-RESEARCH PAPER-

International Journal of Mining and Geo-Engineering IJMGE

IJMGE 59-1 (2025) 31-42 DOI: 10.22059/IJMGE.2025.377703.595173

An algorithm to predict shear wave velocity using well log data and deep learning algorithms

Farhad Mollaei [a] , Ali Moradzadeh [a, *] and Reza Mohebian [a]

[a] *School of Mining, College of Engineering, University of Tehran, Tehran, Iran.*

Article History:

Received: 30 June 2024.
Revised: 17 December 2024.
Accepted: 18 January 2025.

ABSTRACT

Shear wave velocity (Vs) is one of the most critical parameters for determining geomechanical properties to predict reservoir behavior. Determining shear wave velocity (Vs) through methods, such as core analysis requires a significant amount of time and cost. Additionally, due to the scarcity of core samples and the heterogeneity of reservoir rocks, determining this parameter using conventional methods is often not very accurate. While many empirical methods have been developed for estimating Vs, their applicability across different regions is often limited. Therefore, estimating Vs using conventional well logs is crucial. An efficient method for predicting Vs is the use of intelligent algorithms, which offer low-cost and accurate predictions. It is feasible to predict Vs using well log data. In this study, Vs was predicted using empirical relations and some deep learning (DL) algorithms in one of the hydrocarbon fields in southern Iran. In order to use the DL methods, the autoencoders deep network was used to select the effective features in predicting the Vs, and then, with multi-layer perceptron (MLP), long-short term memory (LSTM), convolutional neural network (CNN), and convolutional neural network + long-short term memory (CNN+LSTM) networks, Vs was predicted. The performance of these models was tested by a blind data set that the models had not seen before. Furthermore, the results were checked and evaluated by set of statistical measures, including MAE, MAPE, MSE, RMSE, NRMSE, and R^2 values calculated for train, test, and blind datasets. It was found that all four deep learning models used in this study well performed effectively for Vs prediction, but the combined CNN+LSTM model results indicated that the least root mean squared error (RMSE) was equal to 0.0243 (2.43%) and the best coefficient of determination (R^2) equal to 0.9993 for blind dataset. We found that Vs can be predicted from a series of well log data by considering their variation trends and context information with depth by means of DL algorithms. This approach is particularly suitable for problems involving various series data, such as Vs prediction. By comparing the results obtained from DL algorithms with those from conventional empirical methods and processing real petrophysical well log data, it can be concluded that deep learning algorithms not only offer more predictive accuracy and robustness but also hold promising use prospects in Vs prediction studies. The results showed that the used CNN and CNN+LSTM networks, as new deep learning algorithms, are able to predict Vs adequately.

Keywords: *Shear wave velocity (Vs), Well log data, Empirical relations, Deep learning algorithm.*

1. Introduction

One of the most important sources of information for determining the properties of rocks in subsurface formations are well log data. The velocity of seismic waves along with other petrophysical information is a suitable tool to determine the properties of rocks and underground fluids. Vs is a key parameter in geophysical and geomechanical studies and provides valuable information for determining pore pressure, type of fluids in pores and determining mechanical properties of reservoir rocks. Vs is measured with DSI tools or laboratory measurements. In many wells, especially old wells, DSI tools are not used due to their high cost. Additionally, laboratory testing of cores is laborious and expensive and since the number of cores in wells is limited and discontinuous, it cannot express the properties of underground rocks accurately. Therefore, it would be ideal to use a method that can continuously predict Vs for a well in a short period of time and lower cost. Many methods have been introduced to predict the Vs that use petrophysical logs. One of these methods is empirical relations, the most important of which are the relations of Castagna et al. [1], Eskandari et al [2] and Coello [3]. Common empirical relations are presented in Table 1.

One of the efficient methods for predicting Vs is the use of intelligent methods. DL is a subset of machine learning (ML) that allows computers to tackle complex problems by employing neural networks with an increased numbers of neurons, layers, and interconnected nodes. DL is the knowledge that should be given to the computer so that it can interact with its environment based on a series of rules. Benigo [4] believes that deep learning makes the computer learn abstract concepts. The basis of deep learning is that computers, like humans, have a different understanding of concepts and have the power to generalize. The general classification of DL is as follows [5]: supervised learning, unsupervised learning and hybrid learning networks.

In supervised learning, the categories of data are known from the beginning, and each training data is assigned to a specific category. In other words, during training, information in addition to the training data (the same as the title of each category from which the title of the supervisor) is provided to the learner. But in unsupervised learning, no information except the educational data is available to the learner, and it is the learner who must look for a specific structure in the data [6]. DL algorithms are a subset of ML algorithms designed to uncover multiple levels of distributed representations within input data [7].

* Corresponding author. *E-mail address:* a_moradzadeh@ut.ac.ir (A. Moradzadeh).

Most modern DL models utilize artificial neural networks (ANNs), though they may incorporate propositional formulations or organized layers of hidden variables in generative models such as nodes in deep belief networks and deep Boltzmann machines. In DL, each level learns to transform its input data into a progressively more abstract and composite representation. Crucially, DL processes autonomously determine the optimal features to place at each level.

Table 1. Common experimental formulas used to predict Vs.

References	Equations
Castagna et al., 1985	$V_s = 1.0168V_p - 0.05509V_p^2 - 1.0305$
Eskandari et al., 2004	$V_s = 1.612V_p - 0.1236V_p^2 - 2.0357$
Coello, 2007	$V_s = 1.09913326V_p^{0.9238115636}$

In recent years, the use of intelligent methods in predicting oil engineering problems has increased significantly. Eskandari et al. [2] employed multiple regression and ANNs to estimate Vs. Rezaei et al. [8] predicted Vs using fuzzy logic, ANNs, and adaptive neuro-fuzzy inference systems. Rajabi et al. [9] used intelligent methods to predict the Vs and Stoneley waves based on well log data. Motazadian et al. [10] used a genetic algorithm to predict Vs and compression wave velocity (Vp). Asoudeh and Bagheripour [11; 12; 13] used ANNs, fuzzy logic and neuro-fuzzy to predict Vs. Maklay et al. [14] used the combination of support vector regression and neural networks to predict Vs. Gholami et al. [15] predicted Vs using MLP. Akhondi et al. [16] predicted Vs using ANN. Kadkhodaei [17] used the ant colony optimization algorithm to estimate the Vs. Bagheripour et al. [18] used support vector regression to predict Vs. Al-Dossari et al. [19] estimated Vs using generalized reduced gradient. Singh and Kanli [20] predicted Vs using feedforward ANNs. Behnia et al. [21] estimated Vs with Gene expression programming and adaptive neuro-fuzzy inference. Shiroudi et al. [22] estimated Vs with committee fuzzy machine, surgeon's fuzzy inference, adaptive neuro-fuzzy inference and optimized fuzzy inference. Mehrghini et al. [23] estimated Vs with multi-layer perceptron and Elman neural network. Alkinani et al. [24] estimated Vs with nonlinear autoregressive network with exogenous inputs network. Zhang et al. [25] estimated Vs with long short-term memory (LSTM). Wood et al. [26] predicted Vs with the transparent open box learning network. Gholami et al. [27] predicted Vs with optimized fuzzy inference, optimized neural network, optimized support vector regression and committee machine. Wang et al. [28] estimated Vs with particle swarm optimization- long short-term memory, long short-term memory, recurrent neural network). Ghorbani et al. [29] predicted Vs with Least-squares support-vector machines-genetic algorithm. Olayiwola and Sanuade [30] estimated Vs with ANN, Least-squares support-vector machines and adaptive neuro-fuzzy inference. Ebrahimi et al [31] estimated Vs using three different ML methods named as: MLP- ANN, adaptive neuro-fuzzy inference system and multi-gene genetic programming. Gholami et al [32] using ANNs, fuzzy logic, and neuro-fuzzy estimated Vs. Nasrnia et al [33] used intelligent methods, including neural network, fuzzy logic, and an adaptive neuro-fuzzy inference system.

Given the significant importance of Vs in geomechanical and petrophysical modeling, and the high cost of log DSI, a method that can predict Vs with high accuracy is crucial. One of the methods used is intelligence methods. According to previous researche, many methods have been developed to estimate Vs, but because accurate estimation of these parameters is very important for modeling and geomechanical studies, this paper has introduced algorithms for estimating Vs using DL algorithms and well log data with high accuracy and low cost. On this basis, four DL algorithms are used to predict Vs (multilayer perceptron network, transformed convolutional neural network, recurrent neural network (RNN) and convolutional neural network + RNN). The aim was to establish a model capable of estimating Vs based on conventional petrophysical logs. For this study, a dataset was selected from a vertical well in a hydrocarbon field located in southern Iran. To reach the goal, the auto-encoders deep network was first used to select the effective features. The selected logs were then used as input parameters for the model to predict Vs. Subsequently, using the designed MLP, LSTM, CNN, and CNN+LSTM deep learning algorithms the values of Vs was predicted and compared with those obtained by empirical models.

