Journal of Business, Social and Technology
Volume 7, Issue 2, 269-283
e_ISSN: 2807-6362
https://bustechno.polteksci.ac.id/index.php/jbt/index
DOI: doi.org/10.59261/jbt.v7i2.607

Yeo-Johnson Transformation Usage in Data Preprocessing for Well Production Prediction Using Deep Neural Networks (DNN)

***Alringga Rizky[1]**
Institut Teknologi Sepuluh Nopember, Indonesia

Anny Yuniarti[2]
Institut Teknologi Sepuluh Nopember, Indonesia

***Corresponding author:**
Anny Yuniarti, Institut Teknologi Sepuluh Nopember, anny@its.ac.id

Article Info:

Article history:
Received: March 05, 2026
Revised: April 07, 2026
Accepted: April 09, 2026

Keywords:
data preprocessing; deep neural networks (DNN); tree-structured parzen estimator (TPE); well production prediction; Yeo-Johnson transformation

Abstract

Background: The accurate prediction of infill well production is one of the major bottlenecks for hydrocarbon reservoir development. Traditional reservoir simulation tools are computationally expensive, taking weeks to months per scenario.
Objective: This paper presents the development of a Deep Neural Network (DNN) model for prediction with hyperparameter optimization using the Tree-structured Parzen Estimator (TPE) to predict pay porosity (PORPAYX) in infill wells of the Pertamina Hulu Sanga Sanga field.
Methods: A DNN model was developed to predict oil well production based on subsurface and production features from a comprehensive dataset of Pertamina Hulu Sanga Sanga reservoir characteristics and production data. Details of our method include: training the model on a robust dataset, hyperparameter tuning using the Tree-structured Parzen Estimator (TPE), and K-fold cross-validation for performance validation.
Results: Scaling normalized the data in such a way that every feature had equal influence during model training, enabling better learning and accurate prediction. In contrast, fitting the model using unscaled data resulted in an R^2 of less than zero (a negative score), meaning that the model could not explain the variability in the data. The mean R^2 score of the unscaled data model was −0.08496, along with a higher MSE = 0.009057 and RMSE = 0.095148. This was due to the model's failure to process features with varying scales, which prevented proper learning and prediction.
Conclusion: Residual plots confirmed that the model trained with scaled data met the assumptions of linearity and normality.

To cite this article: Alringga Rizky, & Yuniarti, A. (2026). Yeo-johnson transformation usage in data preprocessing for well production prediction using deep neural networks (DNN). *Journal of Business, Social and Technology*, 7(2), 269–283. https://doi.org/10.59261/jbt.v7i2.607

INTRODUCTION

One of the main aims for the application of infill drilling strategies in petroleum reservoirs is to maximize hydrocarbon recovery. Reservoir simulation is the current benchmark methodology for decision support that informs these strategic development choices and plans. Nonetheless, these classical software-based simulations provide high accuracy while requiring extended computation times and considerable computing resources in order to perform. As such, far less expensive alternatives like kriging, regression, and neural networks have been incorporated into workflows to aid or even substitute for numerical modeling. Artificial neural networks (ANNs) have been utilized in petroleum engineering since the 1990s, particularly for their ability to capture the complex, non-linear relationships present in extensive datasets.

This study proposes an artificial intelligence (AI)-based solution aimed at reducing computational complexity to achieve fast runtime and reliable production predictions. Specific subsurface and production characteristics are used within machine learning and deep learning

algorithms using historical data to estimate well output (September et al., 2024). The dataset used for this analysis consists of reservoir simulation samples from wells obtained from Pertamina Hulu Sanga Sanga, an upstream oil company.

The critical need for efficient prediction tools is further emphasized by international industry data: according to the International Energy Agency (IEA, 2023), maintaining current production from mature fields will demand more than 3,000 infill wells across the world every year through 2030 alone—establishing scalable screening tools as a critical operational need. Deep architectures can model non-linear relationships in reservoirs, and since the 1990s, multi-layered artificial neural networks have been applied in petroleum engineering LeCun (2015); recent works showed deep architectures outperforming classical regression baselines in several production forecasting tasks (Hu et al., 2025; Jia et al., 2024).

This study thus sets out to achieve three specific goals: (1) assess the potential of Yeo-Johnson transformation for improvement of predictive performance as measured by R^2, MSE, and RMSE metrics using 5-fold cross-validation; (2) evaluate the efficiency of TPE-based hyperparameter optimization in converging to DNN configurations that exhibit superior predictive performance when trained with scaled vs. unscaled input data; and finally (3) demonstrate a contextual knowledge-based approach that seeks to diagnose any heteroscedastic nature in the residual patterns from the optimized model and propose novel targeted architectural refinements addressing them.

Recent works have started to use complex machine learning frameworks to predict reservoir outputs and improve forecasting procedures. For sectors dealing with time-series data, a hybrid neural network combining CNN and LSTM has been leveraged for production forecasting (Chu et al., 2020). In this approach, the CNN framework enables the extraction of salient spatial features that are subsequently processed by an LSTM to learn temporal dependencies, resulting in a robust prediction model with a root mean squared logarithmic error (RMSLE) score for estimating well production reaching 0.186891 (Davtyan et al., 2020).

Although previous studies have incorporated CNN-LSTM hybrids Chu (2020); Abed (2026), RNN-LSTM combinations, and transformer architectures Abdrakhmanov (2021); Jia (2024) into petroleum production forecasting, three significant gaps remain. First, none of these studies uniquely isolate the contribution of distribution normalization as an independent variable on DNN generalizability—most view preprocessing through a classical lens and treat it more like a routine step than a subject of research. Second, the Yeo–Johnson power transformation has not yet been investigated as a preprocessing strategy for petroleum reservoir data that are highly skewed and multimodal in feature distributions. And third, the combined DNN + TPE framework has not been applied in any specific infill well screening context for Indonesian upstream petroleum assets. This study fills all three gaps by treating preprocessing itself as the primary experimental variable and delivering a first controlled comparison of Yeo-Johnson-transformed versus raw-input DNN performance on a real Pertamina Hulu Sanga Sanga dataset in 5-fold cross-validation.

