International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Analysis of Extreme Learning Machine Based on Multiple Hidden Layers

M. Vinoth Kumar[1], G. Kiran Kumar[2], Ahmed Mudassar Ali[3], S.K. Rajesh Kanna[4], K. Muthu Lakshmi[5]

Submitted: 18/10/2023 **Revised:** 02/12/2023 **Accepted:** 11/12/2023

Abstract: Humans can process vast volumes of data, learn about the behavior of the data, and make better decisions based on the analysis that results from machine learning (ML). ML has uses in many different domains. DL and ML techniques have gained widespread recognition and are being used in many real-time engineering applications due to their remarkable performance. To create intelligent and automated programs that can manage data in fields like cyber-security, health, and intelligent schemes, one must possess a solid understanding of machine learning. The multiple hidden layer exponential logistic regression model (also known as MELM) proposed in this study retains the properties of the parameters of the first hidden layer. A system that approaches the expected hidden layer output with the real output zero error can be constructed in order to determine the parameters of the remaining hidden layers. Extensive studies on the MELM algorithm for regression and classification demonstrate that, in comparison to other multilayer ELMs, the ELM, and two-hidden-layer ELM (TELM), it may yield the intended outcomes based on average precision and strong generalisation performance. This research will function as a point of reference for scholars and experts in the industry. Additionally, from a technological perspective, it will provide a standard for decision-makers across many application domains and real-world situations.

Keywords: *automated programmes, machine learning, intelligent systems, industry*

1. Introduction

People, businesses, and society as a whole now have the chance to gather a lot of data thanks to the Internet's and related technologies' explosive expansion. Yet, information overload is frequently the result of these massive data sets. A person experiences information overload when their cognitive abilities cannot handle the volume of input (data, for example). Overwhelming amounts of information can cause people to ignore, misunderstand, or overlook important facts [1]. The cognitive ability to process vast volumes of data is not present in humans. As a result, the field of data science has been established. To extract information from enormous data sets, data science integrates the traditional fields of databases, distributed systems, statistics, and data mining. ML is among the types of data analysis available to data scientists. Artificial intelligence can be used to teach computers without explicit programming. A computer can use its newly acquired knowledge to identify patterns in similar data once it has identified patterns in a training set of data. Computer systems may also adapt and learn from their experiences thanks to machine learning.

There are several uses for machine learning algorithms. Examples include medicine reaction prediction, social media, banking and finance, network security, education, spam filtering, housing pricing prediction, and data architecture in healthcare systems. About the difficulties that several fields—with topics like social media, network security, healthcare, education, and banking and finance being difficult subjects, the goal of this book is to give a brief overview of the pertinent studies. It also offers several study avenues to explore the function and possibilities of using machine learning to these problems. The approach used in this work is shown in Figure 1.1. We then go over each of these processes in detail.

[1]*Associate Professor,*
Department of Information Science and Engineering,
RV Institute of Technology and Management, Bangalore-560076
Email: vinojimail@gmail.com
ORCID: 0000-0001-9394-5078
[2]*Assistant Professor, Computer Science Data Science,*
Madanapalle Institute of Technology & Science
Email: kirankumarg@mits.ac.in
ORCID: 0009-0009-3874-165X
[3]*Director, MEASI Institute of Information Technology*
Email: mudassar.a.ali@gmail.com
ORCID: 0000-0003-1098-3605
[4]*Professor, Mechanical Engineering Department,*
Rajalakshmi Institute of Technology
Email: skrkanna@gmail.com
ORCID: 0000-0003-1013-008X
[5]*Associate Professor, Department of Information Technology, Panimalar Engineering College.*
Email: muthulakshmi_it@panimalar.ac.in
ORCID: 0000-0003-1230-2789