2. Material and methods

2.1. Methods

2.1.1. Multilayer perceptron neural network

Multi-layer perceptron ANNs are a type of feedforward neural network capable of making accurate estimates by appropriately selecting the number of layers and neurons. In this type of network, the adjustable parameters are the weights of connections between layers. The training process aimed to find the optimal values for these weights, which enable effective communication between neurons within the network. The most common learning algorithm for this feedforward network is the backpropagation algorithm. This type of network typically consists of three layers: the input layer, hidden layers, and the output layer. In feedforward networks, neurons are arranged in layers that start at the first (input) layer. In a neural network structure, a set of neurons is interconnected, where each neuron is connected to all neurons in the next layer. The connections between the input, hidden, and output layers are established through weights and biases, which are parameters of the MLP network [34]. To train MLP neural networks, different learning algorithms are used, such as gradient backpropagation algorithm, gradient reduction algorithm, Bayesian regularization algorithm and Levenberg-Marquardt algorithm. The choice of each algorithm affects the learning time and accuracy of the network. Figure 1 shows a schematic of a multilayer perceptron neural network.

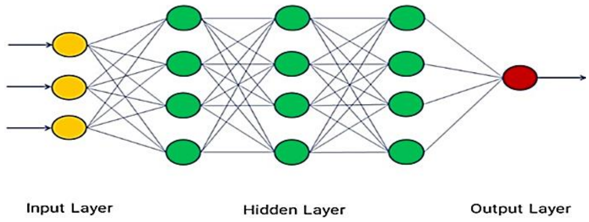


Figure 1. A prototype diagram of the MLP neural network [34].

2.1.2. Recurrent neural network

A recurrent neural network (RNN) is a type of ANN where the links between nodes form a directed graph along a time sequence, enabling the algorithm to exhibit temporal dynamics (Figure 2).

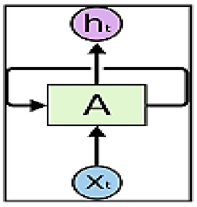


Figure 2. Schematic diagram of the RNN (http://colah.github.io).

RNNs have loops. In Figure 2, section A receives the x_t value as an input and outputs the h_t value. The loop causes information to be sent from one stage to the next. The performance of RNN is not significantly different from that of normal neural networks. RNN can be conceptualized as multiple identical repetitions of a neural network, where each repetition passes its information to the next network [35]. In Figure 2, the state of the recurrent neural network can be observed if the loop depicted in Figure 3 is opened.

In the past years, these networks have been used, which have led to remarkable successes in various fields. Many of these successes are attributed to the use of LSTM networks, which are a special type of RNN that in most cases have a better performance than standard recurrent neural networks. It can be said that most of the successes of recurrent neural networks were achieved when LSTM networks were used. LSTM networks are designed to capture and learn long-term dependencies. They were first presented by Hochreiter and Schmidhuber [35] with the primary goal of addressing issues related to long-term dependencies in traditional RNNs. One of the key attributes of the LSTM networks is their inherent ability to retain information over long intervals, which is achieved through their unique structure. Unlike standard recurrent neural networks, LSTM networks incorporate specialized units that facilitate effective memory retention across distant time steps. This structural feature enables LSTM networks to excel in tasks requiring the capture of long-term dependencies. In standard RNNs, the recurrent units typically consist of simpler structures, such as a single hyperbolic tangent layer (tanh) in the hidden layer (Figure 4) [36].

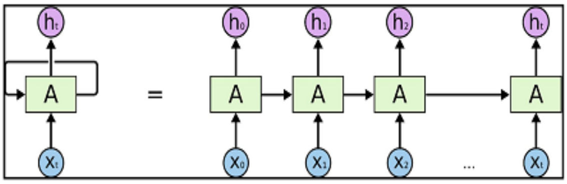


Figure 3. Recurrent neural network shown in Figure 2 (http://colah.github.io).

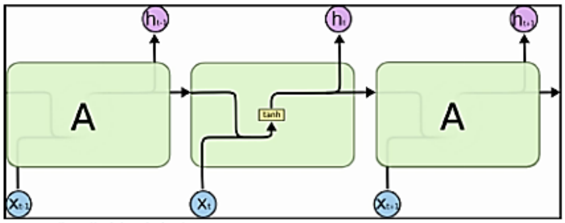


Figure 4. Recurrent units in standard recurrent neural networks (http://colah.github.io).

LSTM networks also exhibit a sequence-like structure, but their repeating unit differs significantly. Instead of comprising a single neural network layer, LSTM units consist of four interacting layers that communicate with each other according to a specialized architecture (Figure 5).

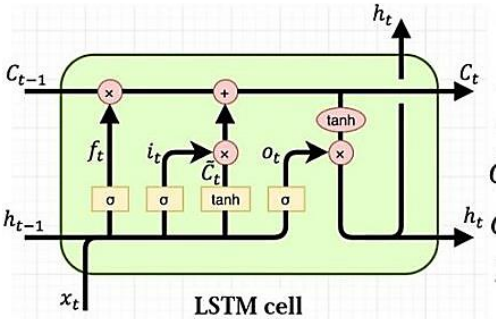


Figure 5. Repetitive units in the LSTM network, each hidden layer of which is four layers interacting with each other (http://colah.github.io).

2.1.3. Convolutional Neural Network

This method was introduced in the 1980s and 1990s. Although CNN was forgotten for a while, since 2012, and with the modification of this network, it has made great progress in most of the machine field and is expanding rapidly. The CNN is distinct from traditional deep neural networks with multiple hidden layers. Instead, CNNs are designed to mimic the processes of the visual cortex, allowing them to effectively identify and process images. This characteristic highlights a significant departure in how CNNs operate compared to earlier neural network models. Before the advent of CNNs, feature extraction was typically performed by researchers in various fields, requiring substantial effort and time. However, the performance of these manually designed feature extractors was often inconsistent and not always optimal. Importantly, these traditional feature extractors operated independently of ML techniques. In contrast, CNNs integrate feature extraction directly into their training process, eliminating the need for manual design. The feature extraction in CNNs is achieved through layers of neural networks whose weights are adjusted during training, optimizing their ability to automatically learn and extract relevant features from input data. A typical CNN consists of three layers; convolutional, pooling, and fully connected layers. Figure 6 illustrates a generalized schematic of a CNN for image classification, showcasing these layers and their functions. The convolution layer employs filters to perform convolution, followed by a Rectified Linear Unit (ReLU) layer for element-wise operations, resulting in a smoothed feature map. The pooling layer then decreases the dimension of the feature map through downsampling, flattening it into a linear vector. Finally, the fully connected layer classifies and identifies images. Figure 7 illustrates a general CNN structure, featuring parallel filters acting on input data to extract various features. The output vectors from each filter layer are concatenated and processed through a dense layer, akin to a multilayer perceptron neural network. This dense layer comprises neurons whose number is determined through trial and error or optimization. The model is executed to find weights and biases for the neurons, thereby optimizing the accuracy of predictions for the dependent variable.

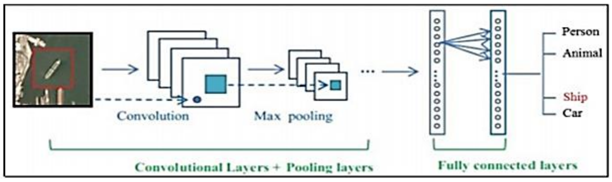


Figure 6. General scheme of a convolutional neural network and its layers [37].

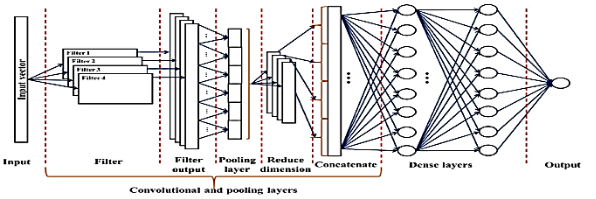


Figure 7. A sample diagram of a deep learning CNN structure [38].

In each convolutional neural network, there are two stages for training: the forward step and the backward or backpropagation step. In the forward step, the image is passed through the network and its feature vector is extracted. Once the input image has been classified into one of the output classes, a loss function is defined to quantify the error between the predicted and actual classes. The weights of the filters (or kernels) in the convolutional layers are adjusted to minimize this loss function. The process of optimizing these weights involves using gradients in iterative steps, aiming to progressively improve the network's ability to accurately classify images. The gradients are calculated using the recursive law along the layers, and the weights and parameters are updated according to the gradient [39]. In the initial stage, the input image is fed into the network. At this stage, each neuron's parameters undergo pointwise multiplication with the input, followed by the application of the convolution operation within each layer. Subsequently, the network computes its output based on these operations.

