

Article

One-Dimensional Convolutional Neural Network with Adaptive Moment Estimation for Modelling of the Sand Retention Test

Nurul Nadhirah Abd Razak ¹, Said Jadid Abdulkadir ^{1,2,*}, Mohd Azuwan Maoinsar ³, Siti Nur Amira Shaffee ⁴ and Mohammed Gamal Ragab ¹

¹ Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; nurul_19000992@utp.edu.my (N.N.A.R.); mohd.gamal_20497@utp.edu.my (M.G.R.)

² Centre for Research in Data Science, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia

³ Department of Petroleum Engineering, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; mazuwan.maoinsar@utp.edu.my

⁴ Advanced Computational Modelling, PETRONAS Research Sdn. Bhd., Bandar Baru Bangi 43000, Malaysia; nuramira_shaffee@petronas.com

* Correspondence: saidjadid.a@utp.edu.my

Abstract: Stand-alone screens (SASs) are active sand control methods where compatible screens and slot sizes are selected through the sand retention test (SRT) to filter an unacceptable amount of sand produced from oil and gas wells. SRTs have been modelled in the laboratory using computer simulation to replicate experimental conditions and ensure that the selected screens are suitable for selected reservoirs. However, the SRT experimental setups and result analyses are not standardized. A few changes made to the experimental setup can cause a huge variation in results, leading to different plugging performance and sand retention analysis. Besides, conducting many laboratory experiments is expensive and time-consuming. Since the application of CNN in the petroleum industry attained promising results for both classification and regression problems, this method is proposed on SRT to reduce the time, cost, and effort to run the laboratory test by predicting the plugging performance and sand production. The application of deep learning has yet to be imposed in SRT. Therefore, in this study, a deep learning model using a one-dimensional convolutional neural network (1D-CNN) with adaptive moment estimation is developed to model the SRT with the aim of classifying plugging sign (screen plug, the screen does not plug) as well as to predict sand production and retained permeability using a varying sand distribution, SAS, screen slot size, and sand concentration as inputs. The performance of the proposed 1D-CNN model for the slurry test shows that the prediction of retained permeability and the classification of plugging sign achieved robust accuracy with more than a 90% value of R^2 , while the prediction of sand production achieved 77% accuracy. In addition, the model for the sand pack test achieved 84% accuracy in predicting sand production. For comparative model performance, gradient boosting (GB), K-nearest neighbor (KNN), random forest (RF), and support vector machine (SVM) were also modelled on the same datasets. The results showed that the proposed 1D-CNN model outperforms the other four machine learning models for both SRT tests in terms of prediction accuracy.



Citation: Razak, N.N.A.; Abdulkadir, S.J.; Maoinsar, M.A.; Shaffee, S.N.A.; Ragab, M.G. One-Dimensional Convolutional Neural Network with Adaptive Moment Estimation for Modelling of the Sand Retention Test. *Appl. Sci.* **2021**, *11*, 3802. <https://doi.org/10.3390/app11093802>

Received: 14 February 2021

Accepted: 3 April 2021

Published: 22 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: adaptive moment estimation; deep learning; one-dimensional convolutional neural network; sand retention test; stand-alone screens

1. Introduction

The sand retention test (SRT) is a procedure often used to select the most optimal sand screen to be used in reservoir sand control [1]. Optimal sand screen selection refers to the selection of the most appropriate screen for sand control completion, minimizing sand production and maximizing hydrocarbon production. SRT simulates the sand production

into a wellbore after passing through a filter, a standard test in the upstream sector in the oil and gas industry. It helps engineers choose the right screen and aperture size for the right field and environment.

Implementing SRT helps in providing a better understanding of sand retention efficiency and plugging performance. It can be done in the laboratory or by simulation of mathematical models. Both results from laboratory and simulation are compared to examine the method's accuracy [2]. The SRT laboratory method that focuses on stand-alone screen (SAS) application is classified into two groups: the slurry test and the sand pack test [3]. Generally, the slurry test refers to an experiment where sand is suspended in a fluid to form a slurry, which is then pumped through the screen, while for the sand pack test, sand is placed directly onto the screen along with the confining stress to ensure that the sand is compressed with the screen before the fluid is pumped through the sand pack and the screen [4]. One of the experimental setups of slurry and sand pack tests are shown in Figure 1.

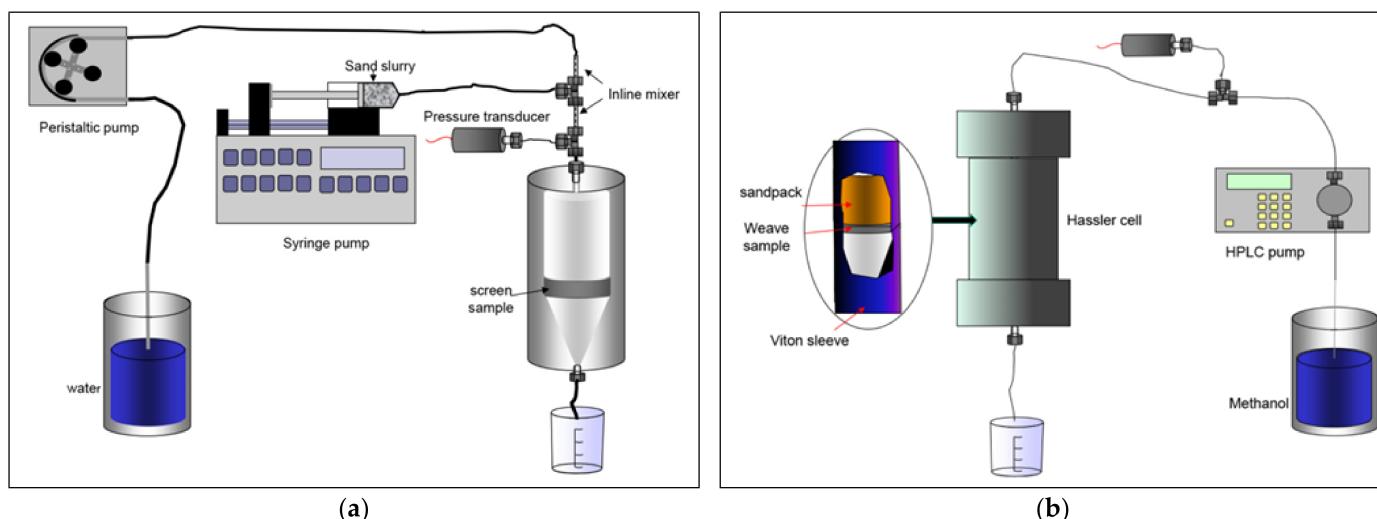


Figure 1. Sand retention test (SRT) experimental setup [4]: (a) Slurry test; (b) Sand pack test.

However, the SRT problems observed from previous studies [1–10] showed no standard guideline to carry out the SRT experiment. There are various experimental setups and different ways of interpreting the results. The plugging performance and sand retention efficiency are analyzed differently in selecting the most compatible SAS and screen slot size. A few changes made to the experimental setup can cause a huge variation in results. Besides, conducting many laboratory experiments is expensive and time-consuming [11]. Testing one particular screen and slot size to analyze the sand production can take hours to reach the final result despite using the laboratory or simulation tests.

Deep learning is a sub-branch of machine learning inspired by biological neurons of the brain called artificial neural networks (ANNs) with representation learning. It highlights the use of multiple connected layers with a dominant learning ability to take inputs as the features, which are modelled to predict outputs [12]. Deep learning has become the most vital discovery and research hotspot, which can tackle intricate patterns in a massive dataset of any data type [13]. It is not only implemented to model the prediction or classification problem, but it can also be applied for modelling forecasting time-series data [14–20]. It can automatically apply feature extraction where a separate implementation is needed for shallow machine learning approaches.

Convolutional neural networks (CNNs) is a specific type of ANN that focus on many deep layers of neural networks, which uses convolutional and pooling strategies [21]. It is a deep learning model used primarily in computer vision problems, such as image classification [22], image segmentation [23], and video object segmentation [24], where it

shows promising results. It can also be used for regression problems to generate complex models with higher accuracy using complex datasets, which cannot be done with just a simple regression function. CNN typically uses two- or three-dimensional neural layers for classification problems that take the input in the same dimension to undergo a feature learning process. It works the same way despite having one, two, or three dimensions. The differences between dimensions are the input structure, pool size, filter size, and how it shifts from features to features. Developing CNN starts with a configuration of the external parameters known as hyperparameters [25]. Adjusting the hyperparameters is significant to get a better model with high predictive power.

The application of CNN in the petroleum industry has been undertaken to detect hydrocarbon zones in seismic images where 2D-CNN is used in the image segmentation process, and the model achieved more than 80% accuracy [26]. The developed model used 2D-CNN because the input of seismic images is in two dimensions (2D). Other than image segmentation, the implementation of CNN has also been applied for image recognition in the petroleum industry. Zhou et al. [27] developed a well pump troubleshooting model in which the 2D power card images are taken as input to classify different types and severity of pump troubles. The analysis of power card images helps to identify issues that could impact the oil well production and help in the configuration of pumping parameters. The accuracy of power cards equivalent to seven pump troubles achieved more than 96%, except for the undersupplying trouble type, which obtained 87% accuracy.

Besides, the CNN model's development on the regression problem was carried out by Daolun et al. [28], where a radial composite reservoir model was developed using 2D-CNN and verified by the oilfield measurement data. The mean absolute error (MAE) was used as the performance metric to verify the developed model. All outputs showed a small error with less than 0.7 for both validation and testing sets. Kwon et al. [29] applied 2D-CNN to determine the location of an oil well under geological uncertainty by predicting the cumulative oil production of reservoir simulation. The performance of the 2D-CNN model was compared with a shallow machine learning model, ANN, and the results showed that the 2D-CNN model achieved 88% accuracy with a relative error of 0.035. In comparison, the ANN model achieved 84% accuracy with 0.24 relative error. The CNN model outperforms the ANN model in predicting cumulative oil production. Li et al. [30] applied the CNN method to simultaneously predict the volume flow rate of multiphase flow of oil, gas, and water phases. The results are very convincing, with more than 80% accuracy for all phases and were verified by the measurement of an individual well in the petroleum industry.

Since the application of CNN in the petroleum industry attained promising results for both classification and regression problems, this method was applied on SRT to reduce the time and effort to run the laboratory test by predicting the plugging performance and sand production. The application of deep learning has yet to be imposed in SRT. Therefore, a one-dimensional convolutional neural network (1D-CNN) was developed for SRT modelling to classify plugging sign, predict sand production, and retain permeability using various sand distribution, stand-alone screen, screen slot size, and sand concentration. Classification and prediction of the SRT results were used to filter the most compatible SAS and screen slot size accordingly. Then, 1D-CNN hyperparameters were tuned manually to determine optimal hyperparameters combination that will provide higher predictive power for the developed model.

1.1. Variable Identification in the Sand Retention Test

Since there is no implementation of any deep learning algorithm in SRT modelling, the SRT dataset requires feature selection using statistical analyses to determine appropriate features to fit into the model. Therefore, all the experimental setup and correlation results from various studies were compared to identify the factors that affect the plugging performance and sand retention efficiency to create the SRT dataset.

Some of the factors considered to interpret screen plugging are particle size distribution (PSD), flow rate, weave open volume, and pressure gradient. In contrast, the variables

that affect screen retention and sand production are PSD, fines content, sorting and uniformity coefficients, SAS, screen slot width, fluid viscosity, fluid density, flow rate, pressure gradient, and sand concentration in the test cell. These factors were classified into four groups: sand characteristics, screen characteristics, fluid characteristics, and the condition in test cell.

