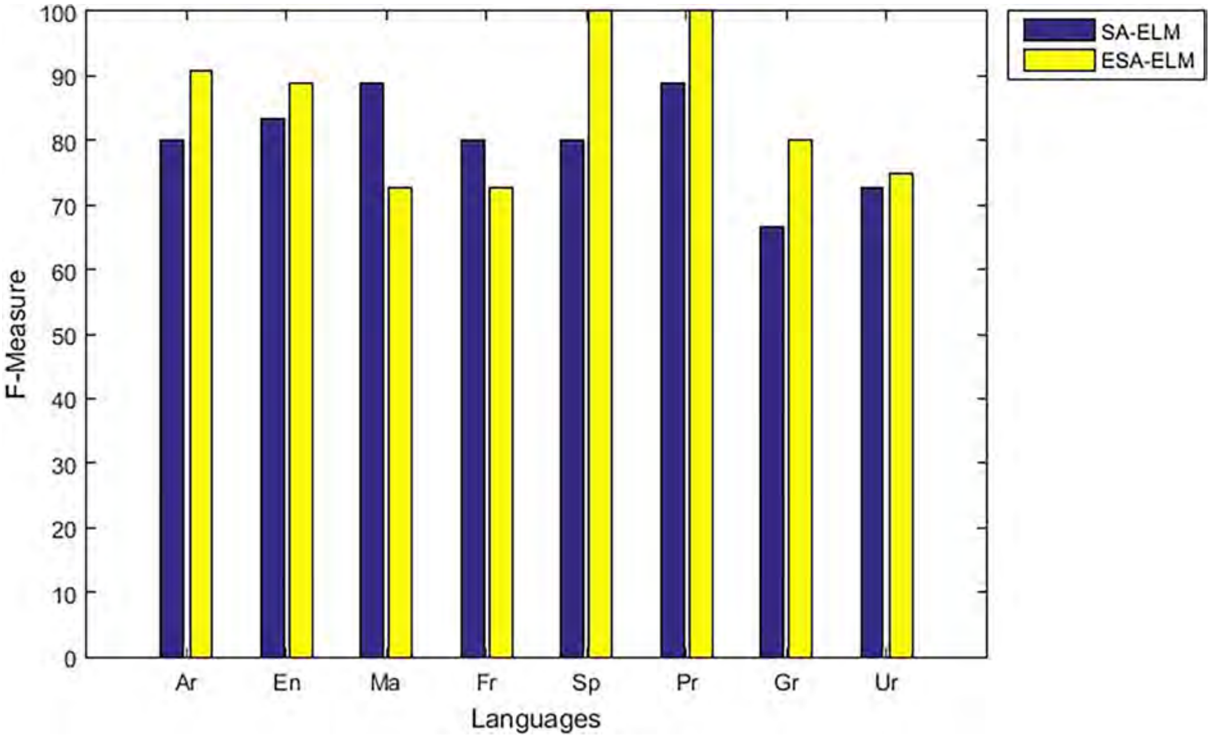

Fig 13. F-measure measurement of the ESA-ELM and the SA-ELM for each language separately.