2.2. Data presentation

The Zagros basin, in the southwest of Iran and the east of Iraq, hosts many of the world's largest hydrocarbon fields, which were created by the continuous and long-term convergence between the Arabian and Eurasian plates during the closure of the Neotetis oceanic basin. The studied field is one of Iran's oil fields, which was discovered by Iran Oil Exploration and Production Company, and the length of this field is about 39 km and its width is 5 km. This field includes two oil reservoirs, Asmari and Bangestan. The Asmari reservoir of this field is 30 km long and 3 km wide [27]. From the point of view of geology, the Asmari formation is divided into 8 layers, and according to the contact surface of water and oil, only layers 1, 2 and 3 are located in the oil zone of the reservoir. Bangestan reservoir consists of two formations, Ilam and Sarvak. The Ilam formation consists of 3 layers, the 3rd layer is its production. Sarvak formation has 2 production layers 4 and 6 and layer 5 does not participate in production. The total available well-logging data is 77319, which are located in the depth range of 305.26 to 2977.29 meters. Available well logs are density (RHOB), caliper (CHAL), neutron (NEUT), laterolog7 (LL7), microlaterolog (MLL), photoelectric (PEF), primary and secondary velocity (Vp and Vs), and gamma ray (GR).

2.2.1. Feature selection to reduce dimension

The correlation between Vs and conventional well logs presents a complex nonlinear system problem, influenced by various elements within the Earth's system. This matter complicates the analysis and prediction of Vs. Choosing to use the most relevant logs instead of all existing conventional logs for model training and prediction can streamline data processing, enhance processing speed, and improve model efficiency [40]. Furthermore, feature selection enhances the accuracy of estimation and the general applicability of the model. In other words, incorporating all possible factors as model inputs would introduce numerous variables, complicating the network structure and potentially reducing estimation precision. Therefore, to simplify the model structure, enhance modeling capabilities, improve prediction efficiency, and mitigate the impact of non-critical variables on prediction results, feature selection is essential. In this paper, using Auto-encoder algorithm, the effective features and effective logs were determined to predict Vs.

The data of RHOB, CHAL, NEUT, LL7, PEF, Vp, Vs, MLL and GR were available and were determined by applying the Auto-encoder's deep learning algorithm to the main features and logs. CHAL, RHOB, Vp, GR, and NEUT were selected as inputs for the MLP, LSTM, CNN, and CNN+LSTM models. Finally, four logs were determined as the main features and input of the algorithm, because selecting five features according to the deep learning algorithm, led to increased error and decreased accuracy. Figure 8 shows the workflow of feature selection using deep learning Auto-encoder algorithm.

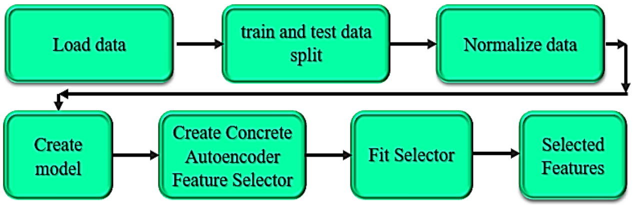

Load data
train and test data split
Normalize data
Create model
Create Concrete Autoencoder Feature Selector
Fit Selector
Selected Features

Figure 8. Display the workflow of feature selection using the Auto-encoder algorithm.

2.2.2. Dataset division

The reliability of prediction stands out as a primary concern in the performance evaluation of supervised DL algorithms. (Consonni et al. 2010; Alsina et al. 2017). In this study, 77319 data points were available in the depth range of 305.26 to 2977.29 meters. At first, 19% of these data were separated from the end of the dataset as blind data and from the other data, 80% were taken as training data and 20% as testing data. To avoid over fitting, a validation split 0.1 of the training data is considered.

2.2.3. Normalizing of data

To reduce errors arising from variances in the amount of input data, the original data must undergo preprocessing in the experiment. In this paper, the Min-Max Normalization method is employed to scale the original data to the [0,1] range. This normalization process also helps eliminate dimensional discrepancies among the input variables.

3. Model evaluation

The efficiency of the DL models for Vs prediction was evaluated using several commonly used statistical criteria stated in Eqs. 1-6. Here, Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), Normalized RMSE (NMSRE), and Coefficient of Determination (R^2) were used to evaluate the performance of the model predictions.

These criteria include: Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), Normalized Root Mean Squared Error (NMSRE), and Coefficient of Determination (R^2). These metrics were employed to assess and compare the accuracy and reliability of the model predictions.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Z_{mesured} - Z_{predict})^2 \tag{1}$$

$$RMSE = \sqrt{MSE} \tag{2}$$

$$NRMSE = \frac{RMSE}{MAX(Z_{mesured}) - MIN(Z_{mesured})} \tag{3}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Z_{mesured} - Z_{predict}\right| \tag{4}$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{Z_{mesured} - Z_{predict}}{Z_{mesured}}\right| \tag{5}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Z_{mesured} - Z_{predict})^2}{\sum_{i=1}^{n}(Z_{mesured} - Z_{average})^2} = 1 - \frac{MSE}{\sigma^2} \tag{6}$$

4. Results and discussion

In this research, data were collected to predict Vs, which include logs RHOB, CHAL and NEUT, LL7, PEF, Vp, Vs, MLL, and GR. Figure 9 shows the workflow of shear wave velocity prediction using the deep learning method.

4.1. Empirical models for VS prediction

In this section, the common models of Castagna, Eskandari, and Coello were reviewed, and their relations are illustrated in Table 1. Table 2 presents the values of MAPE, MAE, MSE, RMSE, NRMSE, and R^2. Figure 10 shows an assessment of measured and predicted Vs for the presented empirical models. According to Table 2, Figure 10 and by observing the errors and the coefficient of determination, it can be stated that the Castagna et al. model had less error and a higher coefficient of determination in Vs prediction than other models.

4.2. Deep learning algorithms for Vs prediction

As stated previously, 77319 samples have been recorded in the depth range of 305.26 to 2977.29 meters from logs in one of the wells of one of the hydrocarbon fields of Iran. At first, 19% (1634 sample) of this dataset was separated as blind data, and from the remaining part, 80% (5565 sample) of the 6957 recorded data were selected as training and 20% (1392 sample) as testing data for evaluation. To predict the Vs using deep learning models, the data of logs, CHAL, RHOB, Vp, GR, and NEUT were selected by the Auto-encoder deep learning algorithm as model inputs for MLP, LSTM, CNN and CNN+LSTM models.

Table 2. Vs prediction errors and accuracy for all data records using empirical models.

Empirical Models	MAE	MAPE	MSE	RMSE	NRMSE	R²
Castagna et al.	0.2520	11.6562	0.0786	0.2804	0.0837	0.8529
Eskandari et al.	0.2749	14.9621	0.0919	0.3031	0.0905	0.8282
Coello	0.3020	19.1391	0.1000	0.3163	0.0944	0.8130

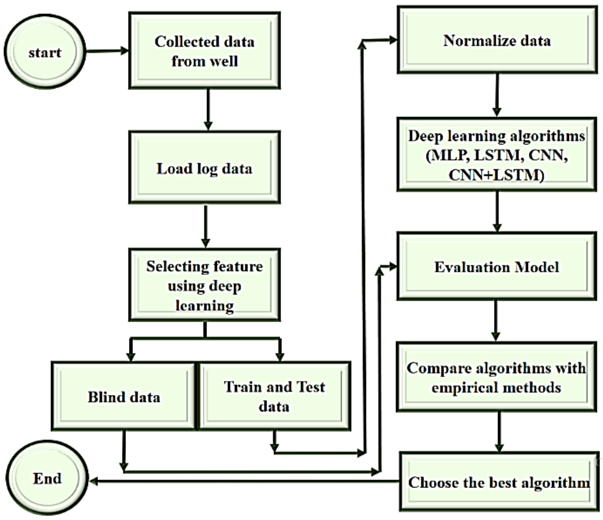

start
Collected data from well
Load log data
Selecting feature using deep learning
Blind data
Train and Test data
End
Normalize data
Deep learning algorithms (MLP, LSTM, CNN, CNN+LSTM)
Evaluation Model
Compare algorithms with empirical methods
Choose the best algorithm

Figure 9. The workflow schematic for VS prediction.