1.1.1. Sand Characteristics

The PSD of d_1 , d_5 , d_{10} , d_{30} , and d_{50} are used to analyze plugging performance and sand production [3–8]. The PSD of d_x refers to the percentage of sand in the sample with a specific size. For example, if d_1 is equal to 300 microns, then 1% of the sample has a sand size of 300 microns. The PSD of d_5 , d_{10} , d_{30} , and d_{50} of the same sample will have a smaller sand size than d_1 because the higher the value of x , the smaller the sand size in the sample. PSD of d_1 , d_5 , and d_{10} , which represent the large sand grains, shows a good correlation with sand retention, while PSD of d_{50} , representing the median sand size, gives a weaker relationship [3]. Markestad et al. [5] used only one point on the particle distribution curve, which is d_{10} , but it cannot accurately predict plugging nor sand production. In other words, parameters other than d_{10} must be considered when choosing the slot width of the sand control screen. In contrast, Ballard and Beare [4,6] used PSD of d_{10} and d_{30} as an indicator to select the screen slot size, which can control the amount of sand produced for a particular screen. For good sand retention, the screen slot size should be smaller than d_{10} because the bigger the screen slot size, the higher the amount of sand produced [7,8].

Next, the other sand characteristics used in SRT are fines content, sorting, and uniformity coefficients. Ballard and Beare [4] found that a high fines content has a shallow impact on the sand production compared to the largest grains in the sand distribution. If the size of the sand in the sample is mostly less than 45 microns, then the amount of sand that passes through the screen will be high. The sorting and uniformity coefficients show a bad interaction with the amount of sand passing through the weave [4,6]. However, the combination of the fines content with the sorting coefficient gives a good interaction, where a poorly sorted sand with many fine particles results in a high risk of sand production. In contrast, the well-sorted sand with high fines content results in a smaller amount of sand passing through the weave [3,5].

1.1.2. Screen Characteristics

The factors involved in screen characteristics are weave open volume, type of screen, and screen slot size. Ballard and Beare [6] found that plugging also tends to occur for a small weave open volume. The lower the weave open volume, the higher the tendency of sand to lodge in the weave, reducing the overall area open to flow. Besides, the type of screen used also has an impact on retention. Ballard and Beare [4] used two kinds of screen: premium and wire wrapped screen. The performance of each screen can be identified by investigating the effect of d_{10} on the amount of sand produced. The premium screen shows excellent performance, but the wire-wrapped screen indicates otherwise. Mathisen et al. [9] recommended a single wire-wrapped screen if the sand distribution is good and has a low tendency for more sand production. Otherwise, a premium screen should be used. Ballard and Beare [6] compared the screen performance between wire wrap with a metal mesh screen on the screen slot size by observing the pressure gradient. It turns out that the pressure gradient on the wire wrap screen is more sensitive to the screen slot size than the metal mesh screen due to the lower flow area and different flow regime.

1.1.3. Fluid Characteristics

The density and viscosity of fluid used in the sand slurry also affect sand production in SRT. The high density, which is a more viscous carrier fluid, is used along with low flow rates, resulting in a longer delay between coarse particles and fines particles reaching the screen [6]. As a result, the sand is carried with the flow into the slots and leads to high sand production.

1.1.4. The Condition in Test Cell

The test cells' conditions that consider plugging performance and sand retention efficiency are the flow rate, sand concentration, and pressure. The different flow rates used in the experiment lead to different results. The high flow rate initiated at the beginning of the experiment leads to plugging, but gradually increasing the flow rate throughout the experiment reduces the risk of plugging [5]. However, initiating the experiment with a low flow rate without increasing it leads to a higher amount of sand passing through the screen [6,10].

The sand concentration in the test cell also affects sand production. Ballard and Beare [6] found that the sand slurry's low sand concentration leads to an increasing amount of sand produced before bridging occurs. Large grains are essential for the bridging process to start on the screen and must be large enough to fit through the slot. This finding is supported by Fisher and Hamby [10], where the lower volume fraction of formation sand in the flow stream causes a higher sand production.

Other than that, pressure drop and pressure gradient are used to interpret the sand production and plugging performance. According to the practice of SAS selection recommended by Mathisen et al. [9], the screen with the lowest pressure drop and the highest permeability are associated with high sand retention. Ballard et al. [7] investigated the correlation of a sand reaching screen with the pressure drop. It showed a steep gradient for simulated laser particle size analysis (LPSA) sand compared to simulated sieve analysis and reservoir sand. An increase in the pressure drop represents an increase in the flow resistance through the screen, due to the build-up of particles on top and inside the screen [9]. Therefore, a higher pressure drop produces a more elevated amount of sand production [10].

Since there is no standard way to interpret the trend of the pressure gradient towards plugging, Mathisen et al. [9] mentioned that the pattern that shows a linear pressure build-up represents the formation of the permeable sand pack on top of the screen while the exponential behavior is a sign of plugging occurring on the screen. However, the observation of pressure during the laboratory test shows that plugging does not contribute to the decrease or increase of pressure [6]. The pressure slowly falls when there is some plugging of the screen and decreases initially when some new sand is washed through the screen. In other words, the pressure gradient results come from variations in sand characteristics rather than plugging.

1.1.5. Variable Summary

According to various studies [3–10], the variables that were considered for data collection are PSD of d_1 , d_5 , d_{10} , d_{30} , and d_{50} ; fines content; sorting and uniformity coefficients; weave open volume; type of screen; screen slot size; fluid viscosity; fluid density; flow rate; sand concentration; pressure drop; pressure gradient; screen plugging; amount of sand produced; and retained permeability. The detailed procedures of developing a 1D-CNN model for SRT are described in Section 2, and the result of the model and the comparative model performances are discussed in Section 3.

2. Materials and Methods

The 1D-CNN model development workflow is presented in Figure 2 and the preparation of the methods are explained in Section 2.1 until Section 2.5. The workflow of 1D-CNN model development starts with data collection where all variables related to slurry and sand pack tests are collected and identification of inputs and outputs are made thoroughly. Next, the collected data is analyzed to explore the data and gain useful information. After that, the data undergoes pre-processing and normalization to be fitted for modelling. The modelling phase begins with the initialization of hyperparameters for 1D-CNN. Once the hyperparameters are initialized, the 1D-CNN model is trained with an adaptive moment estimation (Adam) optimizer. The hyperparameters are tuned and iterated using the trial-and-error method until the model shows good performance metrics with a minimal

loss function. In other words, the iteration will stop when the loss function has reached convergence for both training and testing data, but the hyperparameters are tuned if the loss function does not converge. Besides, the stopping criteria for each iteration depends on the number of epochs. All models with a different set of hyperparameters are evaluated and validated. Lastly, the final 1D-CNN model with the Adam optimizer is developed for SRT.

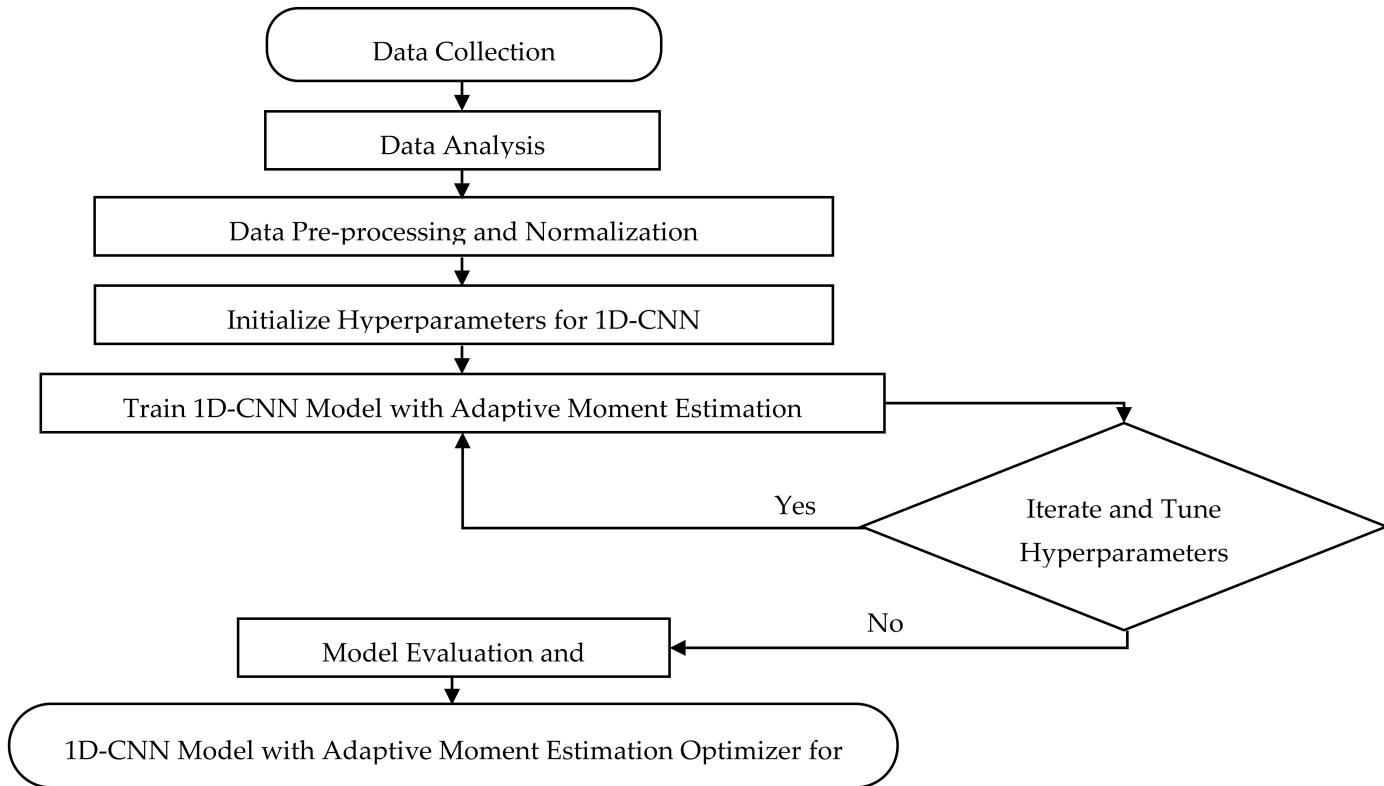


Figure 2. One-dimensional convolutional neural network (1D-CNN) model development workflow.

2.1. Data Collection

The SRT dataset was extracted from various works of literature related to slurry [1–34] and sand pack tests [3–36]. The sand retention experimental report from PETRONAS Research Sdn. Bhd. (PRSB) was also added into the dataset to identify the standard variable used for the whole SRT process. The standard set of variables is needed to ensure that no essential variable is left behind. As the SRT modelling problem falls under both classification and regression problems, variable identification works were done thoroughly. The variable was classified into input and output, specifying whether it is qualitative or quantitative and labelled as a discrete or continuous variable.