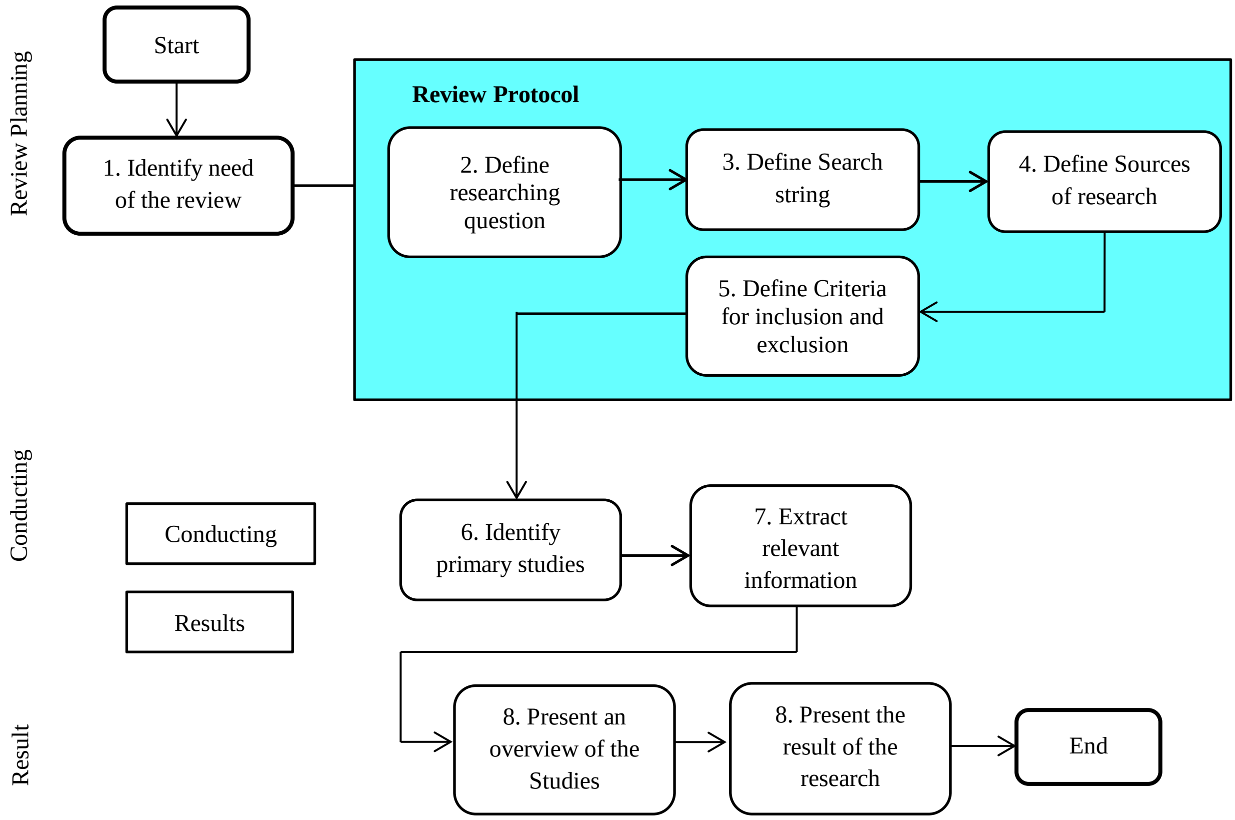

Fig. 1.1. The selection process used to include publications

Deep learning has been employed by academics and industry to analyze data and extract pertinent information as a result of the expansion of image and big data claims. Deep learning architectures have difficulties when used in large networks, including high computing costs, sluggish convergence, disappearing gradients, and hardware constraints during training. In this comprehensive analysis, our primary focus is on examining the potential of CELM as a fast-training substitute for deep learning architectures, while circumventing the need for gradient computations which are essential for network weight updates. In many applications, CELMs have been able to maintain a respectable quality of results despite resolving some of the most challenging deep learning problems in recent years.

An incredibly quick training stage and good generalization performance characterize the ELM. The foundation of high train speed is the random selection of input weights and biases, followed by analytical computation of the output weights; the primary distinctions between ELM techniques and conventional feed-forward neural network gradient-based learning techniques. Below is a summary of the contributions made by this paper.

- Highlighting some of the older books and their shortcomings that addressed these problems.
- Outlining possible frameworks for the application of machine learning and talking about its character and potential in tackling these issues.
- The learning phase of the ELM moves very quickly.
- Simple problems like the ELM can often handle issues with local minima; learning rate, momentum rate, and over-fitting that arise in traditional gradient-based learning algorithms.
- Many non-differentiable activation functions can be trained in single layer feed forward networks (SLFN) using the ELM technique.

The classic ELM algorithm has gained popularity and is applied to several fields, including image processing, regression issues, feature selection for classification, and power transmission line fault detection [2]. Several new ELM-based techniques have been put forth, including weighted ELM, bidirectional ELM (B-ELM), fully complex ELM (C-ELM), online sequential ELM (OS-ELM), regularised ELM, robust ELM, and pruned ELM. This study's objective is to provide a thorough analysis of the ELM learning process, including its alternatives for dividing up the input weights and figuring out the output weights. The conventional ELM was unstable because the input weights and biases were selected arbitrarily. The accuracy of the ELM was compared to the accuracy of employing present input weights and biases, which were found by back propagation. When compared to ELM and back propagation, the outcomes of the predefining weights approach were demonstrated to be stable and accurate.

Additionally, the outcomes were contrasted with a single-layer FFNN that does not employ a transfer function, input weight, or bias. Moore–Penrose generalized inverse, back-propagation, and linear regression was employed to determine the same's output weights. The accuracy outcomes of classifying various dataset kinds were contrasted.

The structure of the paper is as follows: The main ideas that direct the study are presented in Section 2. These include definitions of ELM, along with the corresponding pruning techniques and correlation coefficients. The procedures and concepts of the suggested methodology for factor-based neuron pruning are provided in section 3. Section 4 will address the test processes, which include the bases and techniques for fine-tuning in extreme learning machines that do binary pattern classification. The paper's conclusions will be given at the end of section 5.

2. Literature Review

Wang, J., et.al [3] The ELM is a feed-forward neural network with a single hidden layer. The centres, impact factors, and Fourier series nodes in RBF nodes and examples of hidden node features are the input weights and hidden node biases in additive nodes that are chosen at random, and the least squares approach is usually used to determine the output weights analytically. The ELM learns significantly quicker than the BP technique since changing the input weights is not necessary. Better generalization performance can also be attained by ELM. To solve the MIL problem, a modified ELM is suggested, wherein the least negative example from the negative bag or the greatest positive example from the positive bag is selected.