https://doi.org/10.1371/journal.pone.0194770.g013

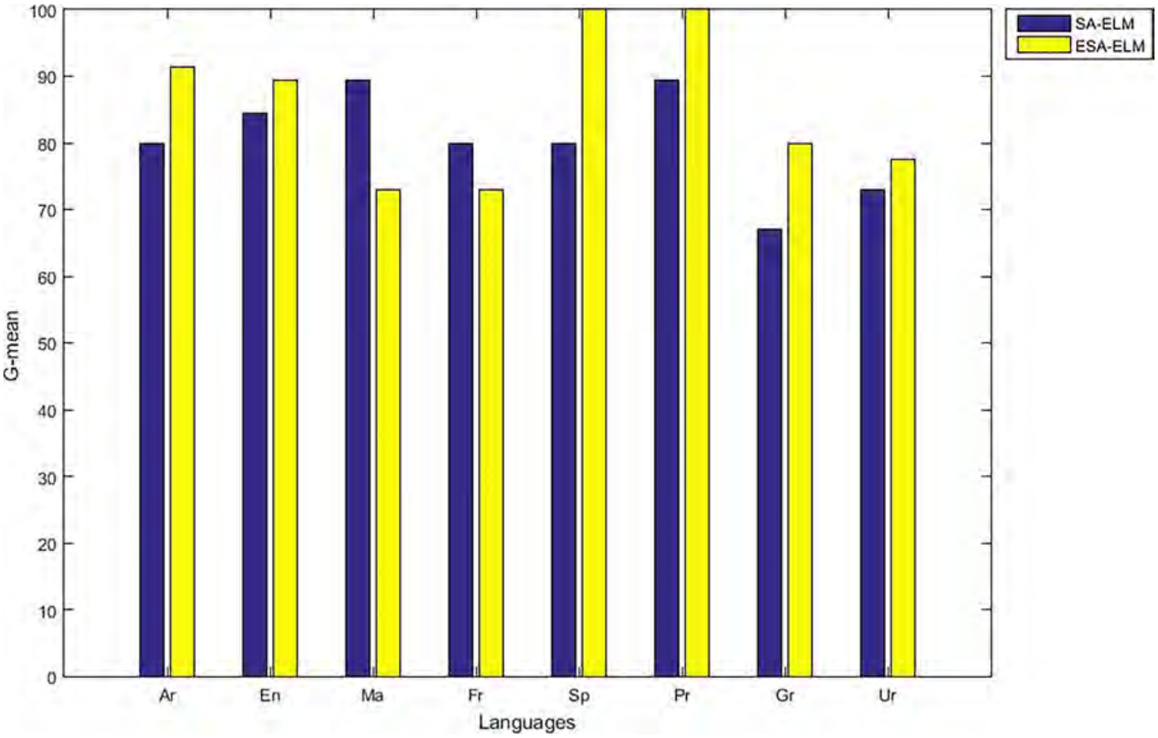

Fig 14. G-mean measurement of the ESA-ELM and the SA-ELM for each language separately.