The Availability of Sand Retention Test Variables

Once the factors that affect screen plugging and sand retention efficiency were identified, data quality assessment was done to identify and remove missing values in the dataset. In total, 38 inputs and 21 outputs with 516 observations were identified for the slurry test, while there were 42 inputs and 19 outputs with 683 observations for the sand pack test. However, due to the unmatched variable used between the literature and PETRONAS's report, a lot of missing values were detected in the dataset, especially for the sand pack test. The previous works from the literature did not reveal all the exact parameters used for SRT, leading to missing values. In this study, the variables were reduced to 8 inputs and 4 outputs for the slurry test with a different number of observations for each output. As for the sand pack test, the variables were reduced to 5 inputs and only 1 output with

263 observations. The reduced set of variables is shown in Table 1, and the set of data used to train 1D-CNN is shown in Table 2.

Table 1. The reduced set of variables.

Variable Description	Variable Abbreviation	Unit	Type of Variable	Test
10% PSD of formation sand	D10_B	micron (μm)	Input	Slurry
50% PSD of formation sand	D50_B	micron (μm)	Input	Slurry
90% PSD of formation sand	D90_B	micron (μm)	Input	Slurry
Uniformity coefficient	UC	-	Input	Slurry
Fines content	FINE_CO_B	%	Input	Slurry
Aperture size of the screen	SLOT_SIZE	micron (μm)	Input	Slurry
Type of screen: Single Wire Wrap (SWW), Premium (PREMIUM), Wire Wrap Screen (WWS), Expandable Sand Screen (ESS), Ceramic (CERAMIC), Metal Mesh Screen (MMS)	SCREEN	-	Input	Slurry and sand pack
Sand concentration in test cell	SAND_CONC_IN_TC	gram/liter (g/L)	Input	Slurry
Sizing criteria 1: $\frac{\text{SLOT_SIZE}}{\text{D10_B}}$	SIZE_CR1	-	Input	Sand pack
Sizing criteria 2: $\frac{\text{SLOT_SIZE}}{\text{D50_B}}$	SIZE_CR2	-	Input	Sand pack
Sizing criteria 3: $\frac{\text{SLOT_SIZE}}{(\frac{\text{D10_B}}{\text{UC}})}$	SIZE_CR3	-	Input	Sand pack
Sizing criteria 4: $\frac{\text{SLOT_SIZE}}{(\frac{\text{D50_B}}{\text{UC}})}$	SIZE_CR4	-	Input	Sand pack
Sign of screen plugging	PLUG_SIGN	-	Output	Slurry
Amount of sand produced in gram	SAND_PRODUCED	gram (g)	Output	Slurry
Retained permeability	RETAINED_PRM	%	Output	Slurry
Amount of sand produced per unit area	SAND_PROD_PER_AREA	pound per square foot (lb/ft ²)	Output	Slurry and sand pack

Table 2. Set of data used to train 1D-CNN.

Set	Test	Number of Observations	Number of Variable (Output)
1	Slurry	162	8 inputs and 1 output (PLUG_SIGN)
2	Slurry	367	8 inputs and 1 output (SAND_PRODUCED)
3	Slurry	65	8 inputs and 1 output (RETAINED_PRM)
4	Slurry	343	8 inputs and 1 output (SAND_PROD_PER_AREA)
5	Sand pack	263	5 inputs and 1 output (SAND_PROD_PER_AREA)

2.2. Data Analysis

Data analysis for the SRT dataset was done by focusing on descriptive and inferential statistics. Descriptive statistics concentrate on the univariate analysis, where the distribution of each variable in Table 1 is visualized to summarize the data. In contrast, inferential statistics highlight the bivariate analysis where a statistical test is used to measure the correlation between input and output variables.

2.2.1. Univariate Analysis

The distribution of continuous variables was visualized using the kernel density plot to display the observed values' dispersion. In contrast, the bar chart was used for categorical variables to represent the group's frequency in each variable, as shown in Figure 3. The kernel density plot was used instead of a typical histogram because it produces a smooth curve without having a normality assumption by considering each observation of the variable to contribute to different classes representing the distribution [37]. It also gives clear information on whether the distribution has a normal, bimodal, or skewed distribution. A bar chart was used to visualize PLUG_SIGN and SCREEN variables because the number of groups in both variables is not huge. The groups in both variables are easily identified and interpreted when looking at the bar chart.

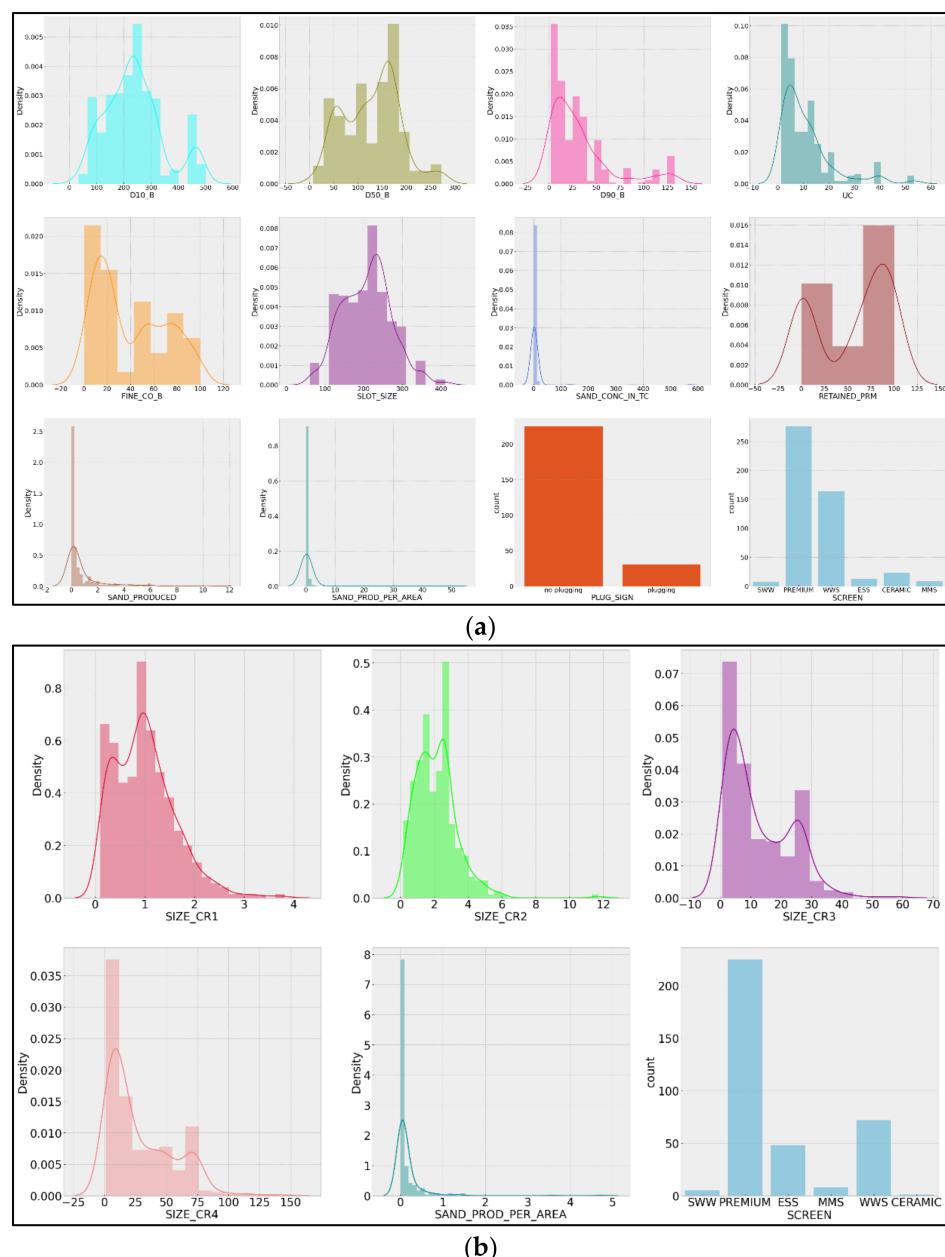


Figure 3. Distribution plot of sand retention test (SRT) variables. (a) Slurry test variables; (b) sand pack test variables.

As for the slurry data, the PLUG_SIGN and the SCREEN variables fall under the categorical type, while the SCREEN variable is the only categorical type in the sand pack data. It shows that the occurrence of the screen to plug is lower than the screen not to plug. The premium screen frequency has the highest number for both the slurry and sand pack data, followed by the WWS screen.

The continuous slurry variables with bimodal distributions were D10_B, D50_B, D90_B, UC, FINE_CO_B, and RETAINED_PRM. Thus, the mode was used as central values for these variables. On the other hand, the SLOT_SIZE variable distribution looked symmetrical, while the SAND_CONC_IN_TC, SAND_PRODUCED, and SAND_PROD_PER_AREA gave positive skewed distributions. Hence, the mean value was used to represent the central value of the SLOT_SIZE, whereas the median was used for the positive skewed distributions.

Furthermore, all continuous variables of sand pack data showed positive skewed distributions where the average value of each variable is greater than the median, and all variables have outliers. However, SIZE_CR1, SIZE_CR2, SIZE_CR3, and SIZE_CR4 variables also had two peaks in the distribution in which the data had more than one center, which means only SAND_PROD_PER_AREA showed a positive skewed distribution with a single peak. Therefore, the useful measure that can accurately capture the central tendency for SAND_PROD_PER_AREA is the median. Simultaneously, the mode is used for SIZE_CR1, SIZE_CR2, SIZE_CR3 and SIZE_CR4 because both the mean and median have no meaningful interpretations for a bimodal distribution.

2.2.2. Bivariate Analysis

Bivariate analysis focused on correlation analysis where Pearson's product-moment and Spearman's rank correlations were used to evaluate the degree of association between two continuous variables and determine the direction and strength of the relationship [38]. The significance of both correlations was tested using the p -value, where it represents the probability of rejecting the null hypothesis when the p -value is less than the significance level of 5%, $\alpha = 0.05$ [39]. The null hypothesis refers to the hypothesis of the correlation coefficient of two continuous variables that are not significantly different from zero and no statistical significance exists in the population. By rejecting the null hypothesis, two continuous variables are statistically significant. The significance of the p -value and the highest correlation coefficient were used as the final output. The only variables that will undergo correlation analysis are between continuous inputs and outputs. There was no computation of association within the inputs or outputs because the most important thing to explore is the dependency of output variables towards the input variables.

The Pearson's product-moment is denoted as r ; and Spearman's rank correlations, R were computed according to Equations (1) and (2) [40]:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left[\sum_{i=1}^n (x_i - \bar{x})^2 \right] \left[\sum_{i=1}^n (y_i - \bar{y})^2 \right]}} \quad (1)$$

$$R = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)} \quad (2)$$

In Equations (1) and (2), x_i is the value of input for the i^{th} observation; \bar{x} is the average value of input; y_i is the value of output for the i^{th} observation; \bar{y} is the average value of output; n is the number of observations in the dataset; and d_i is the difference between the ranks of corresponding variables.

A Pearson correlation coefficient close to zero indicates no linear relationship between variables or the variables are independent while a -1 or $+1$ value indicates a perfect negative or positive linear relationship, respectively [41]. If the relationship between input and output variables is nonlinear, then the linear relationship's degree might be low. Thus, r , is close to zero [37]. When the p -value is less than α , in this case, 0.05, there is a statistically

significant correlation between inputs and outputs. If the r value is close to zero despite having a p -value < 0.05 , the significant correlation might be nonlinear. Therefore, the Spearman's rank coefficient was used instead. The correlation coefficient of both the slurry and sand pack tests were visualized using a heatmap, as shown in Figure 4.