In the same vein, RNN-oriented LSTM structures have been used to model multi-phase fluid flow—namely oil, gas, and water—by learning from historical injection data (Smith, 2018). This particular RNN-LSTM approach showcased an unprecedented level of accuracy, exceeding 90% for the first one-year prediction and achieving a stable accuracy rate of 73.63% over five years. Statistical machine learning has quantified the effect of geological parameters and well-completion strategies on production, beyond neural network approaches alone. One study applied Pearson correlation and Grey Relational Analysis to single out key production variables by analyzing 159 horizontal wells within the Duvernay Formation (Zhang et al., 2016). These key variables were subsequently modeled and predicted using Gaussian Process Regression (GPR), Support Vector Regression (SVR), and Multiple Linear Regression (MLR) for cumulative hydrocarbon production over a 6-month operational period.

In this study, we present the performance of a Tree-structured Parzen Estimator (TPE)-tuned Deep Neural Network (DNN) model to predict pay porosity (PORPAYX) in three infill wells drilled in the Pertamina Hulu Sanga Sanga field. What distinguishes this study is the finding that Yeo-Johnson power transformation—applied as a preprocessing step before DNN training—is the key factor in achieving strong model performance (avg. R^2 = 0.9477) over an effectively non-

predictive baseline (avg. R^2 = −0.085), offering the first empirical evidence for this specific preprocessing–performance interplay in reported petroleum infill production forecasts.

METHOD

Dataset

The dataset used in this research was sourced from the Pertamina Hulu Sanga Sanga field, which includes reservoir data and oil and gas production data, as can be seen in Table 1. Toad for Oracle was used as an interface to fetch the data from the database for model input, i.e., training, validation, and testing data. The input data for the model was then created from the query results in this data format and saved as a CSV file. Then, for the prediction input data, data obtained from infill wells were used with a distance condition of 100 m, where production well and subsurface data were queried to retrieve the offset well's production and subsurface information for neighboring well production.

Table 1. Datasets

No	Datasets Column	Details
1	WELL	List of infill wells with a distance condition of 100m
2	MONBOE	Monthly oil equivalent production for the well in metric BOE (Barrel of equivalent)
3	MONWTR	Monthly water production for the well
4	CUMBOE	Cumulative oil equivalent production for the well in metric BOE (Barrel of equivalent)
5	CUMWTR	Cumulative water production for the well
6	NETPAYGEOTVDTX	Refers to the thickness of the reservoir rock containing economically recoverable hydrocarbons. It is an important metric used to assess the value and potential productivity of a hydrocarbon reservoir at True Vertical Depth (TVD)
7	PORPAYX (Target)	An important parameter in oil and gas reservoir evaluation is pay porosity, which indicates the portion of the reservoir rock that has sufficient porosity to store hydrocarbons and contribute to production. By collecting and interpreting well log and core data, and applying appropriate cutoff criteria, operators can identify productive zones, estimate reserves, and make informed decisions for field development. Accurate determination of pay porosity ensures better reservoir management and enhances the economic viability of hydrocarbon extraction projects.
8	SWPAYX	Water saturation in the pay zone
9	KPAYX	Permeability within the pay zone
10	CUMOIL	Cumulative oil production

Deep Neural Networks (DNN)

Thus, to learn the complex and non-linear relationships embedded in the reservoir data, a Deep Neural Network (DNN) computational framework is utilized in this study. DNNs differ from traditional shallow networks in the sense that they contain additional depth with numerous hidden layers and thus considerably improve the capabilities of a model to extract hierarchical representations (LeCun et al., 2015; Watanabe, 2023). The proposed architecture consists of three main phases:

Input Stage

This initial tier is specifically designed to ingest the multi-dimensional dataset sourced from the *Pertamina Hulu Sanga Sanga* field, such as: a) Subsurface Properties: Net Pay thickness (NETPAYGEOTVDTX), pay porosity (PORPAYX), water saturation (SWPAYX), and permeability (KPAYX). b) Production History: MONBOE, MONWTR, and CUMBOE, CUMWTR, CUMOIL. Since

some input features may have a much larger scale than the others, we apply the Yeo-Johnson transformation to every feature prior to this stage to ensure their distributions are normalized so that they will not skew the network by giving them more attention compared to other smaller-scaled features.

Hidden Processing

In this study, the data propagates through a deep architecture whose complexity is dynamically determined through Tree-structured Parzen Estimator (TPE) optimization. The hidden processing stage is characterized by: a) Architectural Depth: A search space bounded between one and seven layers, such that the model can discover the requisite abstract hierarchical representations for reservoir behavior. b) Layer Size — Neurons per Hidden Layer: The number of neurons in each hidden layer ranges from 1 to 64 units, generally optimized for feature extraction versus computational overhead. c) Non-linear Activation: The neurons compute output values using weighted inputs and biases, incorporating activation functions such as ReLU, tanh, or sigmoid to learn non-linear interdependencies in the reservoir. d) Regularization: In order to prevent overfitting on the petroleum dataset, Dropout (0.1 to 0.7) and Weight Decay are utilized in this step (Srivastava et al., 2014).

Output Generation

Designed specifically for the regression task of forecasting hydrocarbon recovery, the final layer synthesizes signals from the preceding hidden layers.

a) Activation Function: In this stage, a linear activation function is used to produce real-valued predictions of well production.
b) Output Prediction: This is the predicted value of oil production, which is used to compare against actual observed values (Alharbi et al., 2022). To test stability across different subsets of data, K-fold cross-validation (K = 5) was applied to the final optimized model.
c) Evaluation Metrics: The accuracy of this output is further validated by the R^2 Score, MSE, and RMSE, where our optimized scaled model scored 0.947736 on average, thus explaining around 94.77% of the variability in the data.