Jhaveri, R. H., et.al [4] The performance and features of machine learning algorithms, along with the type of data, will ascertain the machine learning-based solution's effectiveness and efficiency. Building data-driven systems can be accomplished successfully with the use of ML techniques such association rule learning, data clustering, regression, feature engineering, and dimensionality reduction. Intelligent data analysis uses a new technology called Deep Learning, which came about as a result of the ANN family of machine learning algorithms. Each ML algorithm has a distinct goal, and even multiple algorithms applied to the same category will produce different results depending on the type and attributes of the data.

Wang, J., et.al [5] Random hidden nodes, meanwhile, claim to be able to approximate anything universally. Based on theoretical study, when trained to achieve the global optimal solution with random parameters, ELMs have a higher chance of success than classical networks with all the parameters set. When compared to SVM, elliptic curve modeling (ELM) typically produces superior classification results with fewer optimization constraints. ELM is frequently used in a range of learning tasks, including feature mapping, regression, classification, and clustering, because of its excellent generalization ability and faster training speed. ELM changed as a result of the numerous modifications that were put forth to increase its generalizability and stability for particular uses.

Ding, S., et.al [6] Feed-forward neural networks have proven to be highly effective in numerous fields in recent decades due to their apparent benefits. From the input samples, it could, directly approximate complicated nonlinear mappings on the one hand. However, it can also provide models for a wide range of artificial and natural phenomena that are difficult for typical parametric techniques to manage. The single-hidden-layer feed-forward network is one of the most often used feed-forward neural networks which has received substantial research attention for its fault-tolerant design and learning capabilities from both a theoretical and practical standpoint.

Zhang, J., et.al [7] An effective foundation for combining the DBN algorithm with the ELM model is provided by the model's serviceability. In unsupervised and semi-supervised learning, the Manifold Regularisation theory is the regularisation framework of choice. Manifold Regularisation ELM in conjunction with our model may be an effective way to extract relevant data and reduce the complexity of the probability distribution. Other machine learning models, such as evolutionary algorithms and upper integral networks, were used in clustering issues in addition to the ELM model.

Huang, G. B et. al [8] Without a doubt, research on neural networks has resumed since the 1980s when hidden layers have become more important in learning. However, hidden neurons in every network must be modified since hidden layers are crucial and require learning circumstances which is a current puzzle in neural network research. This goes against the community's default expectations and understanding. Tens of thousands of academics from nearly every country have been diligently throughout the 1980s; researchers have been looking for learning methods to train various neural network types, mostly by modifying hidden layers.

Zhang, J., et.al [9] Classical ELMs are SLFNs, or generalized feed-forward networks with a single hidden layer. The weights of the output are then computed analytically after the hidden layer parameters of the ELM are randomly assigned. This is in contrast to the conventional gradient-based SLFN training techniques, which are computed using the least squares approach and take a long time and are prone to becoming trapped in the local minimum. The two steps in the ELM training process are computing the generalized inverse of the output weight matrix and randomly selecting 35 hidden layer values from a predefined interval.

de Campos Souza, P. V., et.al [10] The usage of correlation coefficient indices is introduced in this study, with a primary focus on unbalanced data, where the significance of one name is significantly greater than that of another. According to the chosen criterion, the proposed method would choose neurons that meet a specific attribute in order to identify the most representative neurons. This work aims to present a simple approach based on correlation coefficients for neuron pruning in ELM. This will be achieved by running pattern classification tests on unbalanced data to show how well the method for carrying out less important neurons are pruned while keeping the accuracy level of the model appropriate for the type of challenge presented.

3. Methods and Materials

When it comes to training new classifiers, ELM is a low computational time learning strategy since the weights and biases of the hidden layer are assigned at random and the output weights are analytically derived using simple mathematical manipulations. The ELM algorithm has been the subject of increasing research interest in recent years, leading to the proposal of numerous modifications aimed at enhancing its performance. Additionally, the ELM algorithm has been utilized to optimize several issues related to machine learning, computational intelligence, pattern reorganization, and other related fields. Next, we provided a summary of the research findings on the different ELM variants.

3.1 Extreme Learning Machine

The goal of the extreme learning machine, a single-hidden-layer feed-forward neural network with an input layer, hidden layer, and output layer, is to improve generalization performance while skipping the time-consuming iterative training procedure. The ELM computes the hidden layer output matrix, sets the number of hidden neurons in the network, randomises the weights between the input and hidden layers and the hidden neurons' bias during algorithm execution, and determines the weight between the hidden layer and the output layer using the Moore-Penrose pseudoinverse in accordance with the least-squares method [11]. This contrasts with conventional neural network learning methods, which produce local optimal solutions with ease and randomly establish all network training parameters (e.g., the BP algorithm). The ELM has the benefit of quick learning speed due to its straightforward network architecture and condensed parameter computation methods. In Figure 3.1, the original ELM structure is presented.