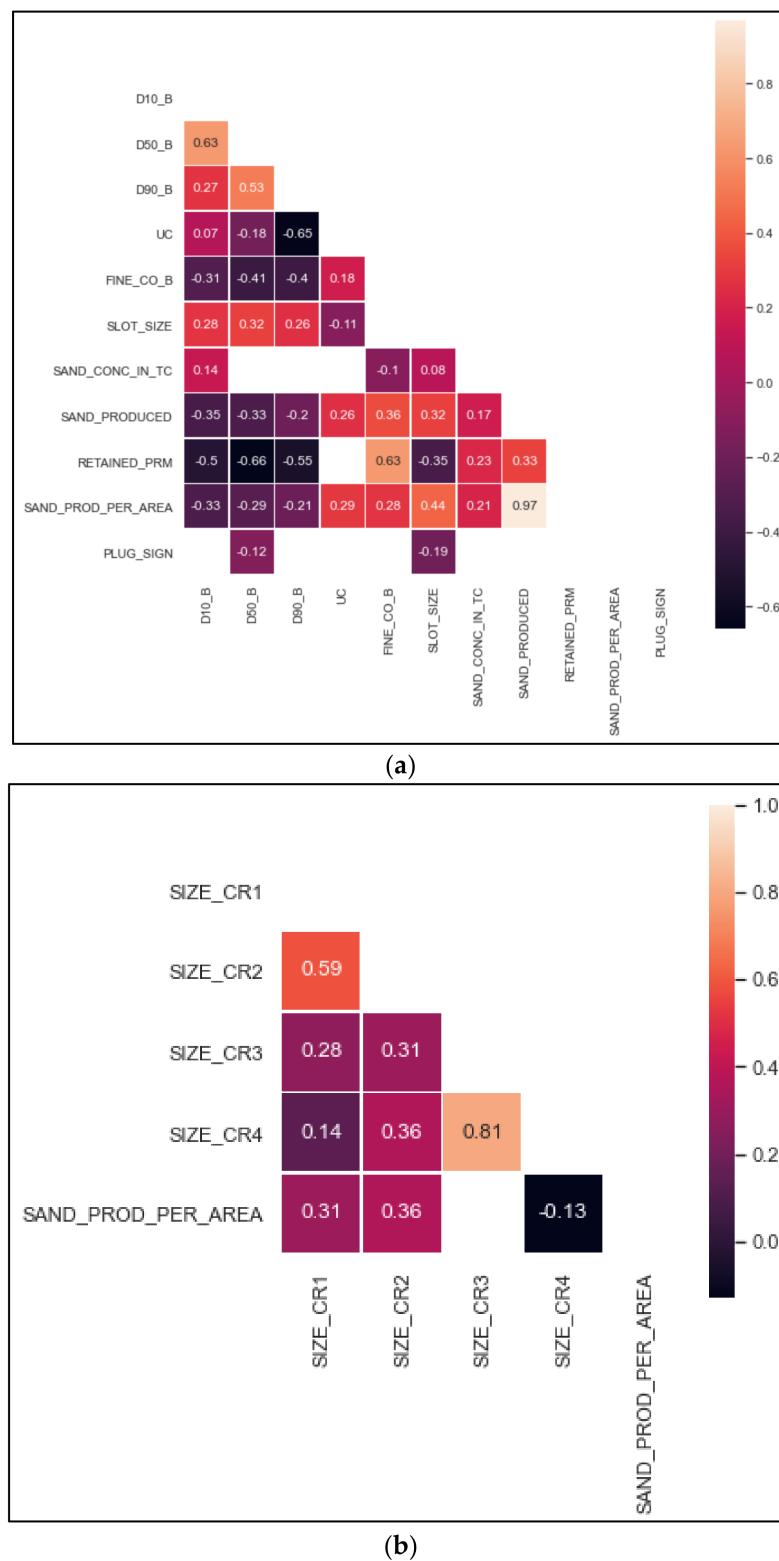


Figure 4. Heatmap plot of SRT variables. (a) Correlation analysis of continuous slurry test variables; (b) correlation analysis of continuous sand pack test variables.

The white box (no value) in Figure 4 represents the hypothesis tests that failed to reject the null hypothesis for both the Pearson and Spearman correlation tests due to the *p*-value, which is greater than the significance level. Thus, the correlation coefficient is not significant for use. The slurry test's highest correlation coefficient is between D50_B (input) and RETAINED_PRM (output) with a -0.66 value. It means that there is a moderate negative correlation between PSD of d_{50} with the retained permeability. Meanwhile, the highest correlation coefficient of the sand pack test was observed between SIZE_CR2 (input) with SAND_PROD_PER_AREA (output) with a 0.36 value, which indicates a low positive correlation between the sizing criteria 2 with the amount of sand produced per unit area.

2.3. Data Preparation

SRT data underwent four processes before it could fit into the 1D-CNN model. The process started with handling categorical data, pre-processing the data using min-max scaler—famously called normalization—, reshaping the scaled data, and splitting the data into training and testing sets.

The categorical variables were identified in the data collection phase, which are the screen and plugging sign. As the group in screen variable is nominal, the one-hot encoding method was used to change the data from the character form to numeric form. For example, if the screen column has three categorical groups: premium, wire wrap, and metal mesh, three new columns were created with the value of 0 and 1. A premium column with a value of 1 means the observation used a premium screen while 0 refers to the other screens, as in wire wrap or metal mesh. As for the plugging sign, the group can be converted into a number using a label encoder without creating a new column. A value of 0 represents screens that do not plug, while 1 is assigned when the screen plugging occurs for respective observations.

Next, normalization was performed to transform variables within the range of 0 and 1. The normalized data have a mean of zero and a standard deviation of one. Using unnormalized data to train the neural network can lead to an unstable or slow learning process [42]. The min-max scaler for normalization was computed as follows:

$$\hat{x} = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (3)$$

In Equation (3), x_i is the value of the variable for the i^{th} observation, $\min(x)$ is the minimum value of variable, and $\max(x)$ is the maximum value of the variable.

Subsequently, the scaling data needed to be reshaped into three dimensions, representing the row, column, and channel. Lastly, the reshaped data were divided into two sets using a 70:30 ratio for the training and testing set.

2.4. Convolutional Neural Network Experimental Setup

A 1D-CNN architecture for the SRT dataset is composed of an input layer, convolutional (CNN) layer, rectified linear unit (ReLU) activation function, pooling layer, fully connected (FC) layer or often called the dense layer, dropout layer, flatten layer, and a fully connected output layer. The process from the input layer to the output layer is called forward propagation, where the input is transferred and processed for feature extraction and prediction is generated. Meanwhile, updating model parameters, such as kernels, weights, and biases, from the output layer back to the input layer is called backward propagation or backpropagation. The forward and backward propagations were applied in the training phase to obtain the model parameters' values to reduce the loss function [43]. The pseudocode of the 1D-CNN modelling process is shown in Algorithm 1 and the detailed sequence of the 1D-CNN architecture after the configuration of the hyperparameters is presented in Figure 5.

Algorithm 1 1D-CNN modelling process

*** Input:**

- Prepare the training and testing datasets
- Split inputs and output of both training and testing datasets
- Initialize model parameters and hyperparameters
- Set the complementary of the regression and classification accuracy
- Set adaptive moment estimation as the optimizer

*** Output:**

- 1D-CNN model with adaptive moment estimation optimizer

1 For each iteration **Do:**

2 Fit the model using training data

3 Calculate accuracy and loss function

4 Evaluate the trained model using performance metrics

5 Compare actual and predicted values of the trained model

6 Backpropagate error and adjust the model parameters

7 If better loss **Do:**

8 Predict the output of testing data using the trained network

9 Evaluate the model using performance metrics

10 Compare actual and predicted values of the model

11 Save network (model, weights)

12 End If

13 Tune hyperparameters and return to step 2

14 End For

The input layer holds the SRT dataset before it is fed into the hidden layers. The hidden layers are the intermediate layers between the input and output layers, where the processing level occurs [44]. The number of layers is empirically optimized during the training and validation process. The hidden layers shown in Figure 5 consist of convolution, ReLU, pooling, dense, dropout, and flatten layers. The CNN layer generates a feature map when each neuron's output is computed by performing the dot product of input values with the weight of filters called kernels [44]. The filter is convolved with the features in the input layer, and it adopts a one-dimensional structure where it will extract the features in sequence according to its size and stride. The stride refers to the number of columns that should be shifted at each step, while the kernel size specifies the number of columns that the filter should extract at a single time [45]. The output feature map will then undergo nonlinear transformation using the rectified linear unit (ReLU) activation function. The ReLU activation function allows the network to learn the complex relationship of the SRT dataset and normalizes each neuron's output into a small range between 0 and 1. It is commonly used for deep learning models to achieve better performance results [46] without changing the function's output shape.

The outputs from the activation function are scaled down in the pooling layer where the downsampling operation, such as the average or maximum function, takes place [44]. The dimensionality of the features is reduced depending on the value of the pool size and stride to give the optimal network structure and enhance feature robustness [47]. In this architecture, the downsampling method used is max pooling, where the pool window slides across the input feature maps from the CNN layer by the stride value and takes the maximum amount as the output. Next, the pooling layer's output is fed into three dense layers where each layer passes through the ReLU activation function to provide stable convergence for the model. The number of neurons in the dense layer refers to the number of units that perform a linear transformation of the inputs with weights and biases [48]. The dense layer helps to interpret the learned features that have gone through feature extraction in the CNN and pooling layer before predicting the output.

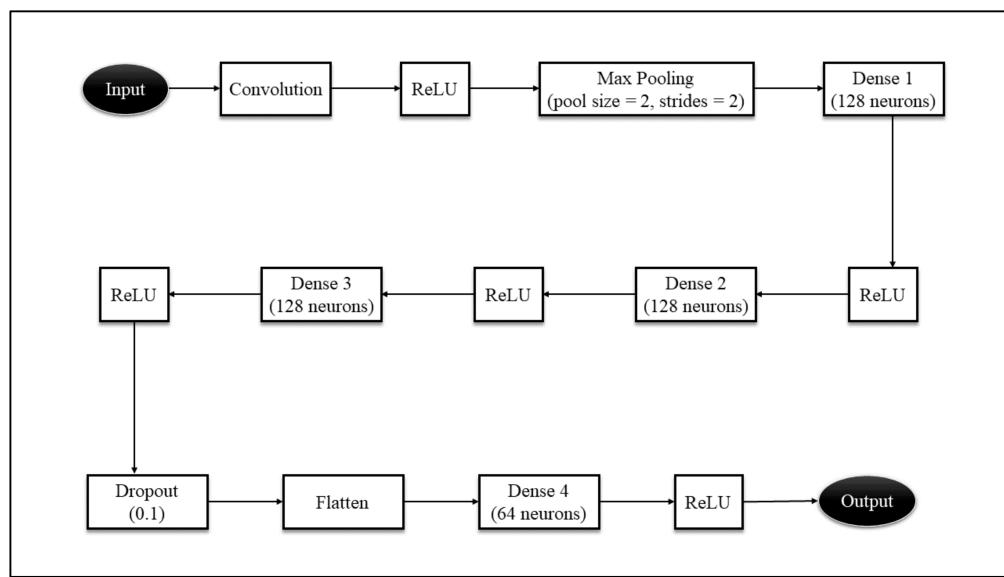


Figure 5. The detailed block of the 1D-CNN architecture for the SRT dataset after the configuration of the hyperparameters.