https://doi.org/10.1371/journal.pone.0194770.g014

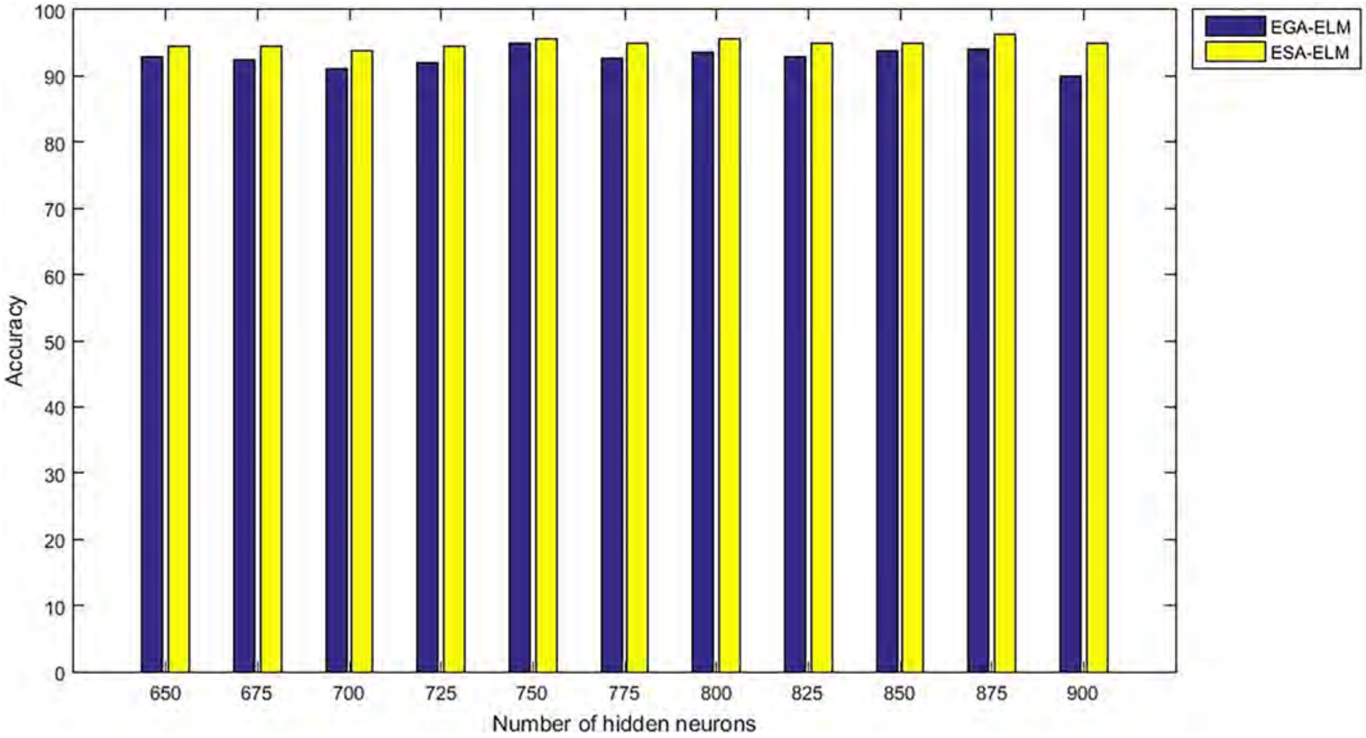

Fig 15. Accuracy measurement of the ESA-ELM and the EGA-ELM.