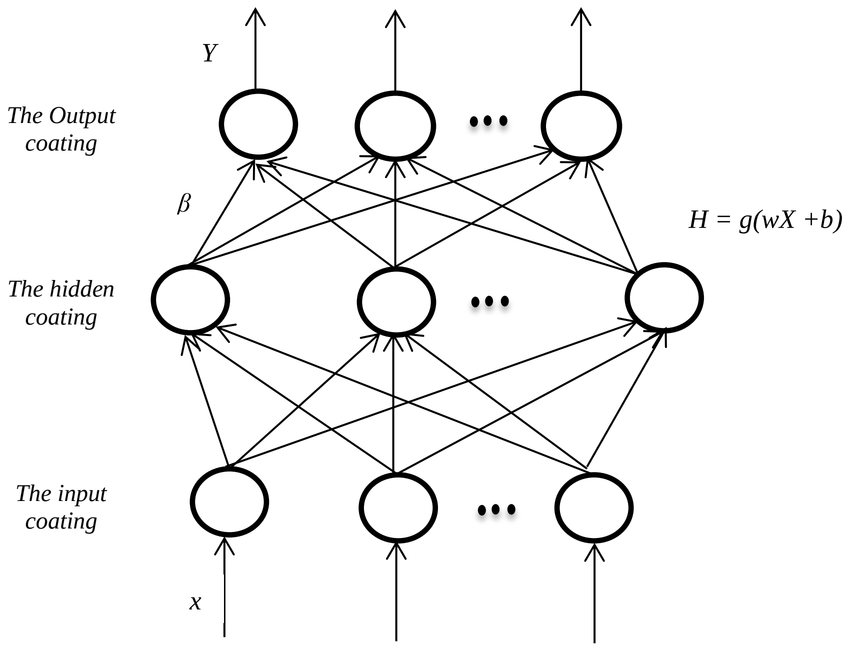

Fig. 3.1. The ELM's structure

The three types of neurons that make up the structure of the extreme learning machine network are input layer neurons, hidden layer neurons, and output layer neurons. This is shown in Figure 3.1. Firstly, let us examine the training sample {y, x} = y_{11}. Additionally, we have an input feature y_{nq} and a desired matrix X_{nq} consisting of the training samples. The matrices X and Y can be represented as follows.

$$Y = \begin{vmatrix} y_{11} & y_{12} & y_{pq} \\ y_{21} & y_{22} & y_{pq} \\ \cdots & & \\ y_{m1} & y_{m2} & y_{nq} \end{vmatrix} \quad (1)$$

$$X = \begin{vmatrix} X_{11} & X_{12} & X_{pq} \\ X_{21} & x_{22} & X_{pq} \\ \cdots & & \\ X_{m1} & X_{m2} & X_{nq} \end{vmatrix} \quad (2)$$

where the input and output matrices' dimensions are represented by the parameters m and n. Next, The ELM casually determines the masses of the input layer and hidden coating:

$$V = \begin{vmatrix} v_{11} & v_{12} & v_{pq} \\ v_{21} & v_{22} & v_{pq} \\ \cdots & & \\ v_{m1} & v_{m2} & v_{nq} \end{vmatrix} \quad (3)$$

where wt denotes the weights between the wth hidden layer neuron and the wth input layer neuron. Thirdly, the ELM makes the following assumption about the weights between the output layer and the hidden layer:

$$C = \begin{vmatrix} c_{11} & c_{12} & c_{pq} \\ c_{21} & c_{22} & c_{pq} \\ \cdots & & \\ c_{m1} & c_{m2} & c_{nq} \end{vmatrix} \quad (4)$$

where w indicates the weights between the u output layer neuron and the v hidden layer neuron. Fourthly, the bias of the hidden layer neurons is randomly adjusted by the ELM:

$$C = c_1, c_2, \ldots\ldots c_n . L \quad (5)$$

The network activation function l_1 is selected by the ELM in the fifth step. As per Figure 3.2, it is possible to express the output matrix T in the following way:

$$L = l_1, c_2, \ldots\ldots c_n . L \quad (6)$$

The following is a list of each column vector in the output matrix L:

$$L_J = \begin{vmatrix} L_{11} \\ L_{12} \\ L_{13} \end{vmatrix} \begin{vmatrix} c_{11} & c_{12} & c_{pq} \\ c_{21} & c_{22} & c_{pq} \\ \cdots & & \\ c_{m1} & c_{m2} & c_{nq} \end{vmatrix} \quad (7)$$

Sixth, think about equations (5) and (6), and we can obtain

$$H \cap = l \quad (8)$$

The least squares approach is used to determine the weight matrix values of the unique solution with the least amount of error.