The forward propagation from the input layer to the input of the neuron where the convolution, activation function, and downsampling operation take place are demonstrated in (4)–(6).

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} (s_i^{l-1}, w_{ik}^{l-1}) \quad (4)$$

$$y_k^l = f(x_k^l) \quad (5)$$

$$s_k^l = y_k^l \downarrow SS \quad (6)$$

In (4)–(6), x_k^l is the input in the CNN layer; b_k^l is the bias of the k^{th} neuron in the CNN layer; s_i^{l-1} is the output of the i^{th} neuron in the input layer; w_{ik}^{l-1} is the kernel from the i^{th} neuron in the input layer to the k^{th} neuron in the CNN layer; $f(x_k^l)$ represents the ReLU activation function applied to the input in the CNN layer; y_k^l is the output of the convolution operation; s_k^l is the output of the k^{th} neuron at the pooling layer; and $\downarrow SS$ refers to the downsampling operation with the factor, SS . Next, the forward propagation from the pooling layer to the input of the neuron in the dense layer is formulated in (7) and (8).

$$x_i^{l+1} = b_i^{l+1} + \sum_{k=1}^{N_l} (s_k^l, w_{ki}^l) \quad (7)$$

$$y_i^{l+1} = f(x_i^{l+1}) \quad (8)$$

In (7) and (8); x_i^{l+1} is the input in dense layer; b_i^{l+1} is the bias of the i^{th} neuron in the dense layer; w_{ki}^l is the weight from the k^{th} neuron in the pooling layer to the i^{th} neuron in the dense layer; and y_i^{l+1} is the output of the dense layer after applying the ReLU activation function.

The dropout layer is added after the third dense layer to reduce overfitting due to the large number of model parameters that the network has. Dropout is a technique that drops specific nodes randomly according to the dropout rate and creates a network ensemble [48]. All the connections from and to the dropped nodes need to be removed as well. The use of dropout works nicely to improve regularization error and boost the testing set performance [44]. Lastly, the output from the dropout layer turns into a single vector in

the flatten layer and passes through a dense layer before the output layer, which follows the format that can be used to generate the final prediction. The flatten layer is needed to convert the output from the dropout layer, which has a 3D output shape into a 1D output shape with a single long vector without changing the output values. The output layer can only take the input with a single vector from the previous layer.

Hyperparameters in the Convolutional Neural Network

Before implementing the 1D-CNN model on the SRT dataset, specific parameter values needed to be configured. There are two types of parameters that are used to train 1D-CNN model to make predictions. The internal parameters, which are learned automatically during the backpropagation process, are called model parameters. The model parameters are present only in the CNN, dense and output layers where the weights of filters or kernels, weights of neurons, and biases are learned during the training of 1D-CNN.

The external parameters that determine the structure of 1D-CNN and how it is trained is called hyperparameters. Hyperparameter tuning is based on a manual trial-and-error process. The list of hyperparameters and the range of values are shown in Table 3. Tuning of the hyperparameters leads to the development of six different models for each dataset, as shown in Table 4.

Table 3. Hyperparameter ranges of 1D-CNN models.

Hyperparameter	Range
Number of CNN layer	(1)
Number of filters	(32,64,128,256)
Kernel size	(2,3,5,9,11)
Stride in CNN layer	(1–3)
Activation function	(ReLU)
Pool size	(2)
Stride in pooling layer	(2)
Number of FC layer	(4)
Dropout rate	(0.1)
Epoch	(30,50,60)
Batch size	(32)
Optimizer	(Adaptive moment estimation)

All the list of hyperparameters is briefly explained previously except for the epoch, batch size, and optimizer. An epoch refers to the entire training dataset that passes through the forward and backward propagation through the network at one time, while the batch size refers to the number of samples (rows of data) in a single iteration before the updating of model parameters [25].

Table 4. Hyperparameter configuration for six models.

Hyperparameter	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Number of CNN layer	1	1	1	1	1	1
Number of filters	64	32	128	256	256	128
Kernel size	2	3	5	11 (for set 1,2,4,5) and 9 (only for set 3)	5	3
Stride in CNN layer	3 (for set 1,2,3,4) and 1 (only for set 5)	1	2	2	2	1

Table 4. Cont.

Hyperparameter	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Activation function	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU
Pool size	2	2	2	2	2	2
Stride in pooling layer	2	2	2	2	2	2
Number of FC layer	4	4	4	4	4	4
Dropout rate	0.1	0.1	0.1	0.1	0.1	0.1
Epoch	60	60	30	50	50	50
Batch size	32	32	32	32	32	32

The total iteration needed to complete one epoch was calculated by dividing the number of samples in the training dataset by the batch size. The number of times that the model parameters will be updated is equal to the total number of iterations. For example, if the training dataset has 160 rows and the batch size is set to 32, then the model parameters are updated five times. Five iterations are needed to complete one epoch. Likewise, if the epoch is set to 60, then 1D-CNN is trained until 60 epochs, where 300 iterations are required for the entire training process.

The optimizer is an optimization algorithm used to update the model parameters iteratively based on the training dataset by calculating the error to minimize the loss function [45]. Adaptive moment estimation is an optimizer that is fast to converge, efficient in learning model parameters, and adequately solves practical deep learning problems [49,50]. Equations (9)–(13) demonstrate the model parameters update using adaptive moment estimation optimizer, and the value of hyperparameters used in the Adam optimizer is presented in Table 5:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (9)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (10)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (11)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (12)$$

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{\hat{v}_t} - \epsilon} \quad (13)$$

In Equations (9)–(13); m_t is the first moment estimate of the gradient at timestep t ; m_{t-1} is the first moment estimate of the gradient at timestep $t - 1$; v_t is the second moment estimate of the squared gradient at timestep t ; v_{t-1} is the second moment estimate of the squared gradient at timestep $t - 1$; g_t is the gradient with respect to the stochastic objective at time t ; g_t^2 is the elementwise square of g_t ; β_1 is the hyperparameter, which controls the exponential decay rate for the first moment estimate; β_2 is the hyperparameter, which controls the exponential decay rate for the second-moment estimate; \hat{m}_t is the bias-corrected estimator for the first moment; \hat{v}_t is the bias-corrected estimator for the second moment; θ is the updated model parameter; α is the step size or learning rate hyperparameter; and ϵ is a parameter configured as a very small number to prevent any division by zero.

Table 5. Values of hyperparameters used in Adam optimizer.

Hyperparameter	Value
β_1	0.9
β_2	0.999
α	0.001
ϵ	1×10^{-7}

2.5. Model Evaluation and Validation

One way to justify how well the model works for the SRT dataset is to evaluate the model performance using standard statistical metrics. The model is estimated using a testing dataset and then returns the model validation metrics for both regression and classification problems. As the SRT problem falls into both classification and regression, the validation metrics, such as mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), coefficient of determination (R^2), and confusion matrix (CM), were used.

Among all the models created using a different set of hyperparameters, the one having the smallest MSE, RMSE, and MAE but the highest R^2 was selected to be the best fit for a regression problem. MSE and RMSE are the most popular metrics for the regression task because of the theoretical relevance in statistical modelling despite being sensitive to outliers [51]. RMSE poses a high penalty on large errors through the defined least-square terms, which implied that it is useful for improving the model performance, especially when the model errors follow a normal distribution [52].

MAE is the average of the absolute difference between the predicted and actual values. It is suitable to portray the errors that show a uniform distribution [51]. Besides, MAE is the most natural and precise measure of the average error magnitude [53].

R^2 is a scale-free score that does not provide the model residuals' information because it only determines the data's dispersion, not the bias [54]. The computation of the regression validation metrics is shown in Equations (14)–(17).

$$MSE = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (14)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (15)$$

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (16)$$

$$R^2 = 1 - \frac{\sum_{j=1}^n (\hat{y}_j - \bar{y})^2}{\sum_{j=1}^n (y_j - \bar{y})^2} \quad (17)$$

In Equations (14)–(17); n is the total number of observations in the dataset; y_j is the actual value for the j^{th} observation; \hat{y}_j is the predicted value for the j^{th} observation; and \bar{y} is the mean of the actual value.

In contrast, for the classification problem, the highest classification accuracy (ACC) was selected as the best fit model. The confusion matrix consists of four components, which are the true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The ACC can be calculated using CM's component, as shown in Equation (18):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (18)$$

In Equation (18), TP is when the actual and predicted output are both 1; TN is when the actual and predicted output are both 0; FP is when the actual output is 0, but the predicted output is 1; and FN is when the actual output is 1, but the predicted output is 0. The result of the 1D-CNN models for each set in Table 2 is shown in Section 3.

3. Results and Discussion

3.1. Slurry Test Model Validation Result

The SRT dataset for the slurry test was divided into four sets, as shown Table 2, where each set was fitted separately by the output variables. Therefore, the prediction of

the amount of sand produced in grams, retained permeability and the amount of sand produced per unit area, and the classification of the plugging sign were evaluated. The best model among all six hyperparameter configurations was selected according to the performance metrics. The validation results for the classification problem are shown in Table 6, while the regression problem is presented in Tables 7–9.

Table 6. Model validation results of set 1 for the slurry test with PLUG_SIGN as the output.

Model	TN	FP	FN	TP	ACC
1	42	3	1	3	0.918
2	41	1	1	6	0.959
3	42	2	2	3	0.918
4	43	2	2	2	0.918
5	46	0	2	1	0.959
6	44	0	2	3	0.959

Table 7. Model validation results of set 2 for the slurry test with SAND_PRODUCED as the output.

Model	MSE	RMSE	MAE	R ²
1	0.508	0.713	0.456	0.625
2	0.704	0.839	0.482	0.649
3	0.669	0.818	0.435	0.621
4	0.470	0.686	0.367	0.712
5	0.330	0.575	0.375	0.766
6	0.374	0.612	0.325	0.747

Table 8. Model validation results of set 3 for the slurry test with RETAINED_PRM as the output.

Model	MSE	RMSE	MAE	R ²
1	117.217	10.827	6.485	0.957
2	70.616	8.403	5.438	0.978
3	30.121	5.488	4.616	0.990
4	79.436	8.913	5.415	0.966
5	18.240	4.271	2.411	0.993
6	21.980	4.688	3.889	0.990

Table 9. Model validation results of set 4 for the slurry test with SAND_PROD_PER_AREA as the output.

Model	MSE	RMSE	MAE	R ²
1	0.065	0.255	0.158	0.553
2	0.041	0.203	0.121	0.613
3	0.089	0.299	0.162	0.579
4	0.077	0.277	0.150	0.585
5	0.032	0.179	0.096	0.772
6	0.032	0.179	0.109	0.761

The model validation results of the slurry test presented above showed that the hyperparameter configuration in model 5 gave the lowest average errors and the highest R^2 in predicting sand production and retained permeability. However, the classification model for plugging sign gave the best result using a combination of hyperparameters in model 2, 5, and 6. Therefore, model 5 is selected as the best model for predicting sand production and retained permeability of the slurry test, but either model 2, 5, or 6 can be chosen as the final model for the plugging sign. The line plot of model 5 was used for the plugging sign to visualize the trend between actual and predicted data. The actual versus predicted plots of model 5 for each slurry test set are shown in Figure 6.