https://doi.org/10.1371/journal.pone.0194770.g015

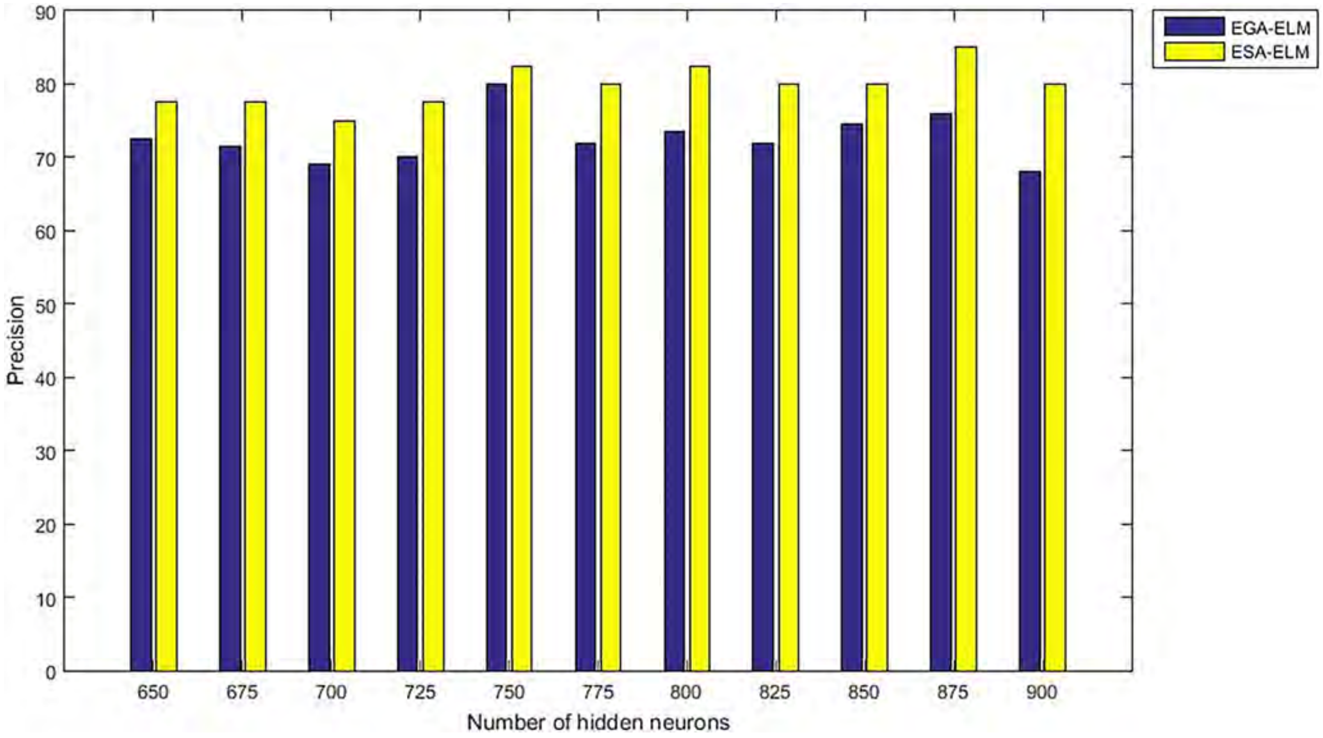

Fig 16. Precision measurement of the ESA-ELM and the EGA-ELM.

https://doi.org/10.1371/journal.pone.0194770.g016

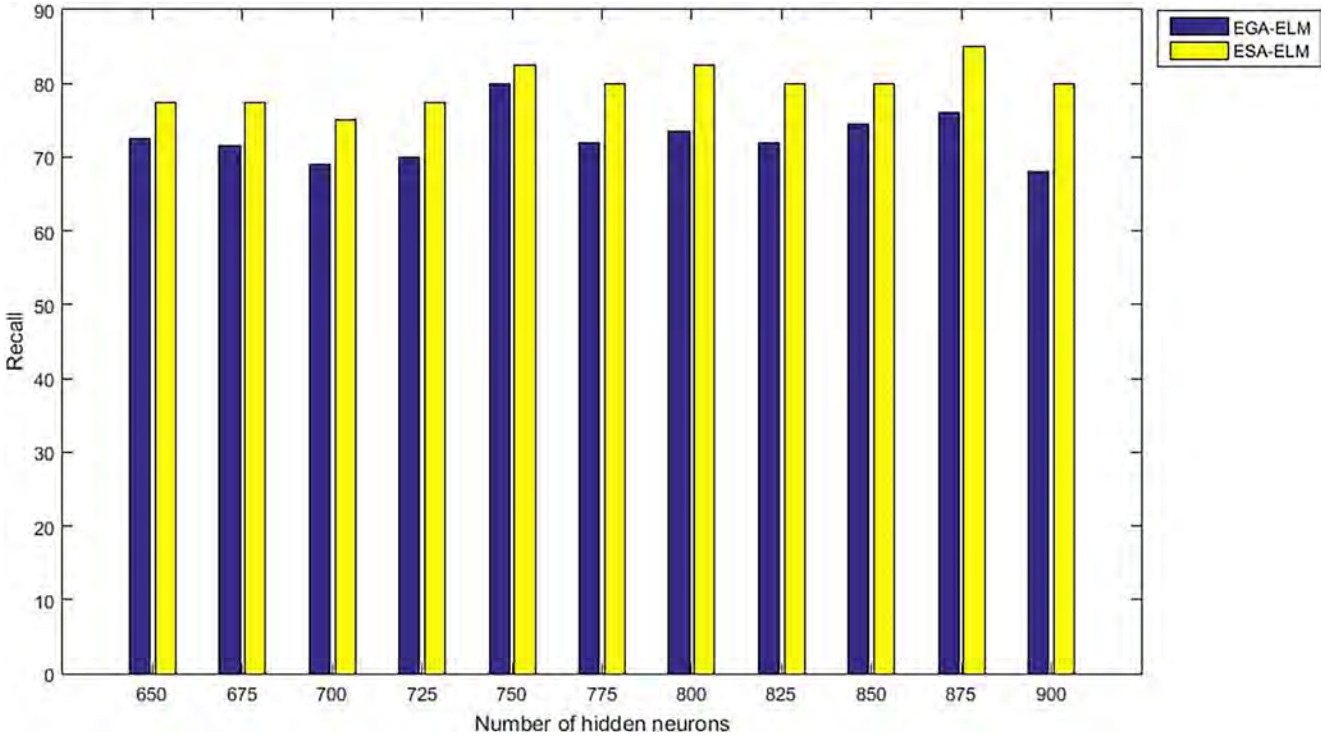

Fig 17. Recall measurement of the ESA-ELM and the EGA-ELM.