$$H \cap = K * l \quad (9)$$

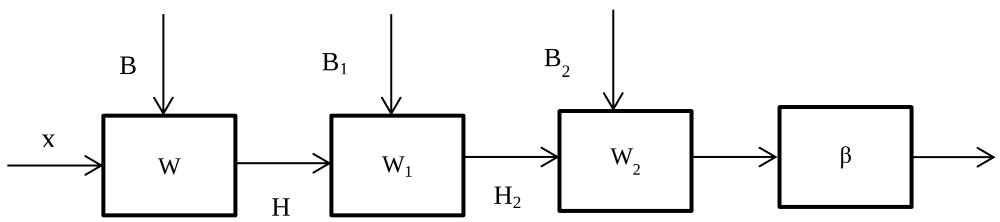

Fig. 3.2. The TELM's workflow

We include a regularization term in the β to increase the network's capacity for generalization and to stabilize the results. When the number of hidden layer neurons is lower than that of training specimens, β can be expressed as follows:

$$\partial = \left(\frac{2}{\ni} + l^T L\right) l^T L \quad (10)$$

β can be communicated as follows when there are more hidden layer nodes than training samples:

$$\partial = l^T \left(\frac{2}{\ni} + l^T L\right) l^T L \quad (11)$$

3.2 Dual-Layer Hidden Layer

The TELM adds the second hidden layer's parameter setting step in an effort to get the outputs of the hidden layer closer to what is desired. The enhanced mapping of the link between input and output signals found by the TELM is ultimately what constitutes the two-hidden-layer ELM. The network architecture of TELM is comprised of an input layer, two hidden layers, and an output layer [12], and l hidden neurons in each hidden layer. For (x), the network's activation function is chosen.

To create a single hidden layer, the TELM first merges the two hidden layers together. The weight u and bias c of the first hidden layer are initialised randomly. Hence, L+H might be used to represent the hidden layer's output. The output layer and the second hidden layer are then connected via the output weight matrix β.

$$\partial = L + H \tag{12}$$

$$H(V_1 L + C) = L_1 \tag{13}$$

where b is the bias of the second hidden layer, c is its anticipated output, d is the weight matrix separating the first and second hidden layers, and c is the output matrix of the first hidden layer. On the other hand, the anticipated result of the secondary concealed layer can be acquired by computing

$$L_1 = T\partial \tag{14}$$

The matrix V_{LF} is now defined by the TELM, making the activation function's inverse function can be used to create a formula that makes it easy to find the parameters of the second hidden layer.

$$V_{LF} = G^{-1}(L)L_F \tag{15}$$

The TELM calculates after choosing the proper activation function (x), updating the second hidden layer's actual output as follows:

$$L_2 = H(V_{LE} L_E z) \tag{16}$$

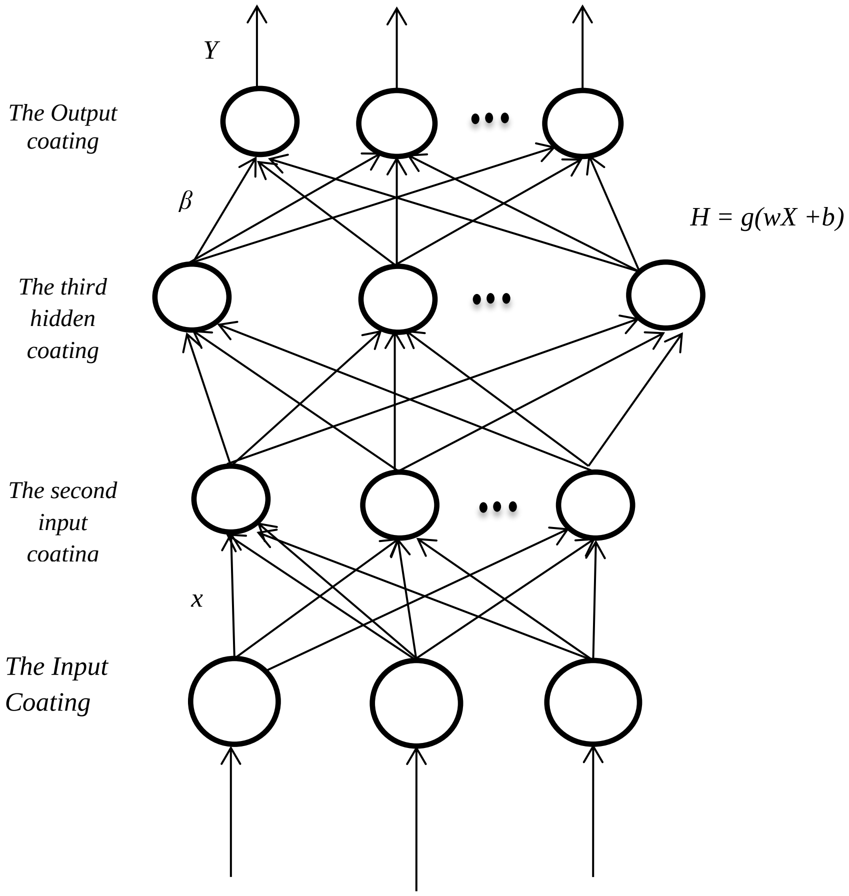

Fig. 3.3. The three-layered concealed-layer ELM architecture

The process of determining and updating the hidden layer's bias, the output weights between the second hidden layer and the output layer, and the weights between the first and second hidden layers is the primary emphasis of the TELM technique. Figure 3.3 illustrates the TELM architecture's workflow. Take into consideration the training sample datasets$H(V_{LE} L_E z)$, where T represents the labelled samples and X represents the input samples.