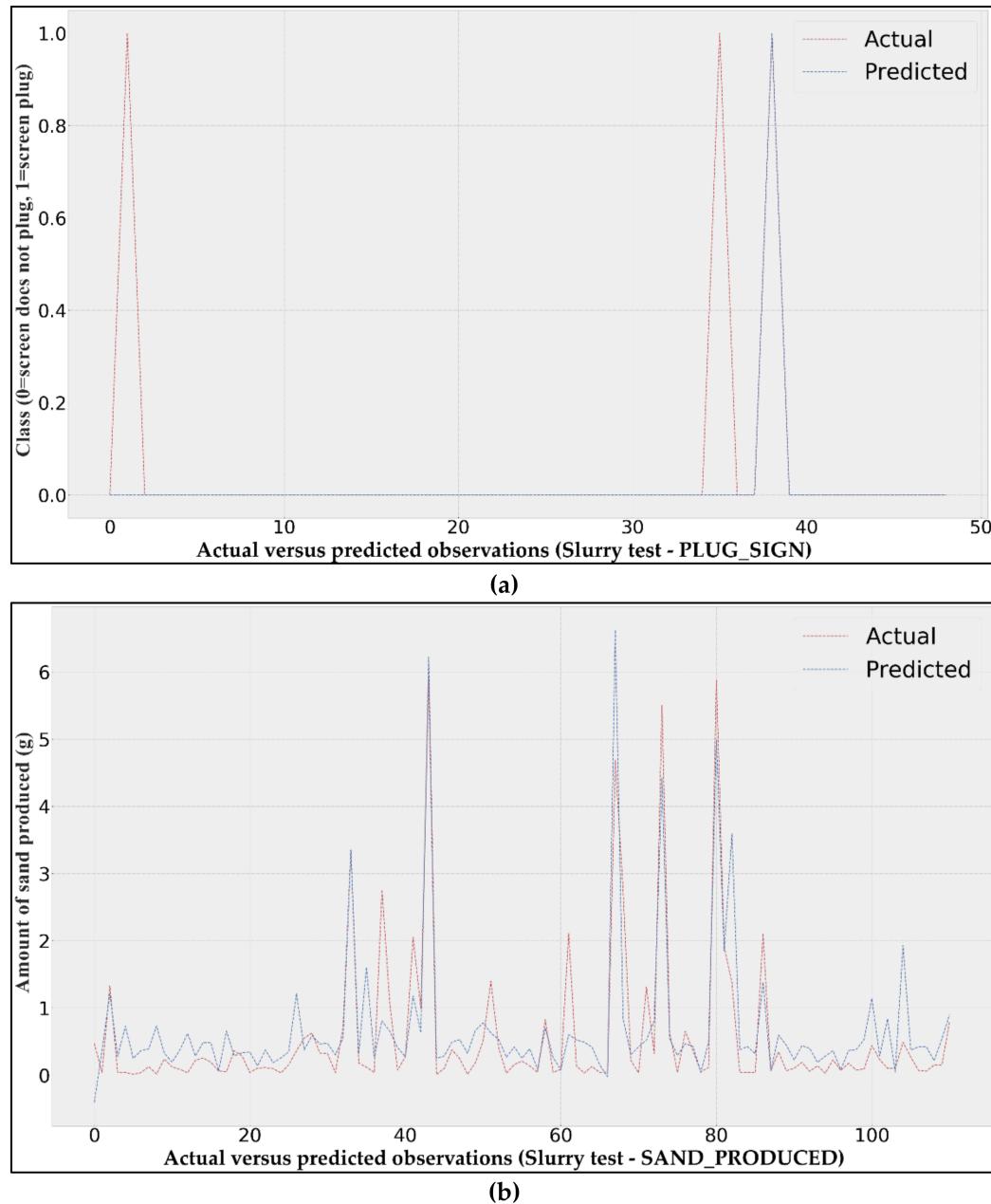


Figure 6. Cont.

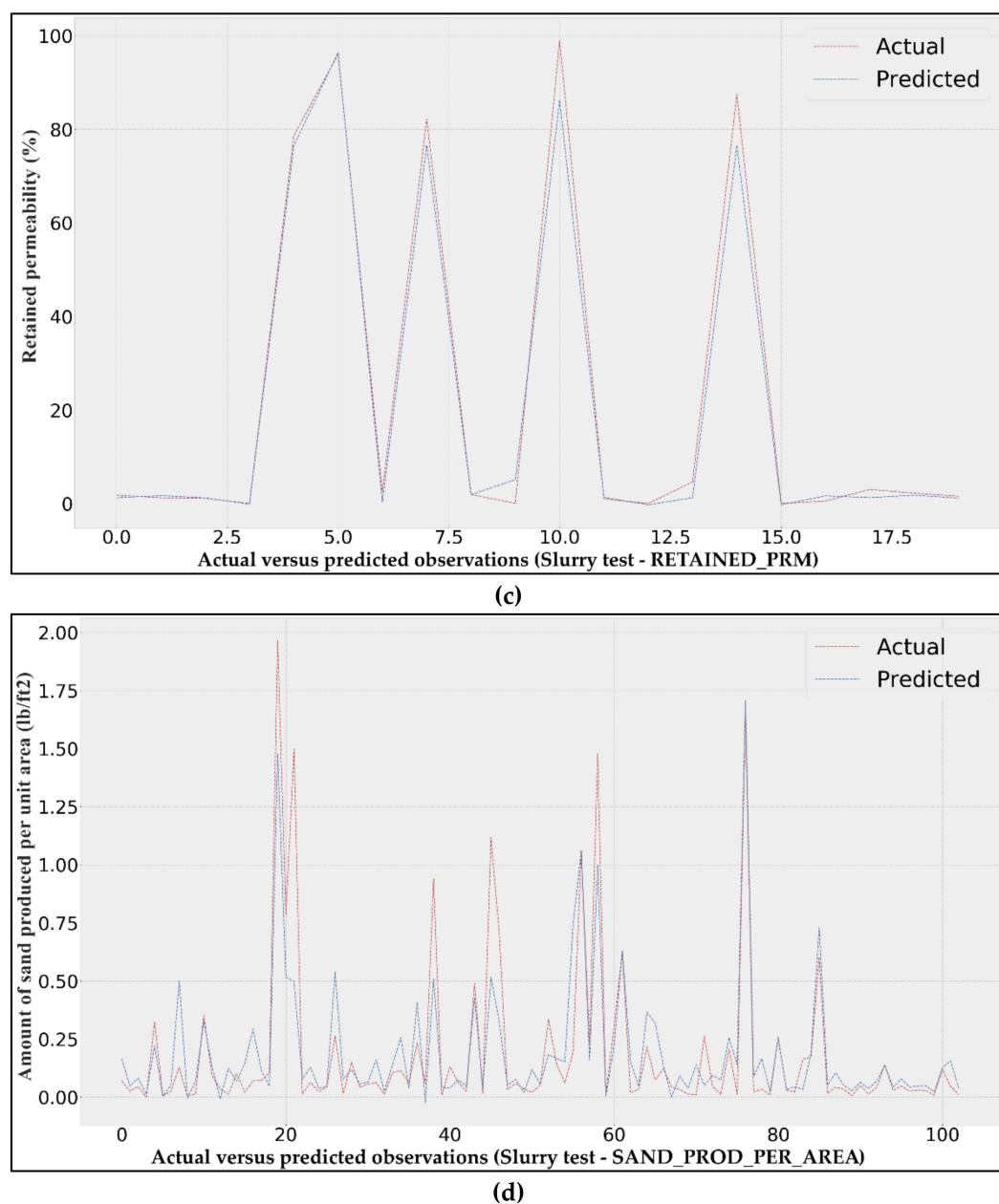


Figure 6. The actual versus predicted plot of model 5 for the slurry test: (a) The plot for plugging sign; (b) the plot for the amount of sand produced in grams; (c) the plot for retained permeability; (d) the plot for the amount of sand produced per unit area.

In Figure 6, the trend of actual data is shown by the red line, while the blue line shows the predicted data for all slurry test sets. The line plot of the plugging sign in Figure 6a represents the classification of 49 observations of the actual and predicted data. The observations that fall under the 0 class refer to the observations with no sign of screen plugging, while the observations that were categorized as 1 refer to a sign of screen plugging. Two peaks at the red line represent actual data with a sign of screen plugging, but the blue line does not follow the peaks that portray the false classification of two observations, as shown in Table 5. The line plot of sand production in Figure 6b represents 111 predicted values along with the actual data. The predicted amount of sand produced in grams approximates the actual trend with 77% accuracy.

Furthermore, the line plot of retained permeability in Figure 6c represents 20 observations of screen-retained permeability. The prediction line closely follows the actual

percentage of retained permeability with 99% accuracy. Lastly, Figure 6d shows a line plot of sand produced per unit area with 103 observations and 77% accuracy. Most of the prediction points are almost identical to the actual points.

3.2. Sand Pack Test Model Validation Result

The SRT dataset for the sand pack test has only one set, as shown in Table 2, where the dataset was fitted only to predict the amount of sand produced per unit area. The best model among all six hyperparameter configurations was selected according to the performance metrics. The validation result for the regression problem of the sand pack test is shown in Table 10.

Table 10. Model validation result of set 5 for the sand pack test with SAND_PROD_PER_AREA as the output.

Model	MSE	RMSE	MAE	R ²
1	0.007	0.083	0.060	0.837
2	0.009	0.096	0.052	0.821
3	0.011	0.104	0.063	0.724
4	0.019	0.136	0.068	0.664
5	0.012	0.111	0.056	0.686
6	0.027	0.166	0.088	0.530

The model validation result for the sand pack test presented above showed that the hyperparameter configuration in model 1 gave the lowest MSE and RMSE but the second lowest for MAE. Furthermore, it provided the highest R² in predicting sand production. Hence, model 1 was selected as the best model to predict sand production for the sand pack test. The actual versus predicted plot of model 1 for the sand pack test is shown in Table 10.

In Figure 7, the actual amount of sand produced per unit area is portrayed by the red line, while predicted values are shown by the blue line with 79 observations and 84% accuracy. The prediction points of sand production below 0.2 lb/ft² do not approximately follow the actual points, but the points above 0.2 lb/ft² are almost identical.

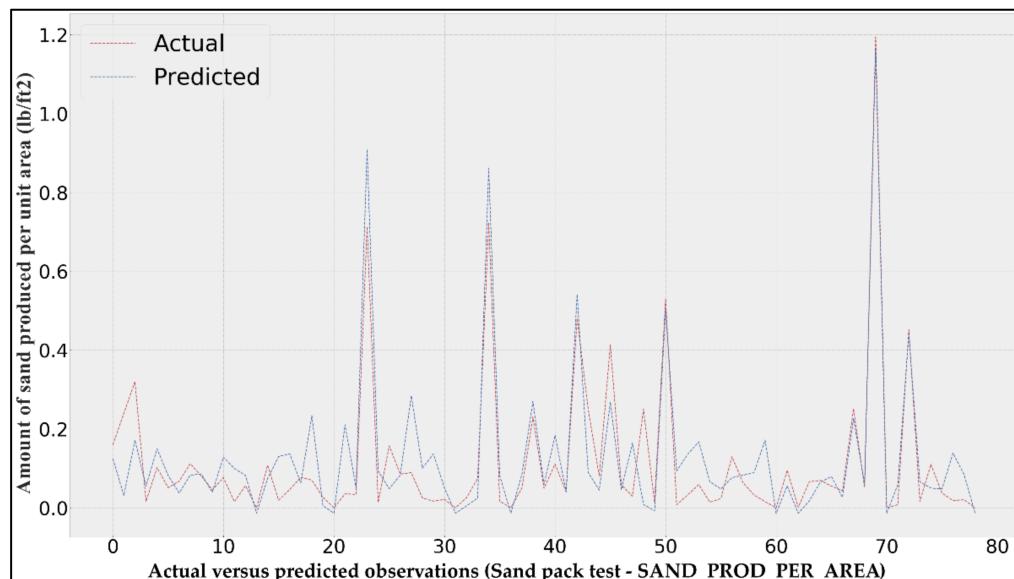


Figure 7. The actual versus predicted plot of model 1 of sand production for the sand pack test.