https://doi.org/10.1371/journal.pone.0194770.g017

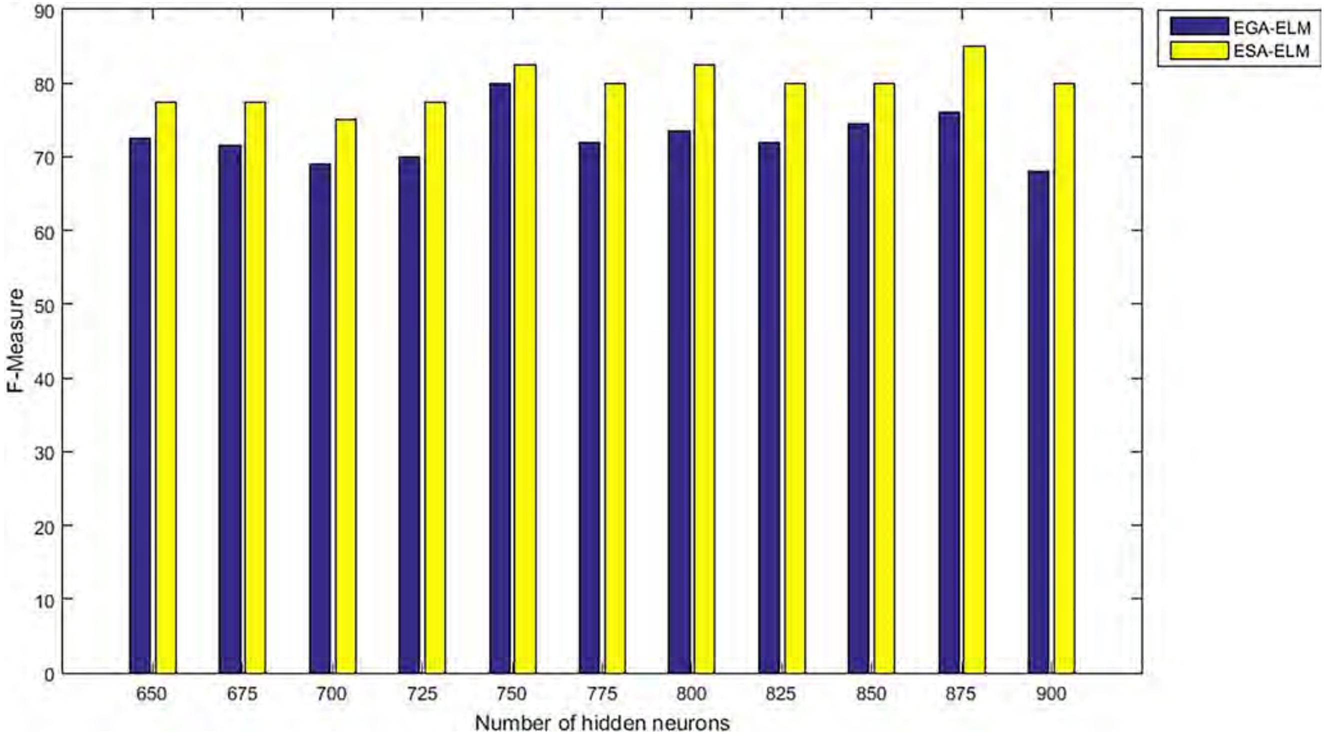

Fig 18. F-measure measurement of the ESA-ELM and the EGA-ELM.