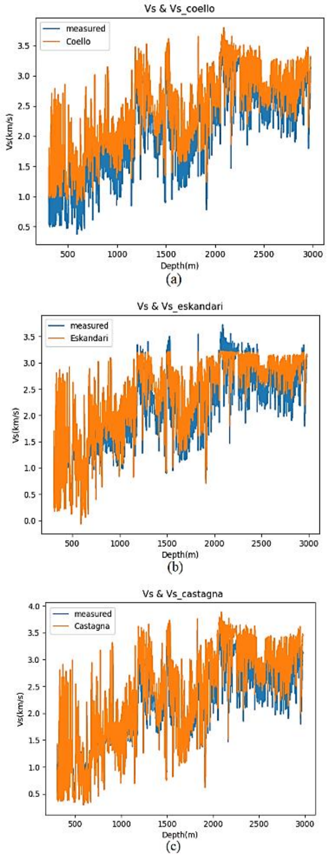


Figure 10. Comparison of the measured Vs (blue color) values and those obtained from the empirical relations (orange color). (a) Comparison of the measured Vs (blue color) values and those obtained from Coello relation. (b) Comparison of the measured Vs (blue color) values and those obtained from the Eskandari et al relation. (c) Comparison of the measured Vs (blue color) values and those obtained from the Castagna et al. relation.

When constructing the MLP network for this purpose, the key parameters were set as follows: batch_size = 50, the learning rate= 0.0001, the number of iterations was 100, two hidden layers (with 500 nodes in the first layer and 300 nodes in the second layer), validation_split=0.1, the dense layer was set to 1, the activation function was the ReLu function and the Adam gradient descent algorithm was used for weight and bias updates.

For building the LSTM network: batch size = 512, learning rate = 0.01, the number of iterations = 150, two hidden layers (with 100 nodes in the initial layer and 200 nodes in the second layer), validation split = 0.1, the Adam optimization function, and one dense layer.

In the case of the CNN model, which aimed to predictV_S, three layers were employed. The number of filters in the initial layer was fixed at 64, in the second layer at 128, and in the third layer at 256. A kernel size of 2, padding set to 'same', strides set to 1, batch size = 128, learning rate = 0.0001, number of iterations = 100, validation split = 0.1, ReLU activation function, and one dense layer with Adam as the optimization function.

For the CNN+LSTM model, which included two CNN and two LSTM layers, the initial CNN layer had 64 filters, the second had 128 filters, a kernel size of 2, padding set to 'same', and strides set to 2. In the LSTM network, two hidden layers were utilized with 50 nodes in the initial layer and 100 nodes in the second layer. Additionally, a dropout rate of 0.2, Adam as the optimization function, learning rate of 0.01, batch size of 512, 150 iterations, and validation split of 0.1 were employed and one dense layer.

Tables 3 displays Vs prediction errors and accuracies, respectively based on the training (80%) subsets. Figure 11 shows the train MSE (blue color) and validation MSE (orange color) error with MLP, LSTM, CNN, and CNN+LSTM algorithms for training data. Also, Figure 12 shows the train MAE (blue color) and validation MAE (orange color) error for four algorithms. According to Figures 11, 12 and Table 3, for Vs training data, four algorithms had a low error, where the MAE and MSE values were equal to $MAE_{MLP} = 0.0009$ $MAE_{LSTM} = 0.0094$ $MAE_{CNN} = 0.0054$ $MAE_{CNN+LSTM} = 0.0009$, and $MSE_{MLP} = 1.3778e - 06$ $MSE_{LSTM} = 0.0001$ $MSE_{CNN} = 6.8188e - 05$ $MSE_{CNN+LSTM} = 1.3778e - 06$, respectively.

Table 3. Vs Prediction errors for training data.

Models	MAE	MAPE	MSE	RMSE	NRMSE	R²
MLP	0.0009	0.0365	1.3778e-06	0.0011	0.0008	0.9999
LSTM	0.0094	0.4162	0.0001	0.0113	0.0077	0.9985
CNN	0.0054	0.2125	6.8188e-05	0.0082	0.0056	0.9992
CNN+LSTM	0.0009	0.0365	1.3778e-6	0.0011	0.0008	0.9999

Figure 13 displays a comparison of the predicted Vs and measured Vs for training data, blue log (Vs measured for training data), orange log (Vs predicted for training data). Figure 14 shows the coefficient of determination of the training data of the measured Vs and the predicted Vs for four algorithms. As seen, the R^2 values for training data were equal to $R^2_{MLP} = 0.9999$ $R^2_{LSTM} = 0.9985$ $R^2_{CNN} = 0.9992$, $R^2_{CNN+LSTM} = 0.9999$. According to Table 3 and Figures 11, 12, 13, and 14 it can be said that all four observed algorithms had very low error and high coefficient of determination for training data.

Tables 4, displays the Vs estimation errors and accuracies based on the testing data (20%) subset for four algorithms. Figure 15 provides a comparison of the predicted and measured Vs values using four algorithms for testing data. According to Figure 15 and Table 4, for Vs testing data, the MAE and MSE values of each algorithm were equal to

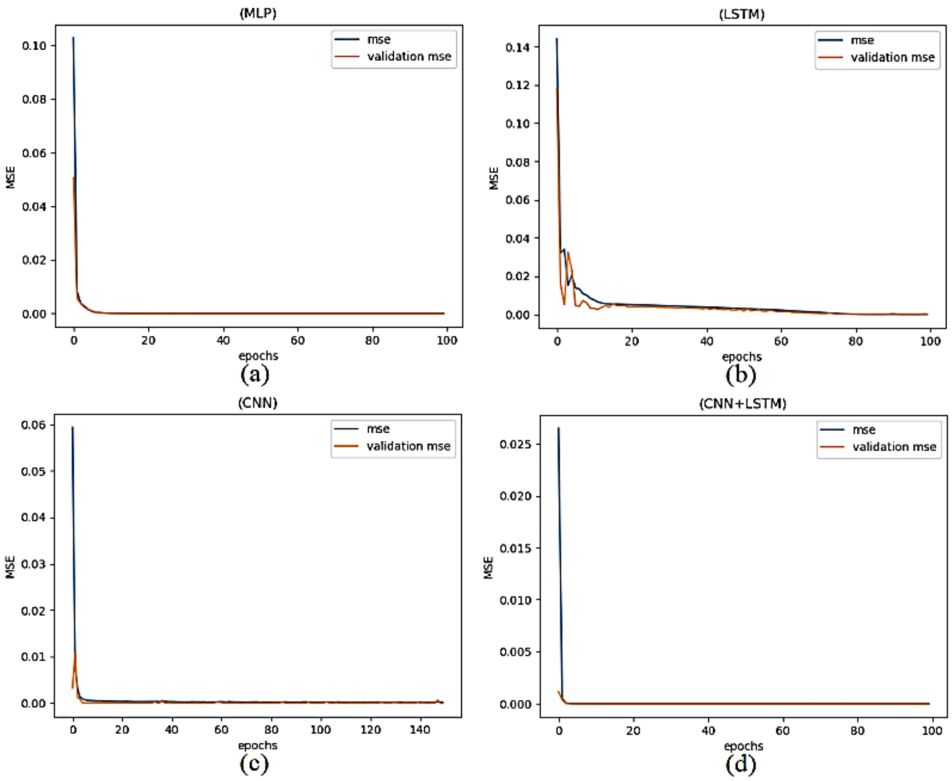


Figure 11. Display of train MSE (blue color) and validation MSE error (orange color) for four algorithms. (a), MLP algorithm. (b), LSTM algorithm. (c), CNN algorithm. (d), CNN+LSTM algorithm.

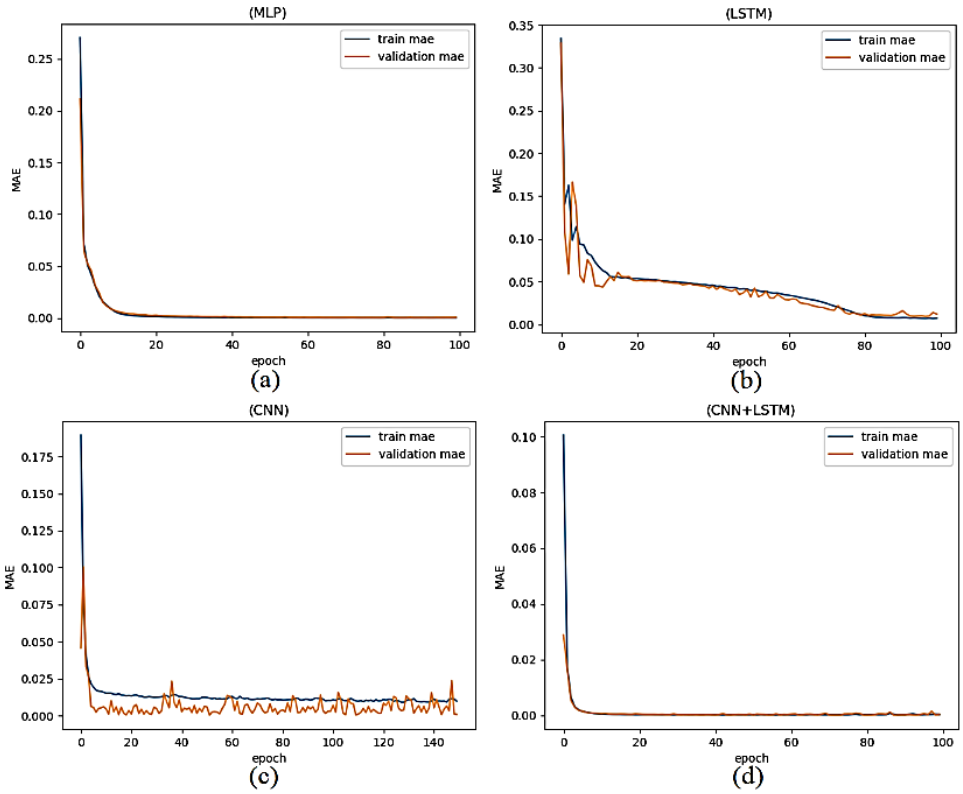


Figure 12. Display of train MAE (blue color) and validation MAE error (orange color) for four algorithms. (a), MLP algorithm. (b), LSTM algorithm. (c), CNN algorithm. (d), CNN+LSTM algorithm.