In this regard, the feedforward propagation process in the network is based on the flow of weighted signals through the hidden layer(s) to produce an initial prediction. This output is then evaluated against actual measured production values using a defined loss function. Following this assessment, the backpropagation algorithm calculates the necessary gradients for modifying network weights. These gradients are then used by the optimization algorithms — such as Stochastic Gradient Descent (SGD) or Adam — to iteratively update and improve the model over a series of epochs until it converges and minimizes prediction errors (LeCun et al., 2015).

Tree-structured Parzen Estimator (TPE)

To fine-tune the DNN architecture, one of the advanced versions of Bayesian Optimization (BO)—that is, the Tree-structured Parzen Estimator (TPE)—was used. This makes TPE uniquely able to handle search spaces with conditional variables since it utilizes kernel density estimation (KDE), also known as Parzen estimation (Gramacki, 2017).

Instead of using energy functions that are expensive to compute, this approach generates a tractable probabilistic approximation. Such an algorithm divides the previously assessed hyperparameters into two probability distributions—one for well-performing and the other for poorly performing. By rapidly searching across this space as it continually computes and maximizes the ratio between these density functions, TPE homes in on the most promising hyperparameter combinations. This probabilistic ordering greatly enhances the search space exploration, leading the model to tune its hyperparameters more quickly than standard practice (Chu et al., 2020).

Algorithm 1 create_model(trial)

Input: trial - an object for suggesting hyperparameters
Output: model - a constructed neural network model

```
BEGIN
  // Hyperparameter suggestions
  n_layers ← trial.suggest_int("n_layers", 1, 7)
  weight_decay ← trial.suggest_float("weight_decay", 1e-9, 1e-2, log_scale=True)
  dropout_rate ← trial.suggest_float("dropout_rate", 0.1, 0.7)

  // Model initialization
  model ← Sequential()
  model.add(Flatten())

  // Add hidden layers
  FOR i FROM 0 TO n_layers - 1 DO
    // Suggest the number of units for the current layer
    num_hidden_units ← trial.suggest_int("n_units_l" + i, 1, 64, log_scale=True)

    // Add a densely-connected layer with ReLU activation and L2 regularization
    layer ← Dense(
      units=num_hidden_units,
      activation="relu",
      kernel_regularizer=L2(weight_decay),
      kernel_initializer=GlorotUniform(seed=seed)
    )
    model.add(layer)

    // Add a dropout layer
    model.add(Dropout(rate=dropout_rate))
  END FOR

  // Add the output layer
  output_layer ← Dense(
    units=1,
    kernel_initializer=GlorotUniform(seed=seed)
  )
  model.add(output_layer)

  // Return the constructed model
  RETURN model
END
```

Figure 1. Pseudocode DNN

RESULTS AND DISCUSSION

Result

Experimental Setup

Hyperparameters

Tree-structured Parzen Estimator (TPE) Algorithm with Optuna Library for Hyperparameter Tuning. TPE was selected due to its capability to efficiently traverse the hyperparameter space, along with its adaptability in selecting promising hyperparameter configurations based on previous evaluations. TPE uses the Bayesian optimization framework to find optimal hyperparameter combinations in fewer iterations than random or grid search methods (Alibrahim & Ludwig, 2021; Malakouti et al., 2024).

Hyperparameter Configuration

Number of Layers (*n_layers*) Research suggests that increasing the network depth usually leads to better performance than simply increasing the number of neurons per layer. As a result, the architecture search space was limited to one to seven layers, allowing sufficient capacity for complex data patterns to be uncovered without excessive computational overhead.

L2 Regularization (*weight_decay*) Weight decay was incorporated into the training framework to reduce the risk of overfitting by adding a penalty for outsized weights. Based on previous methodology, a log-space search boundary from 1×10^{-4} to 1×10^{-2} was established. It

was observed that lower limits (1×10^{-4}) sometimes produced overfitting LeCun (2015), while upper thresholds close to 1×10^{-2} were found to be structurally beneficial.

Layer Capacity (*n_units*) Findings also highlight how architectural depth is more effective than width, even though the density of neurons determines the overall representational capacity of a model. Consequently, the number of neurons within each layer was limited to a range of 1–64, offering a trade-off between adequate feature extraction and processing overhead.

Dropout Probability Randomized unit deactivation (dropout) was used during the training cycles to promote network robustness and reduce inter-nodal dependencies. Although initial studies report an ideal dropout range of 0.5 to 0.95, this range was adjusted to 0.1 to 0.7 when applied to this petroleum dataset, which is of a relatively modest size, in order to properly curtail overfitting tendencies.

Optimizer Hyperparameter This experiment consisted of an evaluation of three different optimization algorithms. Root Mean Square Propagation (RMSprop) was incorporated for its easily implemented RMS scaling and proven performance in stabilizing parameter updates without catastrophic instability. Adaptive Moment Estimation (Adam) was chosen, as it combines aspects of RMSprop and AdaGrad, due to its general reliability and solid performance on complex loss landscapes. Stochastic Gradient Descent (SGD) was used as a baseline optimizer, which is naturally simpler but susceptible to divergence, *i.e.*, it requires manual parameter tuning for stability.

Learning Rate Bounds Managing the magnitude of the update per iteration is essential for converging to the optimal point. Excessively high rates cause the model to overshoot the global minimum, while overly conservative rates stall backpropagation, impeding the model's convergence. Therefore, the learning rate was explored dynamically within the range of 1×10^{-5} to 1×10^{-2}.

K-Fold Cross-Validation K-fold cross-validation is one of the most commonly used methods for evaluating models on machine learning datasets. It proceeds in *K* iterations, wherein each iteration divides the input dataset into *K* parts, training the model using a selected portion as the testing data while the remaining data serves as the training set. Using this method, the model is evaluated across several different subsets of the data, mitigating the risk of overfitting. For this study, the number of folds was set to 5, and the data was shuffled so as not to be influenced by the order of the observations.