https://doi.org/10.1371/journal.pone.0194770.g018

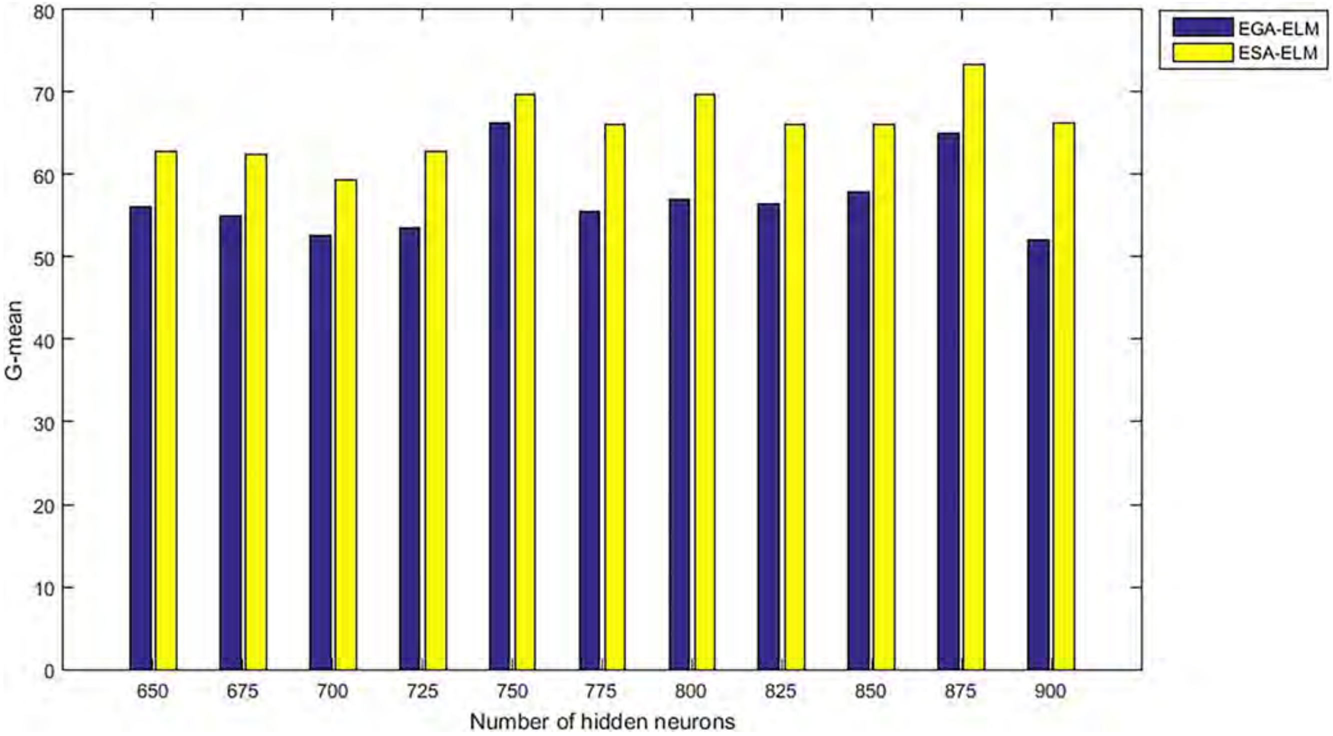

Fig 19. G-mean measurement of the ESA-ELM and the EGA-LEM.

https://doi.org/10.1371/journal.pone.0194770.g019

validated based on the optimisation of standard, but complex multi-variable mathematical functions and compared to the ATLBO. The EATLBO was then used in the ESA-ELM as an optimisation block for the weights of the input hidden layer neurones. The results identify the excellent (i.e. favourable) superiority of ESA-ELM compared to SA-ELM for LID. Moreover, different values of the learning model parameters were tested where the results identified the optimal parameters for learning. Following this study, the plan is to develop the LID system that can accommodate on-line execution of the feature extraction and classification while applying real-time aspects. Because only off-line LID was considered in this study. An online LID system is therefore recommended to accommodate a wider range of LID applications such as conferences, phone services, etc. Additionally, will be explored alternate optimisation methods for ELM being both cost-effective from a computational perspective and quality (integrity) from an accuracy perspective using technology. Furthermore, the front-end (features extraction) required a long time to extract the needed features thus, utilize the parallel processing can reduce the time consumption and cost greatly.