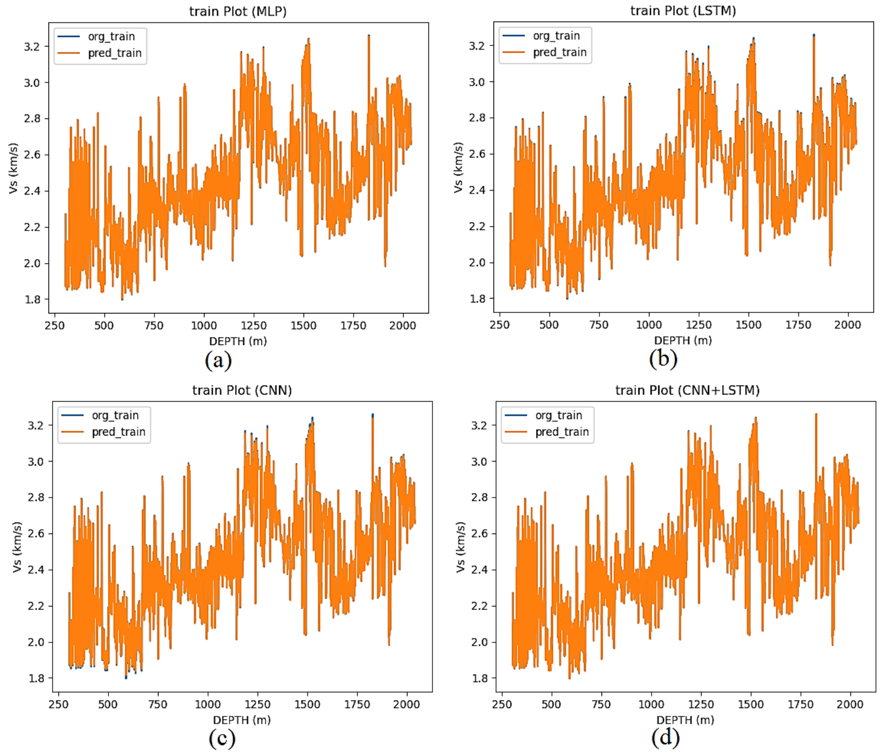


Figure 13. A comparison of the predicted Vs with those of measured Vs data for four algorithms of training data. (a), Vs prediction with the MLP algorithm. (b), Vs prediction with the LSTM algorithm. (c), Vs prediction with the CNN algorithm. (d), Vs prediction with the CNN + LSTM algorithm. Blue log (Vs measured), orange log (Vs predicted).

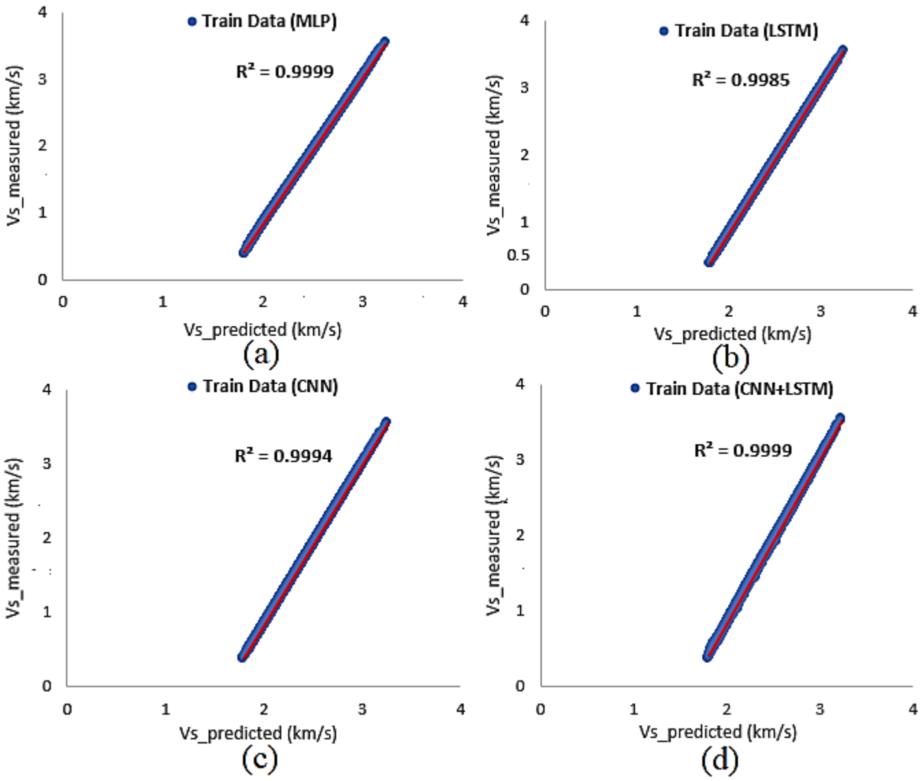
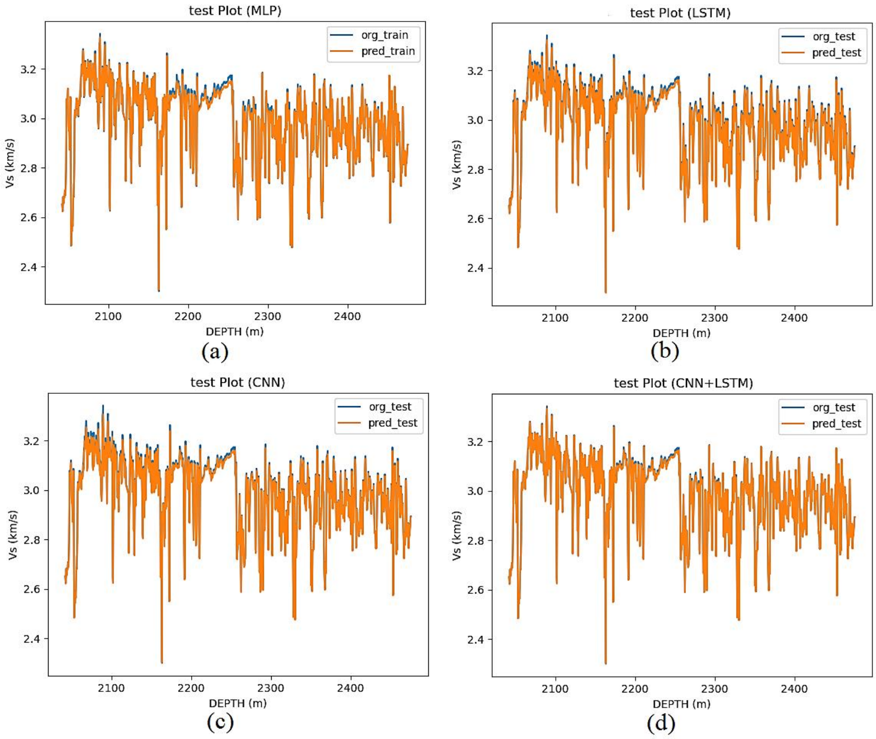


Figure 14. Comparison of the coefficient of determination (R2) of training data for measured and predicted Vs using four algorithms. (a) R2 of the MLP algorithm. (b) R2 of the LSTM algorithm. (c) R2 of the CNN algorithm. (d) R2 of the CNN+LSTM algorithm.

Table 4. Vs Prediction errors and accuracy for testing data.