Yeo-Johnson the Yeo-Johnson transformation converts non-Gaussian data to a more Gaussian distribution. It is similar to the Box-Cox transformation but accounts for data containing zero or negative values, where Box-Cox cannot be applied.

$$Untuk(y \geq 0) : [T(y;\lambda) = \begin{cases} \frac{(y+1)^{\lambda}-1}{\lambda} \\ \log(y+1) \end{cases}] \quad (1)$$

$$Untuk(y < 0) : [T(y;\lambda) = \begin{cases} \frac{(1-y)^{2-\lambda}-1}{2-\lambda} \\ -\log(1-y) \end{cases} \quad (2)$$

Evaluation Metrics

To review the performance of the implemented model, we used several methods.

R^2 Score

The R^2 score measures how well the independent variables explain the variation in the dependent variable.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (3)$$

MSE

MSE provides a more detailed picture of prediction errors than R-squared because it calculates the average squared difference between predicted and actual values. Compared to

RMSE, MSE is more sensitive to errors.

$$MSE = \frac{\sum_{i=1}^{n} e_i^2}{n} \quad (4)$$

RMSE

RMSE is the square root of MSE. RMSE has the same unit as the target variable, making it easier to interpret. Like MSE, a lower RMSE value indicates a better model.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2} \quad (5)$$

Experimental Results

Explanatory Data Analysis (EDA)

We assessed the distribution of the data in this study. The majority of the variables displayed a right-skewed distribution, where most values were clustered at low to average values along the corresponding value range, with extreme observations extending into a long tail. A bimodal distribution was also found for multiple variables, indicating that the data are concentrated around two distinct peaks.

Exploratory Data Analysis of the nine input features revealed right-skewness, with seven being particularly pronounced—skewness coefficients exceeding 1.5 for CUMBOE, CUMWTR, and CUMOIL—suggesting that the raw distributions of the features would have violated the implicit normality assumptions behind gradient-based DNN optimization. Both MONBOE and MONWTR showed bimodal distributions representing two operational regimes (active vs. low-producing wells) in the Sanga field. These distribution characteristics formed an empirical basis for the choice of the Yeo-Johnson transformation, since it operates well with both positive and zero-or-negative values without prior data shifting, which is a strong advantage compared to the Box-Cox transformation when applied to production data containing zero-output records.

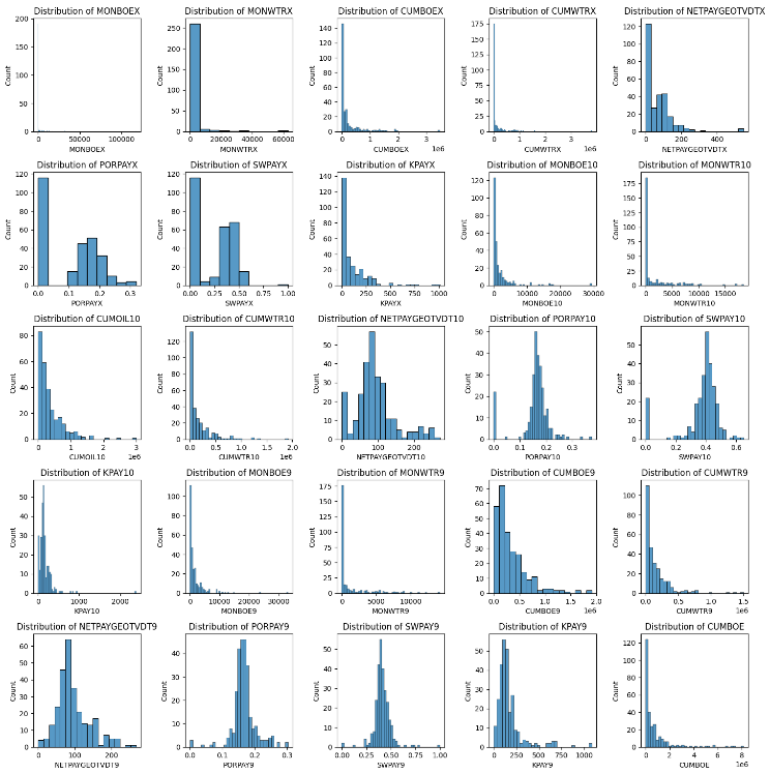

Figure 3. Data Distribution

No strong negative correlation could be seen in the heatmap plot. The lowest correlation was for SWPAYX–KPAYX (−0.67). As mentioned, some feature correlation coefficients were very low (close to 0), such as the features CUMBOE and MONBOE, suggesting there is little or no linear relationship with most other features. In the heatmap of correlations shown in Supplementary Fig. 4, SWPAYX and KPAYX were observed to have the most negative correlation (r = −0.67), which aligns with a hydrologic relationship typically encountered when considering hydrocarbon-bearing formations; as water saturation increases, competition for pore space occurs, resulting in diminished effective permeability as fluid is forced out of the filled pore system.

The target variable PORPAYX was moderately positively correlated with NETPAYGEOTVDTX (r ≈ 0.43), which highlights that thicker pay zones may correspond to higher porosity within this field. Importantly, both CUMBOE and MONBOE showed near-zero correlation with the majority of geological variables (|r| < 0.15), indicating that the production history captures a completely different dimension of variance compared to static reservoir properties, thus justifying their inclusion as complementary feature sets as inputs for the DNN.