Supporting information

S1 File. LID code.
(RAR)

S2 File. LID dataset 2017.
(ZIP)

S1 Text. Provides the languages, youtube channel names, and the URLs for every single channel that we have used to collocate our dataset.
(TXT)

Acknowledgments

This project is funded by Malaysian government under research code FRGS/1/2016/ICT02/UKM/02/14.

Author Contributions

Conceptualization: Musatafa Abbas Abbood Albadr.

Data curation: Musatafa Abbas Abbood Albadr.

Formal analysis: Musatafa Abbas Abbood Albadr, Fahad Taha AL-Dhief.

Funding acquisition: Sabrina Tiun, Fahad Taha AL-Dhief.

Investigation: Musatafa Abbas Abbood Albadr, Mahmoud A. M. Sammour.

Methodology: Musatafa Abbas Abbood Albadr.

Project administration: Musatafa Abbas Abbood Albadr.

Resources: Musatafa Abbas Abbood Albadr.

Software: Musatafa Abbas Abbood Albadr.

Supervision: Sabrina Tiun.

Validation: Musatafa Abbas Abbood Albadr.

Visualization: Musatafa Abbas Abbood Albadr.

Writing – original draft: Musatafa Abbas Abbood Albadr.

Writing – review & editing: Musatafa Abbas Abbood Albadr.

References

1. Lee K, Li H, Deng L, Hautamäki V, Rao W, Xiao X, et al. The 2015 NIST language recognition evaluation: the shared view of I2R, Fantastic4 and SingaMS; 2016. pp. 3211–3215.
2. Zazo R, Lozano-Diez A, Gonzalez-Dominguez J, Toledano DT, Gonzalez-Rodriguez J (2016) Language identification in short utterances using long short-term memory (LSTM) recurrent neural networks. PloS one 11: e0146917. https://doi.org/10.1371/journal.pone.0146917 PMID: 26824467
3. Garg A, Gupta V, Jindal M (2014) A survey of language identification techniques and applications. Journal of Emerging Technologies in Web Intelligence 6: 388–400.
4. Li J, Mohamed A, Zweig G, Gong Y. LSTM time and frequency recurrence for automatic speech recognition; 2015. IEEE. pp. 187–191.
5. Hafen RP, Henry MJ (2012) Speech information retrieval: a review. Multimedia systems 18: 499–518.
6. Ambikairajah E, Li H, Wang L, Yin B, Sethu V (2011) Language identification: A tutorial. IEEE Circuits and Systems Magazine 11: 82–108.
7. Lopez-Moreno I, Gonzalez-Dominguez J, Martinez D, Plchot O, Gonzalez-Rodriguez J, Moreno PJ (2016) On the use of deep feedforward neural networks for automatic language identification. Computer Speech & Language 40: 46–59.
8. Jiang B, Song Y, Wei S, Liu J-H, McLoughlin IV, Dai L-R (2014) Deep bottleneck features for spoken language identification. PloS one 9: e100795. https://doi.org/10.1371/journal.pone.0100795 PMID: 24983963
9. Deng C, Huang G, Xu J, Tang J (2015) Extreme learning machines: new trends and applications. Science China Information Sciences 58: 1–16.
10. van Heeswijk M (2015) Advances in extreme learning machines.
11. Wang Y, Cao F, Yuan Y (2011) A study on effectiveness of extreme learning machine. Neurocomputing 74: 2483–2490.
12. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70: 489–501.
13. Nayak P, Mishra S, Dash P, Bisoi R (2016) Comparison of modified teaching–learning-based optimization and extreme learning machine for classification of multiple power signal disturbances. Neural Computing and Applications 27: 2107–2122.
14. Yang Z, Zhang T, Zhang D (2016) A novel algorithm with differential evolution and coral reef optimization for extreme learning machine training. Cognitive neurodynamics 10: 73–83. https://doi.org/10.1007/s11571-015-9358-9 PMID: 26834862
15. Niu P, Ma Y, Li M, Yan S, Li G (2016) A kind of parameters self-adjusting extreme learning machine. Neural Processing Letters 44: 813–830.
16. Liang N-Y, Huang G-B, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Transactions on neural networks 17: 1411–1423. https://doi.org/10.1109/TNN.2006.880583 PMID: 17131657
17. Pal M, Maxwell AE, Warner TA (2013) Kernel-based extreme learning machine for remote-sensing image classification. Remote Sensing Letters 4: 853–862.
18. Minhas R, Baradarani A, Seifzadeh S, Wu QJ (2010) Human action recognition using extreme learning machine based on visual vocabularies. Neurocomputing 73: 1906–1917.
19. Bhasin V, Bedi P. Multi-class jpeg steganalysis using extreme learning machine; 2013. IEEE. pp. 1948–1952.
20. Pan C, Park DS, Yang Y, Yoo HM (2012) Leukocyte image segmentation by visual attention and extreme learning machine. Neural Computing and Applications 21: 1217–1227.
21. Mohammed AA, Minhas R, Wu QJ, Sid-Ahmed MA (2011) Human face recognition based on multidimensional PCA and extreme learning machine. Pattern Recognition 44: 2588–2597.
22. Peng Y, Wang S, Long X, Lu B-L (2015) Discriminative graph regularized extreme learning machine and its application to face recognition. Neurocomputing 149: 340–353.
23. Xiang J, Westerlund M, Sovilj D, Pulkkis G. Using extreme learning machine for intrusion detection in a big data environment; 2014. ACM. pp. 73–82.
24. Iosifidis A, Tefas A, Pitas I (2016) Graph embedded extreme learning machine. IEEE transactions on cybernetics 46: 311–324. https://doi.org/10.1109/TCYB.2015.2401973 PMID: 25751883