Models	MAE	MAPE	MSE	RMSE	NRMSE	R^2
MLP	0.0185	0.6093	0.0005	0.0235	0.0225	0.9745
LSTM	0.0082	0.2756	8.8125e-5	0.0093	0.0090	0.9959
CNN	0.0052	0.1699	6.8915e-5	0.0083	0.0079	0.9968
CNN+LSTM	0.0023	0.0775	1.7100e-5	0.0041	0.0039	0.9993



(a) (b) (c) (d)

Figure 15. Comparison of the predicted Vs with those of measured values using four algorithms of testing data. (a) Vs prediction for testing data using MLP algorithm. (b) Vs prediction using the LSTM algorithm. (c) Vs prediction using the CNN algorithm. (d) Vs prediction using the CNN + LSTM algorithm. Blue log (Vs measured), orange log (Vs predicted).

$MAE_{MLP} = 0.0185$ $MAE_{LSTM} = 0.0082$ $MAE_{CNN} = 0.0052$ $MAE_{CNN+LSTM} = 0.0023$, and $MSE_{MLP} = 0.0005$ $MSE_{LSTM} = 8.8125e - 05$ $MSE_{CNN} = 6.8915e - 05$ $MSE_{CNN+LSTM} = 1.7100e - 05$, respectively.

Figure 16 shows the coefficient of determination values of the testing data for the measured Vs and predicted Vs using four algorithms. According to Figure 16 and Table 4, the coefficient of determination (R^2) values for testing dataset of each algorithm were equal to $R^2_{MLP} = 0.9745$ $R^2_{LSTM} = 0.9949$ $R^2_{CNN} = 0.9968$ $R^2_{CNN+LSTM} = 0.9993$. The results showed that the MLP algorithm had the lowest R^2 value, while the CNN + LSTM algorithm had the highest R^2 value.

Figure 17 provides a comparison of the predicted and measured Vs values for training and testing data using four different deep learning algorithms.

To confirm the efficiency of the algorithms, they were applied to the blind dataset. Tables 5 depicts the predicted Vs errors and the coefficient of determination based on the blind subsets of data, selected from the 1634 data records. The estimated Vs error values for blind dataset were equal to $MAE_{MLP} = 0.0938$ $MAE_{LSTM} = 0.0510$ $MAE_{CNN} = 0.0373$ $MAE_{CNN+LSTM} = 0.0238$ and $MSE_{MLP} = 0.0090$ $MSE_{LSTM} = 0.0028$ $MSE_{CNN} = 0.0019$ $MSE_{CNN+LSTM} = 0.0006$. Figure 18 shows a comparison of the predicted and measured Vs values for the blind dataset of each algorithm.

Figure 19 shows the coefficient of determination of the blind dataset comparing the measured and predicted Vs for four deep learning algorithms. According to Figures 18 and 19, the coefficient of determination for four deep learning algorithms was equal to $R^2_{MLP} = 0.9004$ $R^2_{LSTM} = 0.9689$ $R^2_{CNN} = 0.9782$ $R^2_{CNN+LSTM} = 0.9934$. The results showed that the MLP algorithm had the lowest coefficient of determination, while the CNN + LSTM algorithm had the highest coefficient of determination.

The deep learning algorithm is one of the newest and highly accurate methods for predicting Vs. In this paper, four algorithms (MLP, LSTM, CNN and CNN + LSTM) were used to predict Vs. In this research, the aim was to evaluate the accuracy of these DL algorithms in Vs prediction. The results showed the high accuracy of the CNN and CNN+LSTM algorithms for Vs prediction. In addition, these algorithms offered more advantages than MLP and traditional empirical models in terms of accuracy and robustness in estimation.

5. Conclusion

In this research, some models were established for estimating Vs based on four algorithms; MLP, CNN, LSTM, and CNN+LSTM. To achieve the goal, first, the Auto-encoder algorithm was used to select the effective features, and the effective features from the CHAL, RHOB, Vp, GR, and NEUT logs were selected. In the next step, the model was defined and trained using four algorithms: MLP, LSTM, CNN, and CNN+LSTM. To estimate Vs, the designed structure of the MLP model included two layers with the initial layer containing 500 nodes and the second layer containing 300 nodes. The LSTM model also included two layers with the initial layer containing 100 nodes and the second layer containing 200 nodes and a dropout rate of 0.2. The CNN model consisted of three layers, 64 filters in the initial layer, 128 filters in the second layer, and 256 filters in the third layer. The CNN+LSTM model

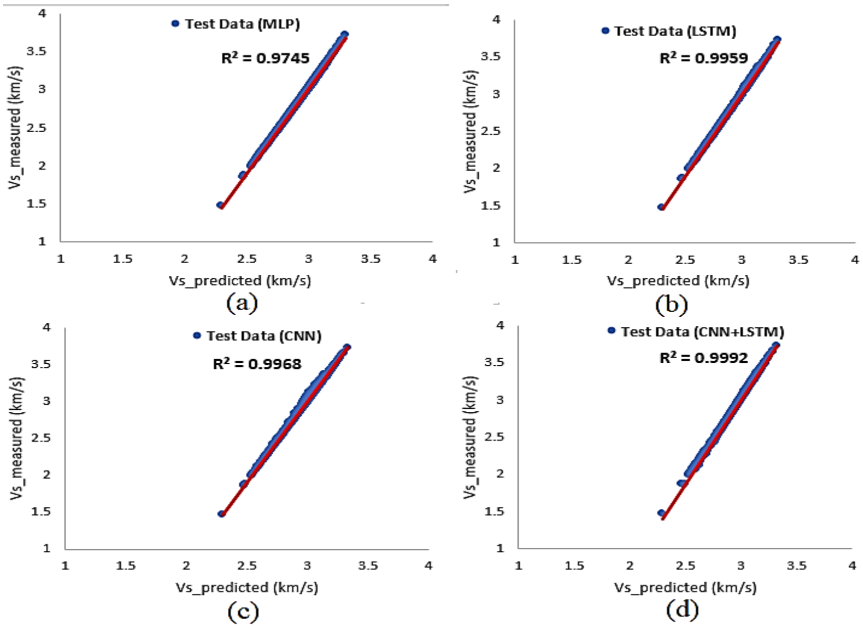


Figure 16. Comparison of the coefficient of determination (R^2) of testing data for measured and predicted Vs using four algorithms. (a) R^2 using the MLP algorithm. (b) R^2 using the LSTM algorithm. (c) R^2 using the CNN algorithm. (d) R^2 using the CNN+LSTM algorithm.

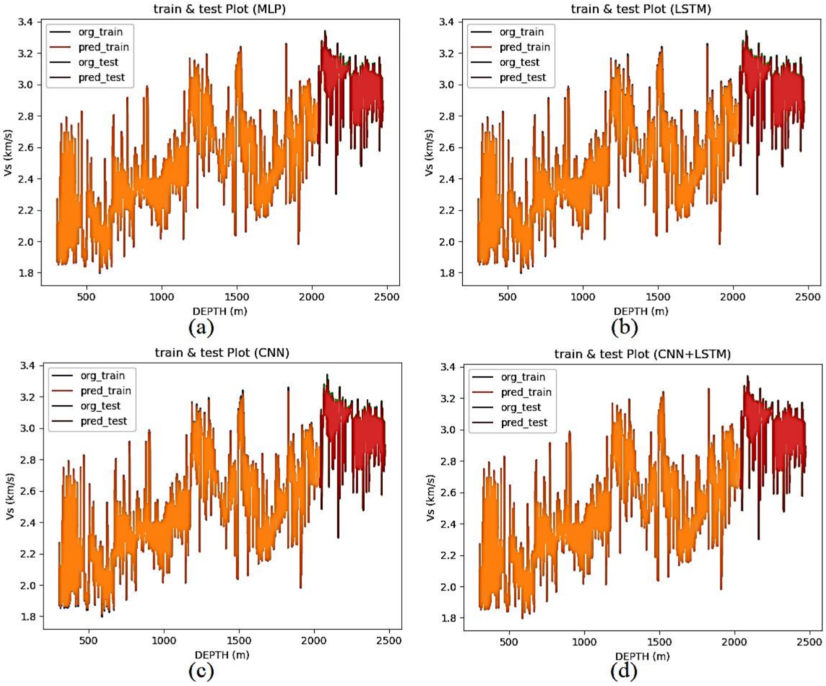


Figure 17. Comparison of the predicted Vs with those of measured values using four algorithms for training and testing data. (a) Vs prediction for training and testing data using the MLP algorithm. (b) Vs prediction using the LSTM. (c) Vs prediction using the CNN. (d) Vs prediction using the CNN + LSTM model.

Table 5. Vs Prediction errors for blind dataset.