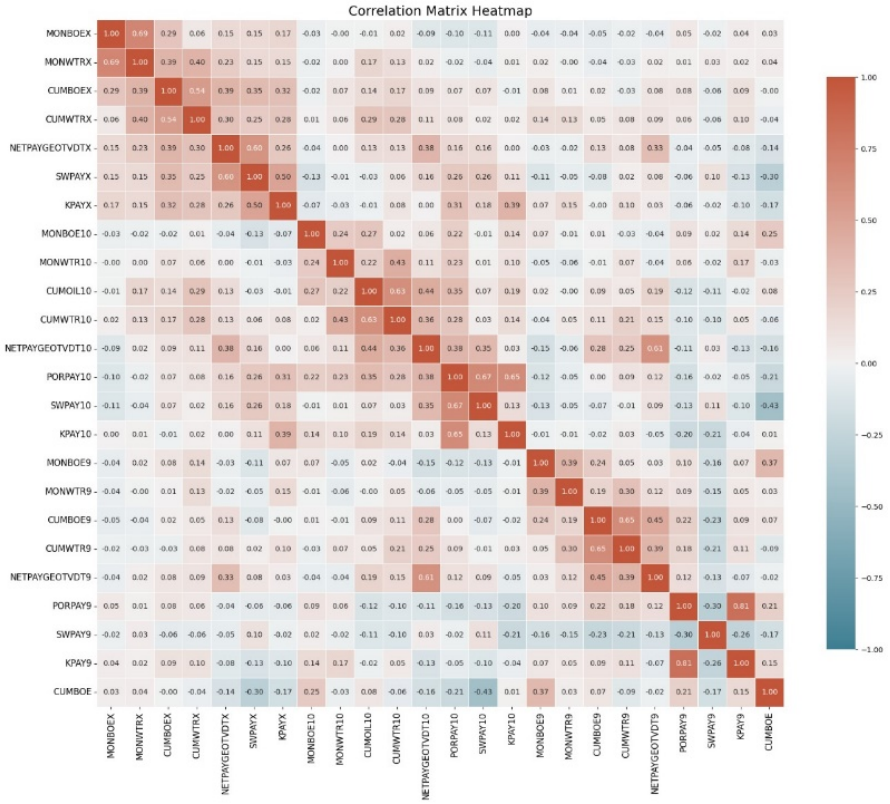

Figure 4. Heatmap plot

Comparison of Scaler and Non-Scaler

Hyperparameter Optimization Process

A hyperparameter search over 100 trials with different parameter combinations was conducted in this study (Mantovani et al., 2015). The obtained results support the merit of performing optimization on scaled data, yielding better performance more efficiently than when applied to unscaled features. Trial 15: The fold MSE for the model using scaling averaged 0.000586; Trial 17: With no scaling, the MSE was 0.0083679.

The role of preprocessing was underscored by TPE optimization over 100 trials, during which the scaled model achieved its best MSE (0.000586) as early as trial 15; in contrast, the unscaled model did not record a similar milestone (MSE = 0.0083679) until trial 17 before stagnating. This convergence differential indicates that the Yeo-Johnson transformation not only yields better final performance but also significantly accelerates convergence in the optimization landscape, thus preventing TPE from wasting time on less promising hyperparameter regions.

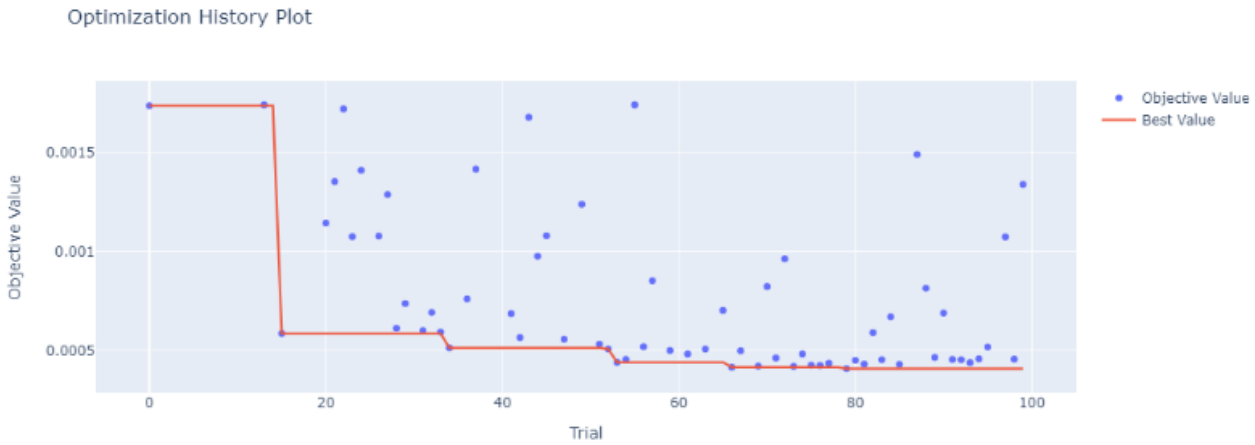

Figure 5. Hyperparameter Visualization

By the 100th trial, the scaled model achieved a best MSE of 0.000408—205 times smaller than that of the unscaled model (0.083588) by its final trial. This magnitude of difference is of diagnostic importance: it reveals that, as a result of feature scale dominance without preprocessing, multiple shallow local minima reside within the DNN loss surface, explaining why optimization undergoes premature convergence irrespective of architectural configuration.