4. Implementation and Experimental Results

The outcomes of our suggested FC-IMRELM tests on benchmark statistics sets for problems with regression and classification are shown in this section. ELM, TEL, and MRELM are also assessed to look into how our methods' learning accuracy has improved. Every performance evaluation is carried out using MATLAB. 2014b computational environment with an Intel Core i5-7200U CPU running at 2.7 GHz, running on Windows 10 NVIDIA M150 graphics card, 2 GB GDDR5, and 8 GB RAM for videos. Additionally, to thoroughly compare as a consequence of the studies [13], the activation function of every algorithm is consistently assigned as the sigmoid function, where 1, is ⁽() = 1/(1 + exp(−)). There are twenty hidden neurons in all. Additionally, after 100 iterations of the program the output will be as the final value, averaged.

Table 1. The classification dataset's properties

Data usual	Qualities	Lessons	Exercise	Challenging
Bank letter	5	3	868	626
Gore	5	3	600	369
Diabetic	20	3	820	541
Copy	20	8	1554	958
Droop	6	3	5440	600
Coal spectral facts	9	5	286	220
Iron spectral facts	6	3	97	48

Table 2. Utilizing classification datasets, the average testing classification accurate percentage for the methods ELM, TELM, MRELM, and FC-IMRELM

Data agreed	ELM	TELM	MRELM	FC-IMRELM
Bank letter	55.28	38.22	86.81	62.67
Gore	58.36	38.12	60.03	36.93
Diabetic	20.95	38.13	82.09	54.16
Copy	20.12	87.24	15.54	95.81
Droop	63.52	35.21	54.40	60.03
Coal spectral facts	93.45	56.45	28.65	22.05
Iron spectral facts	67.77	35.12	97.23	48.23

Table 3. Details regarding the regression models

Data sets	Qualities	Preparation	Difficult
Bodyfat	15	323	50
pyrim	29	57	20
Triazines	70	565	36

4.1 Organization Issues

4.1.1. Features of Datasets for Sorting:

The literature and the UCI website provide the categorization datasets. We tested our FCIMRELM method utilizing real datasets gathered from the coal and iron ore industries as well as basic benchmark datasets to assess its robustness. Table 1 displays the datasets' features.

4.1.2. Assessment of Testing Precision on Classification Sets

To enhance the comprehensiveness of our performance evaluation, real datasets pertaining to intricate industry facts were used. To verify that our IMRELM algorithm improves learning accuracy, on both actual and simple benchmark datasets, the original ELM, TELM, and MRELM algorithms are tested. As we can see, the classification accuracy of each algorithm for the Banknote dataset is quite good from Table 2 and Figure 4.1 [14]. The methods TELM, MRELM, FC-IMRELM, and Wilt all perform better than the ELM algorithm for the datasets with blood, diabetes, coal, and iron spectra; the algorithm with the highest classification precision is FC-IMRELM. While the classification accuracy of the TELM, MRELM, and FC-IMRELM approaches is higher than that of the ELM algorithm for the Image dataset, the FC-IMRELM approach still has the greatest classification accuracy, at 95.98%.

The results of the experiments indicate that compared to the initial ELM, TELM, and MRELM algorithms, our FC-IMRELM approach offers noticeably greater average classification accuracy. Additionally, our technique is readily extensible to real-world applications, as demonstrated by computational experiments conducted with coal and iron spectrum datasets.