Deep Learning models	MAE	MAPE	MSE	RMSE	NRMSE	R^2
MLP	0.0938	3.6227	0.0090	0.0947	0.0612	0.9004
LSTM	0.0510	1.9711	0.0028	0.0529	0.0342	0.9689
CNN	0.0373	1.5226	0.0019	0.0442	0.0286	0.9782
CNN+LSTM	0.0238	0.9286	0.0006	0.0243	0.0157	0.9934

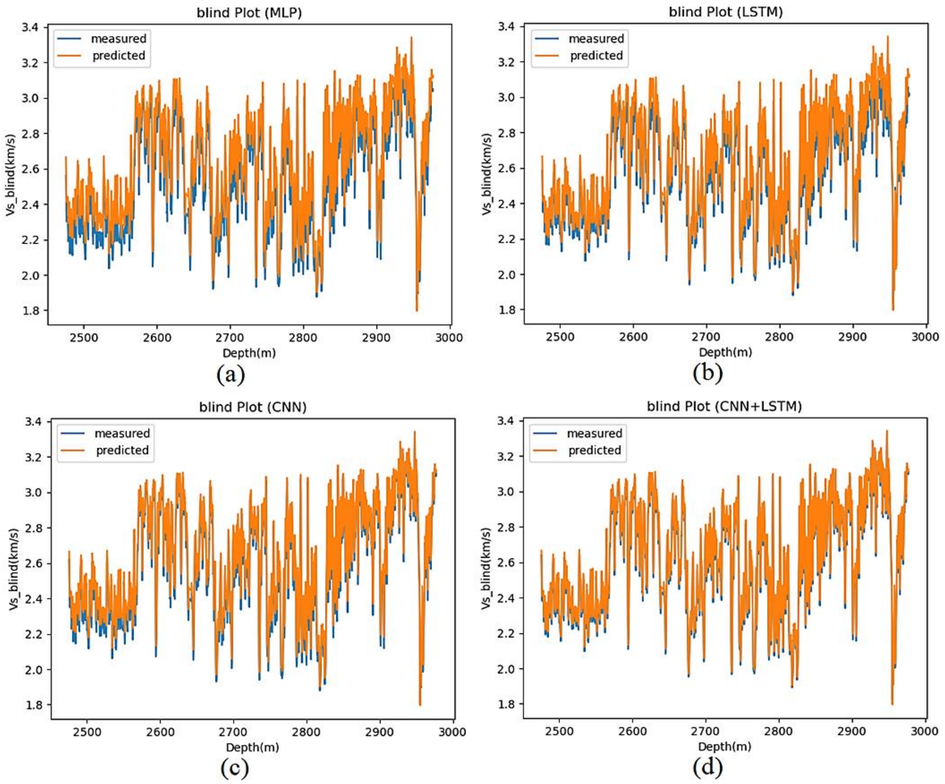


Figure 18. Comparison of the predicted Vs with those of measured values using four algorithms of the blind dataset. (a) Vs prediction using the MLP algorithm. (b) Vs prediction using the LSTM algorithm. (c) Vs prediction using the CNN algorithm. (d) Vs prediction using the CNN + LSTM algorithm.

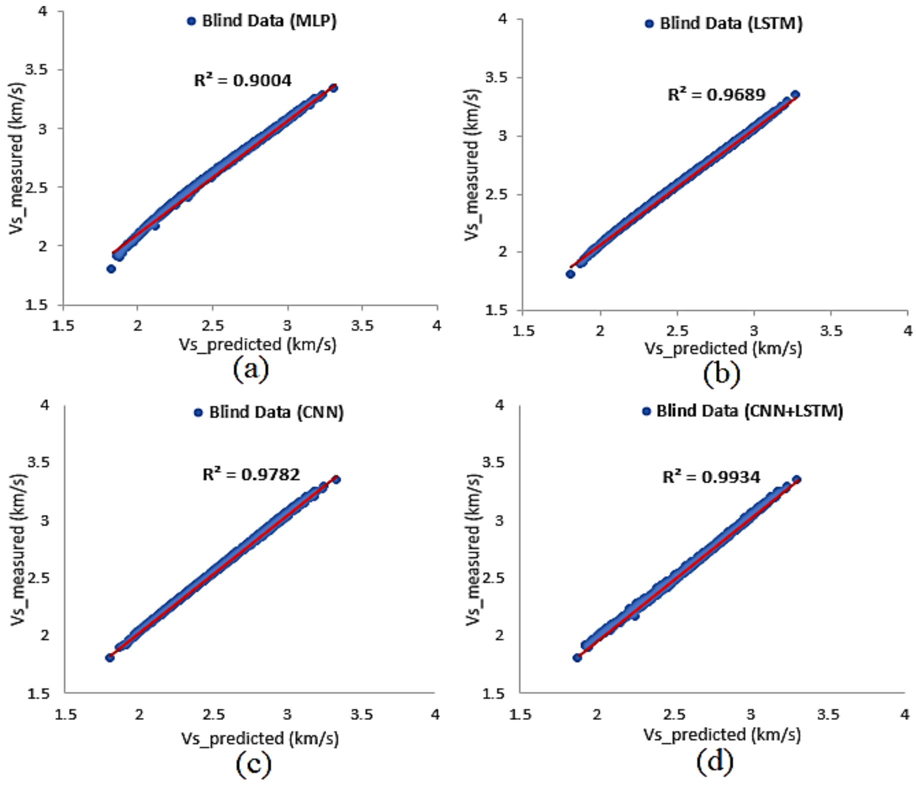


Figure 19. Comparison of the coefficient of determination (R2) of blind data for measured and predicted Vs using four algorithms. (a) R2 using the MLP algorithm. (b) R2 using the LSTM algorithm. (c) R2 using the CNN algorithm. (d) R2 using the CNN+LSTM algorithm.

included two CNN and two LSTM layers; the initial CNN layer had 64 filters, the second had 128 filters, and in the LSTM network, two hidden layers were utilized with 50 nodes in the initial layer and 100 nodes in the second layer. To evaluate the performance and accuracy of the algorithms, the parameters MAE, MAPE, MSE, RMSE, and R^2 were used. In the next step, to ensure the results of the algorithms and generalize prediction results to other areas, these algorithms were applied to blind data, and their Vs error values for blind data were found as: $MSE_{MLP} = 0.0090$ $MSE_{LSTM} = 0.0028$ $MSE_{CNN} = 0.0019$ $MSE_{CNN+LSTM} = 0.0006$ and their R^2 values were found as: $R^2_{MLP} = 0.9004$ $R^2_{LSTM} = 0.9689$ $R^2_{CNN} = 0.9782$ $R^2_{CNN+LSTM} = 0.9934$. The results indicated that acceptable outcomes can be achieved using MLP, LSTM, CNN, and CNN+LSTM models for Vs prediction. However, compared to MLP models, the LSTM, CNN, and CNN+LSTM models demonstrate better stability and a superior ability to predict trend changes. Besides, it was found that these deep learning methods are more accurate than traditional empirical models for Vs prediction. Additionally, the presented methods are highly applicable to well log data gathered from different wells, offering a new approach for research on Vs prediction. These findings reveal that the methods proposed have promising application prospects for Vs prediction. Moreover, it has been found that a combined CNN+LSTM model can improve prediction results.

References

[1] Castagna, J.P., Batzle, M.L., Eastwood, R.L., (1985). Relationships between compressional wave and shear-wave velocities in clastic silicate rocks. Geophysics 50 (4), 571–581.

[2] Eskandari, H., Rezaee, M.R., Mohammadnia, M., (2004). Application of multiple regression and artificial neural network techniques to predict shear wave velocity from wireline log data for a carbonate reservoir, South Iran. Can. Soc. Explor. Geophys. (CSEG) Rec. 29, 42–48.

[3] Coello, C.C., (2007). Evolutionary algorithms for solving multiobjective problems: Springer Science & Business Media. https:// link.springer.com/book/10.1007%2F978-0-387-36797-2.

[4] Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Network. 5 (2), 157–166.

[5] Krizhevsky, A., Sutskever, I., E. Hinton, G. (2017). ImageNet classification with deep convolutional neural networks. Commun ACM 60:84–90. https://doi. org/10.1145/3065386.

[6] Kacprzyk. J., (2008). Studies in Computational Intelligence, Volume 121.

[7] He. K, X. Zhang, S. Ren, and J. Sun. (2016). Deep residual learning for image recognition, Paper read at Proceedings of the IEEE conference on computer vision and pattern recognition.

[8] Rezaee, MR., Kadkhodaei, A., Barabadi, A. (2007). Prediction of shear wave velocity from petrophysical data utilizing intelligent systems: An example from a sandstone reservoir of Carnarvon Basin, Australia. J Petrol Sci Eng 55:201–212. https://doi.org/10.1016/j.petrol.2006.08.008.

[9] Rajabi, M., Bohloli, B., Hangar, E. (2010). Intelligent approaches for prediction of compressional, shear and Stoneley wave velocities from conventional well log data: A case study from the Sarvak carbonate reservoir in the Abadan Plain (Southwestern Iran). Comput Geosci 36:647–664. https://doi.org/10.1016/ j.cageo.2009.09.008.