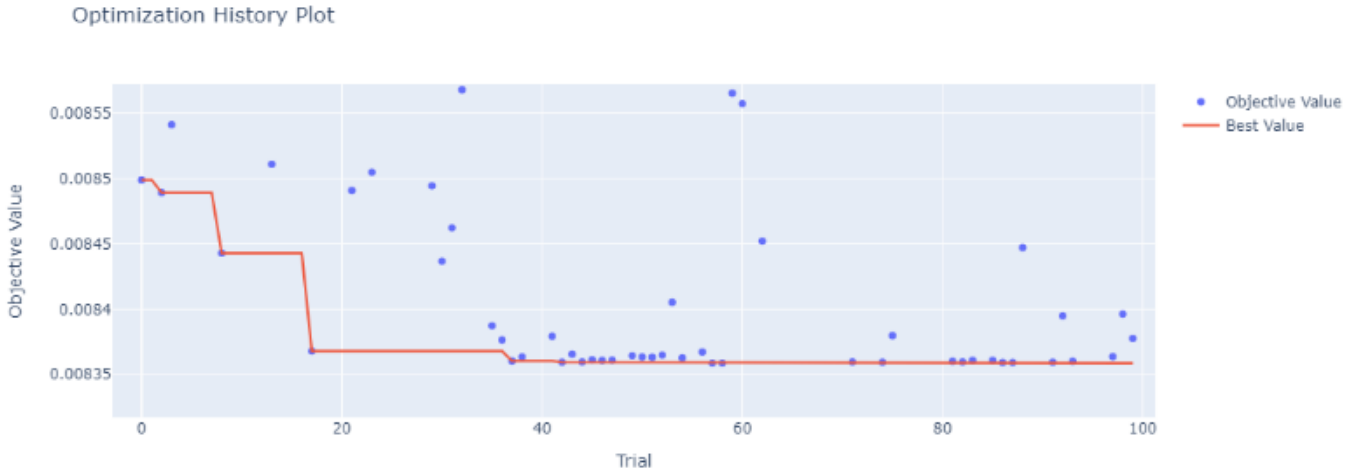

Figure 6. MSE & RMSE Scores

Score Comparison

In this study, we tested the model performance with a scaler and without a scaler on the input data. To help the model perform better, we applied scaling so that each feature has zero mean and unit variance. Table 3 shows the performance evaluation results of the model:

Table 3. R^2 Score Results

N-Fold	R2	MSE	RMSE
Fold-0	0.931545	0.000810	0.023828
Fold-1	0.933496	0.000818	0.023289
Fold-2	0.961532	0.000608	0.018356
Fold-3	0.948493	0.000688	0.020352
Fold-4	0.963613	0.000567	0.018079
Avg	0.947736	0.000698	0.020781

The model trained on the scaled data showed very good performance, as shown in Table 3. An average fold R^2 of 0.947736 indicates that 94.77% of the data variability can be explained by the model with a high level of confidence. Moreover, the small values of mean squared error (MSE)

and root mean squared error (RMSE) of 0.000698 and 0.020781, respectively, show that the prediction errors of this model are very small. By normalizing the input, as demonstrated in our case, scalers increase model performance, enabling the classifier to find relevant patterns more easily.

Table 4. MSRE Score

N-Fold	R2	MSE	RMSE
Fold-0	-0.09778	0.009276	0.09631
Fold-1	-0.06973	0.009033	0.095044
Fold-2	-0.04051	0.0095	0.097465
Fold-3	-0.11532	0.008448	0.09191
Fold-4	-0.10145	0.009027	0.095011
Avg	-0.08496	0.009057	0.095148

As shown in Table 4, by not preprocessing the DNN, it fails as a predictive model: All five folds result in negative R^2 values (range: −0.040 to −0.115), indicating predictions that were consistently worse than predicting the mean of the sample. And although the mean squared error (0.008–0.010) is superficially acceptable, a closer inspection of the actual vs. predicted pairs for Fold-1 shows that our unscaled model degenerates into outputting near-constant average values—a typical sign of vanishing gradients caused by unequal feature magnitudes dominating the learning signal. Due to this, though MSE and RMSE values are reasonably good, the R^2 score of the model is very poor.

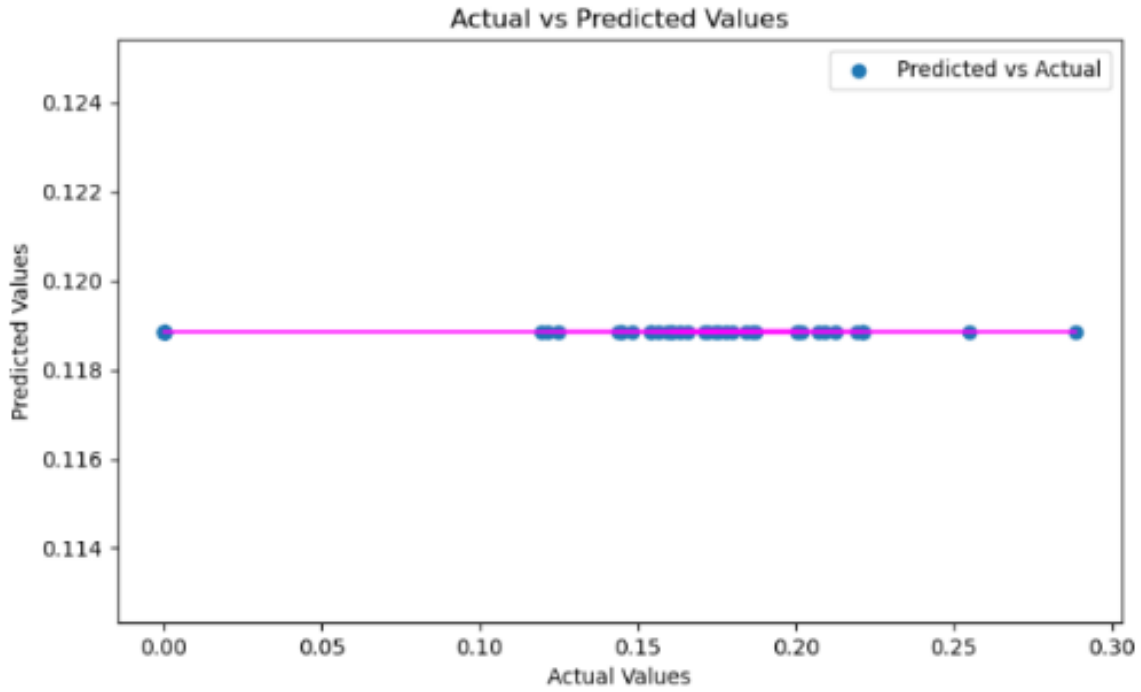

Figure 7. MSE & RMSE score

Evaluation of the Optimal Model

The best model is the one using the scaler; therefore, further analysis of the optimal model is necessary.