3.3. Comparative Model Performance

The performance of different machine learning or deep learning algorithms may vary depending on the datasets. Generally, deep learning algorithms outperform shallow machine learning techniques. To validate this statement, four machine learning models, which are gradient boosting (GB), K-nearest neighbor (KNN), random forest (RF), and support vector machine (SVM), were developed for the SRT problem using the same dataset as in Table 2.

All those four machine learning models are supervised learning which commonly used to solve a classification problem but still can be used for regression. The details of four comparative models are shown in Table 11. The model validation results for the slurry test can be seen in Tables 12–16, while Table 14 is the result of the sand pack test.

Table 11. The details of four comparative models.

Model	Description	Hyperparameter (value)	Formula
GB	GB's prediction used ensemble method with gradient descent to minimize the loss function [55].	Loss function [classification: binary_crossentropy, regression: (east_squares)]	$\hat{F} = \underset{F}{\operatorname{argmin}} \mathbb{E}_{x,y}[L(y, F(x))]$ $L(y, F(x))$ is the loss function; $\underset{F}{\operatorname{argmin}}$ is the argument of function F that minimizes the expected value of loss function.
		Learning rate (0.1)	
		Maximum number of trees (100)	
		Maximum number of leaves for each tree (31)	
		Minimum number of samples per leaf (20)	
KNN	KNN cast the prediction by the weighted average of the targets according to its neighbor's closest distance [56].	Number of neighbors (3)	$y(d_i) = \underset{k}{\operatorname{argmax}} \sum_{x_j \in kNN} y(x_j, c_k)$ d_i is a test example; x_j is one of the KNN in training set; $y(x_j, c_k)$ is an indicator that the training set, x_j belongs to class c_k $\underset{k}{\operatorname{argmax}}$ is the argument of function k that gives the maximum predicted probability.
		Weight function used for prediction (uniform)	
		Algorithm used to compute the nearest neighbors (auto)	
		Leaf size (30)	
		Power parameter (2)	
		Distance metric used for the tree (minkowski)	
RF	RF represents a boosting method consisting of a collection of decision trees where a majority vote acquires the prediction over each tree's predictions [57].	Number of trees (1000)	$g(x) = \frac{1}{B} \sum_{i=1}^B f_i(x')$ B is the total number of trees; $f_i(x')$ is the prediction of the individual tree at i^{th} node.
		Function to measure the quality of a split (classification: Gini, regression: MSE)	
		Minimum number of samples required to split (2)	
		Minimum number of samples required to be at a leaf node (1)	
		Minimum weighted fraction of total weights required to be at a leaf node (0)	
		Number of features to consider when looking for the best split (auto)	
		Random state (42)	
SVM	The prediction of SVM depends on the hyperplane and the decision boundary in multidimensional space where the algorithm will find the best fit line with a maximum number of points [58].	Regularization parameter (1)	$R(w, b) = C \times \sum_{i=1}^n \ell(y_i, f(x_i)) + \frac{1}{2} \ w\ ^2$ C is the regularization parameter; $\ell(y_i, f(x_i))$ is the loss function; $\ w\ $ is the hyperplane with a maximum margin.
		Kernel (classification: linear, regression: RBF)	
		Stopping criteria tolerance (1×10^{-3})	
		Hard limit on iterations within solver (-1)	
		Kernel coefficient for RBF [regression only: scale]	
		Epsilon (regression only: 0.1)	

Table 12. Comparative model validation result of the slurry test with PLUG_SIGN as the output.

Model	TN	FP	FN	TP	Accuracy
1D-CNN	41	1	1	6	0.959
GB	35	1	2	3	0.927
KNN	34	1	2	4	0.927
RF	33	2	3	3	0.878
SVM	31	2	4	4	0.854

Table 13. Comparative model validation result of the slurry test with SAND_PRODUCED as the output.

Model	MSE	RMSE	MAE	R ²
1D-CNN	0.330	0.575	0.375	0.766
GB	0.339	0.594	0.395	0.714
KNN	0.437	0.661	0.396	0.677
RF	0.901	0.949	0.422	0.573
SVM	0.551	0.742	0.514	0.578

Table 14. Comparative model validation result of the slurry test with RETAINED_PRM as the output.

Model	MSE	RMSE	MAE	R ²
1D-CNN	18.240	4.271	2.411	0.993
GB	288.692	16.990	13.900	0.845
KNN	18.570	4.309	2.535	0.990
RF	77.758	8.818	5.854	0.954
SVM	83.300	9.126	8.335	0.904

Table 15. Comparative model validation result of the slurry test with SAND_PROD_PER_AREA as the output.

Model	MSE	RMSE	MAE	R ²
1D-CNN	18.240	4.271	2.411	0.772
GB	288.692	16.990	13.900	0.548
KNN	18.570	4.309	2.535	0.564
RF	77.758	8.818	5.854	0.735
SVM	83.300	9.126	8.335	0.624

Table 16. Comparative model validation result of the sand pack test with SAND_PROD_PER_AREA as the output.

Model	MSE	RMSE	MAE	R ²
1D-CNN	0.007	0.083	0.060	0.837
GB	0.023	0.153	0.086	0.578
KNN	0.014	0.118	0.079	0.603
RF	0.014	0.119	0.092	0.292
SVM	0.026	0.162	0.122	0.570

The results showed that the 1D-CNN model outperformed all four machine learning models for both the slurry and sand pack tests. It justifies that deep learning algorithms perform better than shallow machine learning methods.

3.4. Significance Analysis of Comparative Models

The significant difference between the means of all comparative models was tested using one-way analysis of variance (ANOVA) because it is a parametric test for more than two samples. However, the Kruskal–Wallis test was used to identify the significant difference between the medians of five comparative models since it is a non-parametric test for more than two samples. The verification of the statistically significant models comes

with the condition where the null hypothesis is rejected. The null hypothesis refers to the situation where the mean or median difference between all five models is not statistically significant. Suppose the p -value is less than the significance level of 0.05, and the F statistic value is greater than the F critical value. In this case, the null hypothesis was rejected for the ANOVA test. Still, if the H statistic value is greater than the Chi-square critical value, then the null hypothesis is rejected for the Kruskal–Wallis test. The significance analysis using the ANOVA and Kruskal–Wallis tests are presented Table 17.

Table 17. Significance analysis using ANOVA and Kruskal–Wallis tests.

Output	Test	Test Statistic	p -Value	Critical Value
Slurry–PLUG_SIGN	Kruskal–Wallis	12.2000	0.0159	9.4677
Slurry–SAND_PRODUCED	ANOVA	11.0518	1.254×10^{-8}	2.3881
Slurry–RETAINED_PRM	ANOVA	35.4468	4.254×10^{-18}	2.4675
Slurry–SAND_PROD_PER_AREA	ANOVA	23.9964	3.307×10^{-18}	2.3894
Sand pack–SAND_PROD_PER_AREA	ANOVA	35.3620	3.283×10^{-25}	2.3948

The significance analysis using ANOVA and Kruskal–Wallis tests in Table 17 showed that the p -values are less than 0.05 and the test statistics are greater than the critical values. Thus, the null hypothesis is rejected, indicating that the differences between the mean or median of all five models in predicting the plugging sign, retained permeability, and sand production are statistically significant.

4. Conclusions

This paper proposed a 1D-CNN with adaptive moment estimation for modelling of the SRT dataset focusing on the slurry and sand pack tests that use stand-alone screens as a sand control method to reduce sand production. The proposed method was developed to examine the sand retention efficiency and plugging performance, which can help select the optimal screen and slot size.

The hyperparameters tuning of 1D-CNN was empirically performed using a trial-and-error approach, leading to the development of six models tested for each output. The 1D-CNN model performance showed that the hyperparameters configuration in model 5 of the slurry test fits the best in predicting the amount of sand produced in grams, retained permeability, and the amount of sand produced per unit area because model 5 gave the lowest average errors and the highest R^2 . The best model for both sand productions gave an accuracy of 77%, while the best model for retained permeability gave 99% accuracy. Besides, the set of hyperparameters in model 2, 5, and 6 fit very well with the classification of the plugging sign because all three models gave the same accuracy of 96%. Thus, either model 2, 5, or 6 can be used as the best fit model. In contrast, model 1 of the sand pack test outperformed the other five models in predicting the amount of sand produced per unit area with an accuracy of 84%.

For comparative model performance, the accuracy of the 1D-CNN model was higher than the other four machine learning models in predicting all the outputs of slurry and sand pack tests. Therefore, the proposed deep learning model outperformed the other four machine learning methods based on validation metrics.

Since the proposed deep learning model is the first model developed for SRT problem, further optimization can be proposed for future research by focusing more on the feature engineering process and include more observations for the modelling phase.

Author Contributions: Conceptualization, N.N.A.R. and S.J.A.; methodology, N.N.A.R. and S.J.A.; software, N.N.A.R. and M.G.R.; validation, N.N.A.R., S.J.A., and S.N.A.S.; formal analysis, N.N.A.R. and M.G.R.; data curation, N.N.A.R. and S.N.A.S.; writing—original draft preparation, N.N.A.R.;

writing—review and editing, N.N.A.R. and S.J.A.; visualization, N.N.A.R.; supervision, S.J.A. and M.A.M.; funding acquisition, S.J.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by PETRONAS Group Research & Technology and Universiti Teknologi PETRONAS funded by GR&T-UTP Collaboration Grant with UTP grant number 015MD0-24.

Informed Consent Statement: “Not applicable” for studies not involving humans.