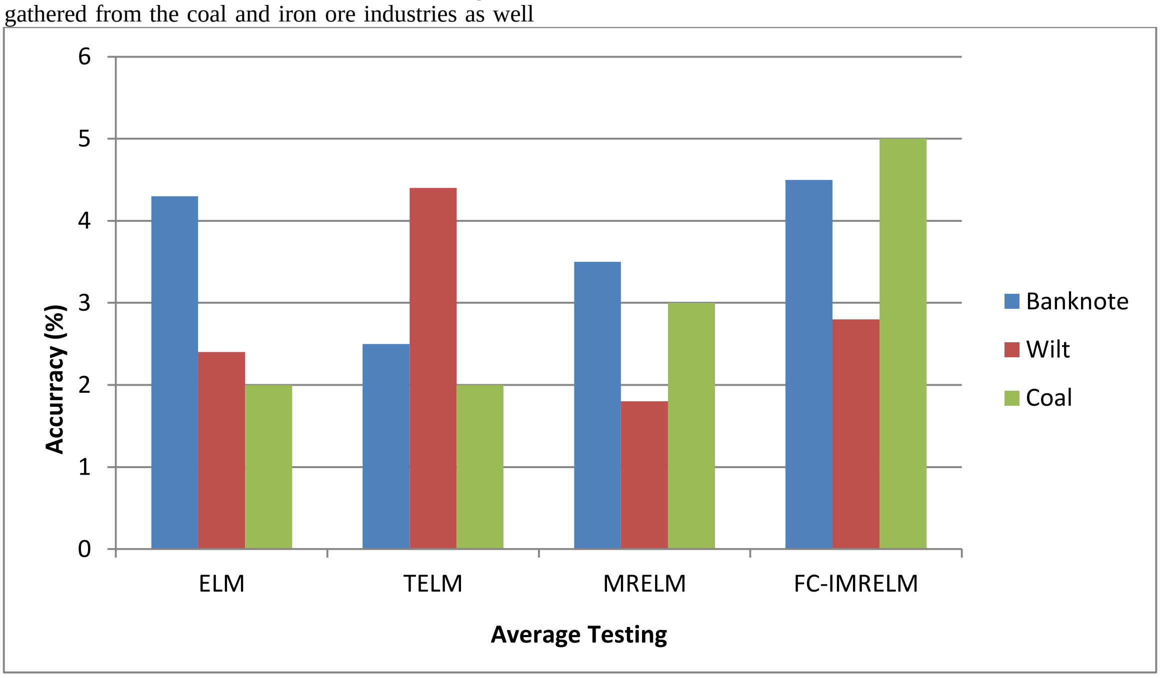

Fig. 4.1. The median testing classification accuracy utilizing classification datasets for the algorithms ELM, TELM, MRELM, and FC-IMRELM

4.2 Reversion Problems

The Reversion Problems in this study uses the root mean-square error (RMSE) and coefficient of determination (R2) as model performance evaluation indicators to verify the efficacy of the proposed FC-IMRELM technique. These are the expressions for R2 and RMSE:

$$RMSE = \sqrt{\frac{\sum_{j=2}^{Mtest}(x_j - x_j)}{Mtest}} \tag{17}$$

$$R2 = 2 = \sqrt{\frac{\sum_{j=2}^{Mtest}(y_j - x_j)}{\sum_{j=2}^{Mtest}(y_j - x_j)}} \tag{18}$$

The function of the model is better when the cost of RMSE is smaller and the value of R2 is closer to 1, as indicated by the range of values of R2.

4.2.1. Features of Datasets for Regression

The LIBSVM website is where the regression datasets may be found. Table 3 displays the attributes of the datasets.

Table 4. Comparing the original ELM, TELM, MRELM, and FC-IMRELM's performance on regression models

Data fixed	ELM		TELM		MRELM		FC-IMRELM	
Bodyfat	1.1132	2.58	1.113	1.396	1.1136	1.258	1.113	1.26
pyrim	1.14	2.36	1.13	1.57	1.26	1.72	1.23	1.58
Triazines	2.36	3.56	1.8	1.33	1.24	1.28	1.236	1.25

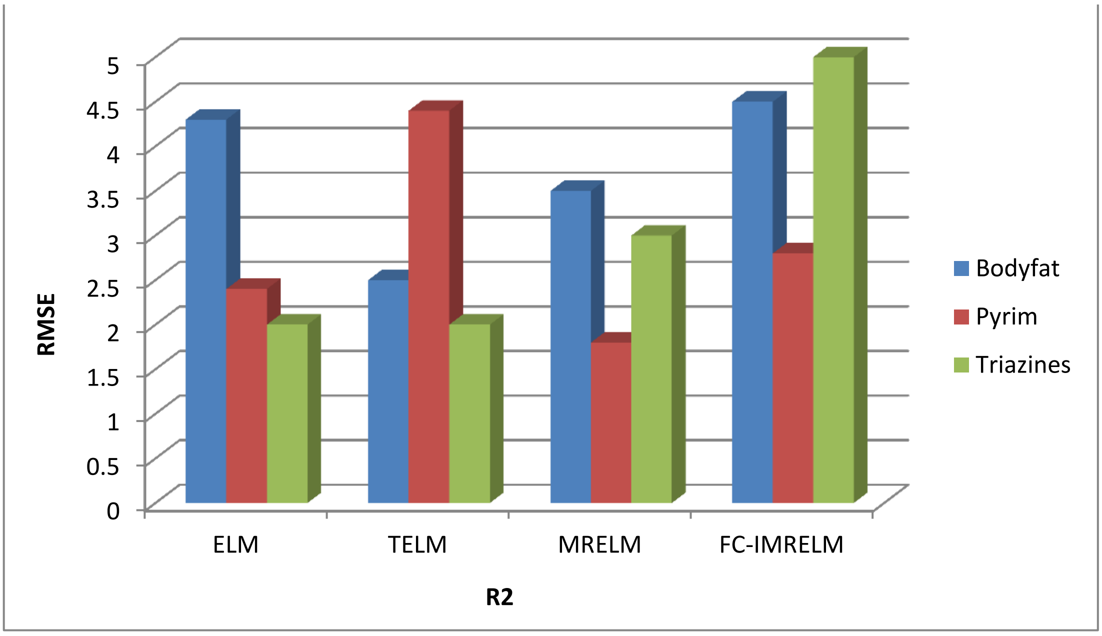

Fig. 4.2. The median R2 and RMSE utilizing regression datasets for the algorithms FC-IMRELM, MRELM, TELM, and ELM

4.2.2. Analysing the Precision of Estimates on Regression Datasets

Table 4 and Figure 4.2 suggest that the Body fat dataset is well-predicted by the algorithms TELM, MRELM, and FC-IMRELM; their R2 values are above 0.98 and their RMSE values are small. The ELM method, on the other hand, yields somewhat inferior predictions; its R2 values are 0.8. For the Pyrim and Triazines datasets, the FC-IMRELM approach yields better prediction results than the ELM, TELM, and MRELM algorithms; the FC-IMRELM approach's RMSE is the lowest and its R2 is the greatest. The aforementioned examination of the experimental findings suggests that the FC-IMRELM algorithm's benefit is its enhanced capacity to extract data characteristics for multiattribute data, hence improving its predictive ability.