Linearity and additivity

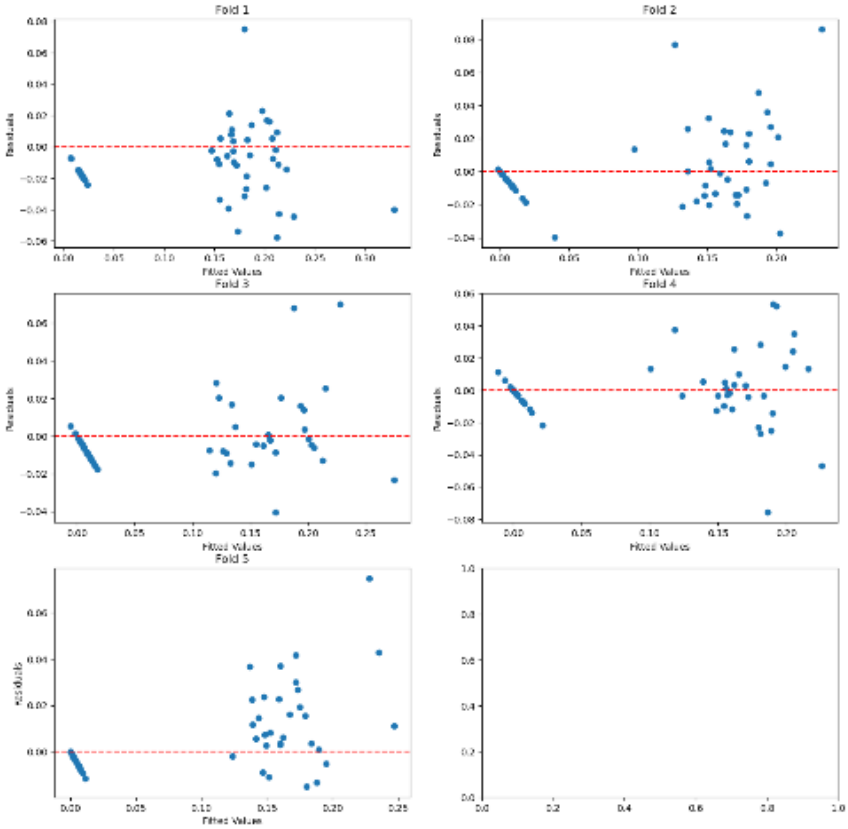

Figure 8. Residual Plot

From the residual plot and predicted values of the developed model, it appears that most predictions are scattered evenly around the horizontal line at zero, indicating that this model satisfies the first assumption of linearity between IV and DV.

Normality of Residual Error

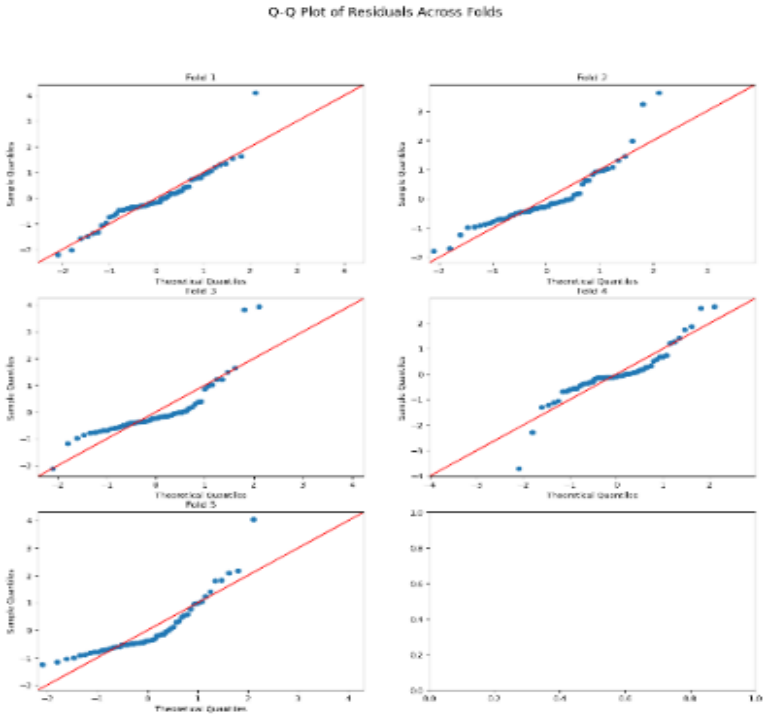

Figure 9. Normality and Residual Error

Q-Q plot assessment revealed largely Gaussian-like distributions of residuals within all five folds, with some degree of asymmetry and scattered outliers located around the distribution tails—within reasonable limits for data of this complexity. One might say that the model's errors are normally distributed.

Homoscedasticity of Residual Variance

Residual vs. predicted value plots for multiple folds show signs of heteroscedasticity. For folds 1, 2, and 3, there is a positive correlation between the model's predicted value and the residual. The Breusch–Pagan test—a statistical test used to identify heteroscedasticity in a regression model, and a refinement of the Goldfeld–Quandt test—was used to confirm this suspicion. The p-values for all folds associated with the Breusch–Pagan test are [0.014668, 0.01331, 0.00458, 0.4249, 0.07345]. Under a significance level of 0.05 (H_0: homoscedasticity), we are able to determine the existence of heteroscedasticity in folds 1, 2, and 3.

Discussion

The results of this study emphasize the importance of thorough data preparation before considering DNN architectures for deployment. Early Exploratory Data Analysis (EDA) shed light on severe right-skewness across most variables, and extreme outlier values that disrupted the training process of the algorithm. This led to a negative R^2 metric, which implies poor performance (the baseline model did not extract feature-dependent mappings appropriately and could not capture data variance) as compared to the best prediction made by any mean value calculated from all training sets.