[10] Moatazedian, I., Rahimpour-Bonab, H., Kadkhodaie-Ilkhchi, A., & Rajoli, M. (2011). Prediction of shear and Compressional Wave Velocities from petrophysical data utilizing genetic algorithms technique: A case study in Hendijan and Abuzar fields located in Persian Gulf. Geopersia, 1(1), 1-17.

[11] Asoodeh, M., Bagheripour, P. (2012). Prediction of compressional, shear, and stonely wave velocities from conventional well log data using a committee machine with intelligent systems. Rock mechanics and rock engineering 45, 45–63. https://doi.org/10. 1007/s00603-011-0181-2.

[12] Asoodeh, M., Bagheripour, P. (2013). Neuro-fuzzy reaping of shear wave velocity correlations derived by hybrid genetic algorithm pattern search technique. Open Geosciences 5, 272–284. https:// doi.org/10.2478/s13533-012-0129-4.

[13] Asoodeh, M., Bagheripour, P. (2014). ACE stimulated neural network for shear wave velocity determination from well logs. J Apply Geophysics 107:102–107. https://doi.org/10.1016/ j.jappgeo.2014. 05.014.

[14] Maleki, Sh., Moradzadeh, A., Ghavami Riabi, R., Gholami, R., Sadeghzadeh, F. (2014). Prediction of shear wave velocity using empirical correlations and artificial intelligence methods. NRIAG J Astron Geophys 3:70–81. https://doi.org/10.1016/ j.nrjag.2014.05.001.

[15] Gholami, R., Moradzadeh, A., Rasouli, V., Hanachi, J. (2014). Shear wave velocity prediction using seismic attributes and well log data. Acta Geophysica 62, 818–848. https://doi.org/10.2478/ s11600-013-0200-7.

[16] Akhundi, H., Ghafoori, M., Lashkaripour, G.R. (2014). Prediction of shear wave velocityusing artificial neural network technique, multiple regression and petrophysical data:a case study in asmari reservoir (SW Iran). Open J. Geol. 4, 303–313.

[17] Kadkhodaie-Ilkhchi., A. (2015). A systematic approach for estimation of reservoir rock properties using Ant Colony Optimization. Geopersia, 5(1), 7-17.

[18] Bagheripour., P., Gholami., A., Asoodeh., M., & Vaezzadeh-Asadi., M. (2015). Support vector regression based determination of shear wave velocity. Journal of Petroleum Science and Engineering, 125, 95-99.

[19] Al-Dousari, M., Garrouch, A.A., Al-Omair, O. (2016). Investigating the dependence of shear wave velocity on petrophysical parameters. J. Petrol. Sci. Eng. 146, 286–296.

[20] Singh, S., Kanli, A.I. (2016). Estimating shear wave velocities in oil fields: a neural network approach. Geosciences Journal 20, 221–228. https:// link.springer.com/article/10.1007%2Fs12303-015-0036-z.

[21] Behnia, D., Ahangari, K., Moeinossadat, S. R. (2017). Modeling of shear wave velocity in limestone by soft computing methods. Int J Min Sci Technol 27:423–430. https://doi.org/10.1016/ j.ijmst.2017.03.006.

[22] Shiroodi, SK., Ghafoori, M., Ansari, H. R., Lashkaripour, G., Ghanadian, M. (2017). Shear wave prediction using committee fuzzy model constrained by lithofacies, Zagros basin, SW Iran. J Afr Earth Sc 126:123–135. https://doi.org/10.1016/ j.jafrearsci.2016.11.016.

[23] Mehrgini, B., Izadi, H., Memarian, H. (2019). Shear wave velocity prediction using Elman artificial neural network. Carbonates Evaporites 34 (4), 1281–1291.

[24] Alkinani, H.H., AL hamidi, S., Dunn norman, R. E., Flori, M. A., AL Alwani, A. R. (2019). Intelligent data-driven analytics to predict shear wave velocity in carbonate formations: comparison between recurrent and conventional neural networks. OnePetro. https://onepetro.org/conference-paper/ARMA-2019-0511.

[25] Zhang, Y., Zhong, H. R., Wu, Z., Zhou, H., Ma, Q. (2020). Improvement of petrophysical workflow for shear wave velocity prediction based on machine learning methods for complex

carbonate reservoirs. J Petrol Sci Eng 192:107234. https://doi.org/10.1016/j.petrol.2020.107234.

[26] Wood, DA. (2020). Bakken stratigraphic and type well log learning network exploited to predict and data mine shear wave acoustic velocity. J Appl Geophys 173:103936. https://doi.org/10.1016/j. jappgeo.2019.103936.

[27] Gholami, A., Seyedali, S M., Ansari, H. R. (2020). Estimation of shear wave velocity from post-stack seismic data through committee machine with cuckoo search optimized intelligence models. J Petrol Sci Eng 189:106939. https://doi.org/ 10.1016/j.petrol.2020.106939.

[28] Wang, J., Cao, J., Liu, Z., Lei, X., Zhou, X. (2020). Method of well logging prediction prior to well drilling based onlong short-term memory recurrent neural network[J]. J. Chengdu Univ. Technol. (Sci. Technol. Ed.) 47 (2), 227–236.

[29] Ghorbani, H., Davoodi, S., Davarpanah, A. (2021). Accurate determination of shear wave velocity using LSSVM-GA algorithm based on petrophysical log, 1 ed. European Association of Geoscientists & Engineers, pp. 1–3. https://doi.org/10.3997/2214-4609.202137015.

[30] Olayiwola, T., Sanuade, OA. (2021). A data-driven approach to predict compressional and shear wave velocities in reservoir rocks. Petroleum 7:199–208. https://doi.org/10.1016/ j.petlm.2020.07.008.

[31] Ebrahimi, A., Izadpanahi, A., Ebrahimi, P., Ranjbar, A., (2022). Estimation of shear wave velocity in an Iranian oil reservoir using machine learning methods. J Petrol Sci Eng 209:109841. https://doi.org/10.1016/j.petrol.2021.109841.

[32] Gholami Vijouyeh, A., Kadkhodaie, A., Hassanpour Sedghi, M., (2022). A committee machine with intelligent experts (CMIE) for estimation of fast and slow shear wave velocities utilizing petrophysical logs. Computers & Geosciences. Volume 165, August 2022, 105149. https://doi.org/10.1016/j.cageo.2022.105149.

[33] Nasrnia, B., Falahat, R., Kadkhodaie, A., Gholami Vijouyeh, A., (2023). A committee machine-based estimation of shear velocity log by combining intelligent systems and rock-physics model using metaheuristic algorithms. Volume 126, Part A, November 2023, 106821. https://doi.org/10.1016/ j.engappai.2023.106821.

[34] Alavi, A.H., Gandomi, A.H., Mollahasani, A., Heshmati, A.A.R., Rashed, A., (2010). Modeling of maximum dry density and optimum moisture content of stabilized soil using artificial neural networks.

[35] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural Comput. 9 (8), 1735–1780.

[36] Gers, F.A., Schraudolph, N.N., Schmidhuber, J. (2003). Learning precise timing with lstm recurrent networks. J. Mach. Learn. Res. 3 (1), 115–143.

[37] Biranvand, N., Kikhai, M., Roein. (2019). presenting a method in the field of revealing targets in satellite images using deep learning and with remote sensing and GIS approach, Geographical

[38] Abad, B., Rajabi, A., Mousavi, S. M. V., Mohamadian, N., Wood, D., Ghorbani, H., Davoodi, Sh., Alvar, M., Shahbazi, Kh., 2021b. Hybrid machine learning algorithms to predict condensate viscosity in the near wellbore regions of gas condensate reservoirs. Journal of Natural Gas Science and Engineering 95,104210. https://doi.org/10.1016/j.jngse.2021.104210.

[39] Sezavar, A., Fersi, H., Mohamadzadeh, S. (2015). Image retrieval using deep learning. 4th International Conference on Applied Research in Computer Engineering and Signal Processing. Vol.3, No. 1, 28-42.

[40] Anemangely, M., Ramezanzadeh, A., Tokhmechi, B. (2017). Shear wave travel time estimation from petrophysical logs using ANFIS-PSO algorithm: a case study from Ab-Teymour Oilfield. J. Nat. Gas Sci. Eng. 38, 373–387.

[41] Kingma, D.P., Ba, J. (2014). Adam: A Method for Stochastic Optimization. https://arxiv. org/abs/1412.6980.

[42] Duchi, J., Hazan, E., Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. J.Machine Learning Research. 12 (Jul), 2121–2159.

[43] Tieleman, T., Hinton, G. (2012). Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning 4, 26–30.