Data Availability Statement: The data that support the findings of this study are available from PETRONAS GR&T. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from the authors with the permission of PETRONAS Group Research & Technology (GR&T).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Wu, B.; Choi, S.; Feng, Y.; Denke, R.; Barton, T.; Wong, C.; Boulanger, J.; Yang, W.; Lim, S.; Zamberi, M. Evaluating sand screen performance using improved sand retention test and numerical modelling. In Proceedings of the Offshore Technology Conference Asia, Offshore Technology Conference, Kuala Lumpur, Malaysia, 22–25 March 2016.
- Chanpura, R.A.; Mondal, S.; Andrews, J.S.; Mathisen, A.M.; Ayoub, J.A.; Parlar, M.; Sharma, M.M. Modeling of square mesh screens in slurry test conditions for stand-alone screen applications. In Proceedings of the SPE International Symposium and Exhibition on Formation Damage Control, Society of Petroleum Engineers, Lafayette, LA, USA, 15–17 February 2012.
- Ballard, T.; Beare, S. Media sizing for premium sand screens: Dutch twill weaves. In Proceedings of the SPE European Formation Damage Conference, Society of Petroleum Engineers, The Hague, The Netherlands, 13–14 May 2003.
- Ballard, T.; Beare, S.P. Sand retention testing: The more you do, the worse it gets. In Proceedings of the SPE International Symposium and Exhibition on Formation Damage Control, Society of Petroleum Engineers, Lafayette, LA, USA, 15–17 February 2006.
- Markestad, P.; Christie, O.; Espedal, A.; Rørvik, O. Selection of screen slot width to prevent plugging and sand production. In Proceedings of the SPE Formation Damage Control Symposium, Society of Petroleum Engineers, Lafayette, LA, USA, 14–15 February 1996.
- Ballard, T.J.; Beare, S.P. An investigation of sand retention testing with a view to developing better guidelines for screen selection. In Proceedings of the SPE International Symposium and Exhibition on Formation Damage Control, Society of Petroleum Engineers, Lafayette, LA, USA, 15–17 February 2012.
- Ballard, T.; Beare, S.; Wigg, N. Sand Retention Testing: Reservoir Sand or Simulated Sand—Does it Matter? In Proceedings of the SPE International Conference and Exhibition on Formation Damage Control, Society of Petroleum Engineers, Lafayette, LA, USA, 24–26 February 2016.
- Agunloye, E.; Utunedi, E. Optimizing sand control design using sand screen retention testing. In Proceedings of the SPE Nigeria Annual International Conference and Exhibition, Society of Petroleum Engineers, Victoria Island, Nigeria, 5–7 August 2014.
- Mathisen, A.M.; Aastveit, G.L.; Alteraas, E. Successful installation of stand alone sand screen in more than 200 wells—the importance of screen selection process and fluid qualification. In Proceedings of the European Formation Damage Conference, Society of Petroleum Engineers, The Hague, The Netherlands, 30 May–1 June 2007.
- Fischer, C.; Hamby, H. A Novel Approach to Constant Flow-Rate Sand Retention Testing. In Proceedings of the SPE International Conference and Exhibition on Formation Damage Control, Society of Petroleum Engineers, Lafayette, LA, USA, 7–9 February 2018.
- Ma, C.; Deng, J.; Dong, X.; Sun, D.; Feng, Z.; Luo, C.; Xiao, Q.; Chen, J. A new laboratory protocol to study the plugging and sand control performance of sand control screens. *J. Pet. Sci. Eng.* **2020**, *184*, 106548.
- Torrisi, M.; Pollastri, G.; Le, Q. Deep learning methods in protein structure prediction. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 1301–1310. [[PubMed](#)]
- Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **2019**, *151*, 107398.
- Abdulkadir, S.J.; Alhussian, H.; Nazmi, M.; Elsheikh, A.A. Long Short Term Memory Recurrent Network for Standard and Poor’s 500 Index Modelling. *Int. J. Eng. Technol.* **2018**, *7*, 25–29. [[CrossRef](#)]
- Abdulkadir, S.J.; Yong, S.P.; Marimuthu, M.; Lai, F.W. Hybridization of ensemble Kalman filter and nonlinear auto-regressive neural network for financial forecasting. In *Mining Intelligence and Knowledge Exploration*; Springer: York, NY, USA, 2014; pp. 72–81.
- Abdulkadir, S.J.; Yong, S.P. Empirical analysis of parallel-NARX recurrent network for long-term chaotic financial forecasting. In Proceedings of the 2014 International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 3–5 June 2014; pp. 1–6.
- Abdulkadir, S.J.; Yong, S.P.; Zakaria, N. Hybrid neural network model for metocean data analysis. *J. Inform. Math. Sci.* **2016**, *8*, 245–251.

18. Abdulkadir, S.J.; Yong, S.P. Scaled UKF–NARX hybrid model for multi-step-ahead forecasting of chaotic time series data. *Soft Comput.* **2015**, *19*, 3479–3496.
19. Abdulkadir, S.J.; Yong, S.P.; Alhussian, H. An enhanced ELMAN-NARX hybrid model for FTSE Bursa Malaysia KLCI index forecasting. In Proceedings of the 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 15–17 August 2016; pp. 304–309.
20. Abdulkadir, S.J.; Yong, S.P. Lorenz time-series analysis using a scaled hybrid model. In Proceedings of the 2015 International Symposium on Mathematical Sciences and Computing Research (iSMSC), Ipoh, Malaysia, 19–20 May 2015; pp. 373–378.
21. Yoo, Y. Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches. *Knowl. Based Syst.* **2019**, *178*, 74–83.
22. Mhiri, M.; Abuelwafa, S.; Desrosiers, C.; Cheriet, M. Footnote-based document image classification using 1D convolutional neural networks and histograms. In Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, Canada, 28 November–1 December 2017; pp. 1–5.
23. Pereira, S.; Pinto, A.; Alves, V.; Silva, C.A. Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans. Med. Imaging* **2016**, *35*, 1240–1251.
24. Shin Yoon, J.; Rameau, F.; Kim, J.; Lee, S.; Shin, S.; So Kweon, I. Pixel-level matching for video object segmentation using convolutional neural networks. In Proceedings of the IEEE international conference on computer vision, Venice, Italy, 22–29 October 2017; pp. 2167–2176.
25. Aszemi, N.M.; Dominic, P. Hyperparameter optimization in convolutional neural network using genetic algorithms. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 269–278.
26. Souza, J.F.L.; Santana, G.L.; Batista, L.V.; Oliveira, G.P.; Roemers-Oliveira, E.; Santos, M.D. CNN Prediction Enhancement by Post-Processing for Hydrocarbon Detection in Seismic Images. *IEEE Access* **2020**, *8*, 120447–120455.
27. Zhou, X.; Zhao, C.; Liu, X. Application of CNN Deep Learning to Well Pump Troubleshooting via Power Cards. In Proceedings of the Abu Dhabi International Petroleum Exhibition & Conference, Society of Petroleum Engineers, Abu Dhabi, UAE, 11–14 November 2019.
28. Daolun, L.; Xuliang, L.; Wenshu, Z.; Jinghai, Y.; Detang, L. Automatic well test interpretation based on convolutional neural network for a radial composite reservoir. *Pet. Explor. Dev.* **2020**, *47*, 623–631.
29. Kwon, S.; Park, G.; Jang, Y.; Cho, J.; Chu, M.-g.; Min, B. Determination of Oil Well Placement using Convolutional Neural Network Coupled with Robust Optimization under Geological Uncertainty. *J. Pet. Sci. Eng.* **2020**, *108*, 118.
30. Li, J.; Hu, D.; Chen, W.; Li, Y.; Zhang, M.; Peng, L. CNN-Based Volume Flow Rate Prediction of Oil–Gas–Water Three-Phase Intermittent Flow from Multiple Sensors. *Sensors* **2021**, *21*, 1245.
31. Chanpura, R.A.; Hodge, R.M.; Andrews, J.S.; Toffanin, E.P.; Moen, T.; Parlar, M. State of the art screen selection for stand-alone screen applications. In Proceedings of the SPE International Symposium and Exhibition on Formation Damage Control, Society of Petroleum Engineers, Lafayette, LA, USA, 10–12 February 2010.
32. Hodge, R.M.; Burton, R.C.; Constien, V.; Skidmore, V. An evaluation method for screen-only and gravel-pack completions. In Proceedings of the International Symposium and Exhibition on Formation Damage Control, Society of Petroleum Engineers, Lafayette, LA, USA, 20–21 February 2002.
33. Gillespie, G.; Deem, C.K.; Malbrel, C. Screen selection for sand control based on laboratory tests. In Proceedings of the SPE Asia Pacific Oil and Gas Conference and Exhibition, Society of Petroleum Engineers, Brisbane, Australia, 16–18 October 2000.
34. Chanpura, R.A.; Fidan, S.; Mondal, S.; Andrews, J.S.; Martin, F.; Hodge, R.M.; Ayoub, J.A.; Parlar, M.; Sharma, M.M. New analytical and statistical approach for estimating and analyzing sand production through wire-wrap screens during a sand-retention test. *SPE Drill. Completion* **2012**, *27*, 417–426.
35. Constien, V.G.; Skidmore, V. Standalone screen selection using performance mastercurves. In Proceedings of the SPE International Symposium and Exhibition on Formation Damage Control, Society of Petroleum Engineers, Lafayette, LA, USA, 15–17 February 2006.
36. Mondal, S.; Sharma, M.M.; Hodge, R.M.; Chanpura, R.A.; Parlar, M.; Ayoub, J.A. A new method for the design and selection of premium/woven sand screens. In Proceedings of the SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers, Denver, CO, USA, 30 October–2 November 2011.
37. Heumann, C.; Schomaker, M. *Introduction to Statistics and Data Analysis*; Springer: New York, NY, USA, 2016; p. 83.
38. Suchmacher, M.; Geller, M. Correlation and Regression. In *Practical Biostatistics: A Friendly Step-by-Step Approach For Evidence-based Medicine*; Academic Press: Waltham, MA, USA, 2012; pp. 167–186.
39. Forsyth, D. *Probability and Statistics for Computer Science*; Springer: New York, NY, USA, 2018.
40. Bonamente, M. *Statistics and Analysis of Scientific Data*; Springer: New York, NY, USA, 2017; p. 187.
41. Swinscow, T. Statistics at square one: XVIII-Correlation. *Br. Med. J.* **1976**, *2*, 680. [[CrossRef](#)]
42. Chollet, F. *Deep Learning with Python*; Manning Publication, Co.: New York, NY, USA, 2018; Volume 361.
43. Abdulkadir, S.J.; Yong, S.P.; Foong, O.M. Variants of Particle Swarm Optimization in Enhancing Artificial Neural Networks. *Aust. J. Basic Appl. Sci.* **2013**, *7*, 388–400.
44. Khan, S.; Rahmani, H.; Shah, S.A.A.; Bennamoun, M. A guide to convolutional neural networks for computer vision. In *Synthesis Lectures on Computer Vision*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2018; Volume 8, pp. 1–207.

45. Michelucci, U. *Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection*; Springer: New York, NY, USA, 2019.
46. Pysal, D.; Abdulkadir, S.J.; Shukri, S.R.M.; Alhussian, H. Classification of children's drawing strategies on touch-screen of seriation objects using a novel deep learning hybrid model. *Alex. Eng. J.* **2021**, *60*, 115–129.
47. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629.
48. Aggarwal, C.C. *Neural Networks and Deep Learning*; Springer: New York, NY, USA, 2018; Volume 10, pp. 978–983.
49. Li, Y.; Zou, L.; Jiang, L.; Zhou, X. Fault diagnosis of rotating machinery based on combination of deep belief network and one-dimensional convolutional neural network. *IEEE Access* **2019**, *7*, 165710–165723.
50. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. Available online: <https://arxiv.org/abs/1412.6980> (accessed on 1 January 2021).
51. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE). *Geosci. Model Dev. Discuss.* **2014**, *7*, 1525–1534.
52. Hyndman, R.J.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688.
53. Willmott, C.J.; Matsuura, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.* **2005**, *30*, 79–82.
54. Onyutha, C. From R-squared to coefficient of model accuracy for assessing "goodness-of-fits". *Geosci. Model. Dev. Discuss.* **2020**, 1–25. [CrossRef]
55. He, Z.; Lin, D.; Lau, T.; Wu, M. Gradient Boosting Machine: A Survey. *arXiv* **2019**, arXiv:1908.06951. Available online: <https://arxiv.org/abs/1908.06951> (accessed on 19 March 2021).
56. Barber, D. *Bayesian Reasoning and Machine Learning*; Cambridge University Press: Cambridge, UK, 2012.
57. Shalev-Shwartz, S.; Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*; Cambridge University Press: Cambridge, UK, 2014.
58. Kubat, M.A. *Introduction to Machine Learning*; Springer: New York, NY, USA, 2017.