25. Huang G, Song S, Gupta JN, Wu C (2014) Semi-supervised and unsupervised extreme learning machines. IEEE transactions on cybernetics 44: 2405–2417. https://doi.org/10.1109/TCYB.2014.2307349 PMID: 25415946

26. Liu B, Xia S-X, Meng F-R, Zhou Y (2016) Manifold regularized extreme learning machine. Neural Computing and Applications 27: 255–269.

27. Huang G-B, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42: 513–529.

28. Huang G-B (2014) An insight into extreme learning machines: random neurons, random features and kernels. Cognitive Computation 6: 376–390.

29. Huang G-B, Chen L, Siew CK (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Networks 17: 879–892. https://doi.org/10.1109/TNN.2006.875977 PMID: 16856652

30. Xu J, Zhang W-Q, Liu J, Xia S (2015) Regularized minimum class variance extreme learning machine for language recognition. EURASIP Journal on Audio, Speech, and Music Processing 2015: 22.

31. Lan Y, Hu Z, Soh YC, Huang G-B (2013) An extreme learning machine approach for speaker recognition. Neural Computing and Applications 22: 417–425.

32. Han K, Yu D, Tashev I. Speech emotion recognition using deep neural network and extreme learning machine; 2014.

33. Muthusamy H, Polat K, Yaacob S (2015) Improved emotion recognition using gaussian mixture model and extreme learning machine in speech and glottal signals. Mathematical Problems in Engineering 2015.

34. Albadra MAA, Tiuna S (2017) Extreme Learning Machine: A Review. International Journal of Applied Engineering Research 12: 4610–4623.

35. Rao RV, Savsani VJ, Vakharia D (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design 43: 303–315.

36. Sokolova M, Japkowicz N, Szpakowicz S. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation; 2006. Springer. pp. 1015–1021.