5. Conclusion

The present research presents an innovative multiple instances learning method based on extreme learning machines. To train the extreme learning machine, the most representative instance is selected from each bag by altering the unique error function for the features of multiple instance difficulties.

Firstly, the FC-IMRELM approach, which is distinct from MRELM, employs forced positive definite Cholesky factorization to calculate the studies hidden layer measurements. With few computations and strong numerical stability, the training procedure is substantially simplified. Furthermore, the MRELM algorithm requires that throughout the training phase, the number of hidden layers stays constant and the network layout is present. On the other hand, the FC-IMRELM method can automatically adjust the network topology depending on training samples and minimise both structural and empirical risk to find the optimal number of hidden layers.

Moreover, unlike CF-FORELM, the FCIMRELM algorithm proposed in this study is meant for semi-positive definite matrices that emerge in the MRELM model's parameter-solving stage. The MRELM model approaches convergence more quickly and maintains numerical stability throughout the modeling process when the matrix's condition number is increased while the matrix is forced to be positive definite. Lastly, by including parameters to the FC-IMRELM technique has significantly enhanced the ELM model's potential to be more broadly applicable than a regular neural network by mitigating its structural and empirical risks. Forced positive definite Cholesky factorization is applied to determine the output weights in order to further minimise the computational cost associated with the growing number of hidden layers. The MRELM model's numerical instability is demonstrated by the prediction example can be effectively prevented using the FC-IMRELM approach. Additionally, it has the advantages of fast computing times and excellent forecast accuracy making it a unique and effective solution to the prediction problem.

References

[1] Injadat, M., Moubayed, A., Nassif, A. B., & Shami, A. (2021). Machine learning towards intelligent systems: applications, challenges, and opportunities. *Artificial Intelligence Review*, *54*, 3299-3348.

[2] Ertuğrul, Ö. F., & Kaya, Y. (2014). A detailed analysis on extreme learning machine and novel approaches based on ELM. *American Journal of computer science and engineering*, *1*(5), 43-50.

[3] Wang, J., Cai, L., Peng, J., & Jia, Y. (2015). A novel multiple instance learning method based on extreme learning machine. *Computational Intelligence and Neuroscience*, *2015*, 3-3.

[4] Jhaveri, R. H., Revathi, A., Ramana, K., Raut, R., & Dhanaraj, R. K. (2022). A review on machine learning strategies for real-world engineering applications. *Mobile Information Systems*, *2022*.

[5] Wang, J., Lu, S., Wang, S. H., & Zhang, Y. D. (2022). A review on extreme learning machine. *Multimedia Tools and Applications*, *81*(29), 41611-41660.

[6] Ding, S., Xu, X., & Nie, R. (2014). Extreme learning machine and its applications. *Neural Computing and Applications*, *25*, 549-556.

[7] Zhang, J., Ding, S., Zhang, N., & Shi, Z. (2016). Incremental extreme learning machine based on deep feature embedded. *International Journal of Machine Learning and Cybernetics*, *7*, 111-120.

[8] Huang, G. B. (2015). What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. *Cognitive Computation*, *7*, 263-278.

[9] Zhang, J., Li, Y., Xiao, W., & Zhang, Z. (2020). Non-iterative and fast deep learning: Multilayer extreme learning machines. *Journal of the Franklin Institute*, *357*(13), 8925-8955.

[10] de Campos Souza, P. V., Araujo, V. S., Guimaraes, A. J., Araujo, V. J. S., & Rezende, T. S. (2018, November). Method of pruning the hidden layer of the extreme learning machine based on correlation coefficient. In *2018 IEEE Latin American conference on computational intelligence (LA-CCI)* (pp. 1-6). IEEE.

[11] Xiao, D., Li, B., & Mao, Y. (2017). A multiple hidden layers extreme learning machine method and its application. *Mathematical Problems in Engineering*, *2017*.

[12] Liu, J., & Le, B. T. (2019). Incremental Multiple Hidden Layers Regularized Extreme Learning Machine Based on Forced Positive-Definite Cholesky Factorization. *Mathematical Problems in Engineering*, *2019*.

[13] Rodrigues, I. R., Neto, S. R. D. S., Kelner, J., Sadok, D., & Endo, P. T. Convolutional extreme learning machines: a systematic review. Informatics 8 (2), 33 (2021).

[14] Ding, S., Zhao, H., Zhang, Y., Xu, X., & Nie, R. (2015). Extreme learning machine: algorithm, theory and applications. *Artificial Intelligence Review*, *44*, 103-115.