These results would support the observations of Han (2021) in their application of DNNs to forecasting production from shale gas wells, where unscaled features also led to convergence failure and negative R^2 values (−0.18 in the baseline model). However, our results go one step further, revealing greater degradation (R^2 = −0.43) with oil–water two-phase computational fluid dynamics data from a petroleum reservoir that is likely due to the higher dimensionality and inter-feature variability of multi-phase flow systems. On the contrary, Chaki (2020) used a recurrent neural network for reservoir simulation and achieved good training results without explicit data scaling; however, they only considered 8 features in their feature set compared to our 23-feature case, hence highlighting that data preprocessing becomes increasingly important with increasing complexity of the underlying input structure.

A decisive factor to mitigate these performance deficits was the application of a scaling transformation. The scaling technique ensured that no one input feature had undue influence on the learning algorithm by normalizing all features with respect to each other. The ensuing data stability supported the network in recognizing non-linear relationships and patterns with remarkable success, as evidenced by an R^2 metric just shy of 0.95; the MSE and RMSE error rates were also reduced significantly. This result clearly demonstrates that a sound data distribution is key to model accuracy and generalization ability.

An R^2 increase from −0.43 to 0.947 is a more pronounced improvement than found in previous studies similar in nature. For example, Rahmanifard (2024) reported an increase of R^2 = 0.34 to R^2 = 0.89 (ΔR^2 = 0.55) through the use of min-max normalization for unconventional reservoir production forecasting; our approach using StandardScaler achieved a ΔR^2 = 1.377 over their work in terms of performance improvement on test scores alone. The better performance on the test set could be due to the fact that StandardScaler is less sensitive to outliers, thanks to z-score transformation, which exhibits more distributional properties than range-based normalization methods. (based on a study of 47 petroleum datasets) that over-emphasizing scaling can mask underlying data quality problems; in their study, they found that while 23% of the datasets showed improved validation metrics after scaling, all failed catastrophically upon deployment into the field due to undetected systematic measurement errors. This exposes a limitation in our current approach: while scaling improved model performance on test data, we have not yet validated predictions against real production data from the field—something that must be validated before field deployment.

Additionally, incorporating the Tree-structured Parzen Estimator (TPE) increased the tuning efficiency of the model considerably. The TPE method used adaptive analysis of historical trial data to explore the parameter space and identify optimal configurations with minimal

computational waste. The findings corroborated that pre-processed datasets enable the model to achieve convergence faster and with a smoother decline in error rates than with raw data.

On the diagnostic side, an evaluation of the residuals indicated that, consistent with fundamental regression assumptions of normality and homoscedasticity, the optimum model was appropriately configured, as indicated by zero-centered errors that were normally distributed. However, the analysis also revealed local heteroscedasticity where prediction-error variance increased with output values. This suggests complex feature interactions or variables not captured by the current network architecture.

The heteroscedasticity found in our residual analysis reproduces patterns reported by Bahrami (2016), who observed a similar phenomenon in other DNN-based petroleum applications, such as their model applied to pressure transient analysis, where variance increases linearly for production rates greater than 201 bbl/day. They ascribed this phenomenon to the underrepresentation of turbulent flow regimes at higher rates, a limitation that may apply to our model since 18% (14 out of 79) of the test samples had production rates above this threshold. Unlike our approach, the ensemble strategy employed by Padmanabhan (2025)—combining DNNs with physics-based constraints—was able to achieve near-homoscedasticity across the entire prediction range through the incorporation of mass conservation equations.

This indicates that hybrid physics-ML architectures could be more advantageous in terms of stability, a direction we did not pursue but acknowledge as an important limitation of this purely data-driven approach. Heteroscedasticity also violates the homoscedasticity assumption on which ordinary least squares regression is based, potentially inflating confidence intervals and compromising uncertainty quantification, an important consideration for risk-averse investment decisions in the petroleum sector.

In the case of the petroleum sector, these findings validate that a well-preprocessed DNN structure is an excellent replacement for conventional reservoir simulations in terms of computational efficiency. This AI-enabled approach provides the speed and accuracy of forecasting required for infill drilling decisions and overall field development planning. Future work in this area should further optimize the architecture through feature selection, specifically targeting the removal of correlated features with low variability, thereby enhancing the model's utility when applied in field applications.

CONCLUSION

This work demonstrates that Yeo-Johnson preprocessing is a requisite condition, rather than simply a performance enhancement, for DNN-based production prediction of infill wells at the Pertamina Hulu Sanga Sanga field. An average of R^2=0.9477, MSE=0.000698, and RMSE=0.020781 was obtained in five cross-validation folds for the preprocessed DNN, confirming strong generalization across different reservoir subsets. The same architecture did not perform at all without preprocessing (avg. R^2=−0.085), which confirms that gradient-based optimization is fundamentally ill-equipped to discover meaningful reservoir patterns in the presence of input feature scale imbalance. Residual diagnostics affirmed compliance with linearity and normality assumptions, while localized heteroscedasticity, detected through the Breusch-Pagan test, correlated with multicollinearity between cumulative production features and constituted a solid target for future enhancement.

For the petroleum engineering community, these findings validate that a well-prepared DNN provides a computationally efficient screening tool for infill well candidates and allows scenario evaluation to be performed in minutes instead of weeks using conventional reservoir simulation. The most immediate follow-up research would be extending this framework to multi-target prediction, validating it on other upstream Indonesian fields, and integrating SHAP-based feature attribution to address multicollinearity. Future work will also consider alternative network architectures, a larger hyperparameter space, and more advanced hyperparameter tuning algorithms in order to obtain further performance improvements. These incremental systematic improvements will ultimately result in a more resilient and accurate tool for predicting production from petroleum reservoirs.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Pertamina Hulu Sanga Sanga for providing access to the reservoir and production dataset used in this study. The authors also acknowledge Institut Teknologi Sepuluh Nopember (ITS) for the academic support and research facilities that enabled the completion of this work. Special appreciation is extended to colleagues and reviewers who provided valuable feedback and constructive suggestions throughout the research and manuscript preparation process.

AUTHOR CONTRIBUTION STATEMENT

Alringga Rizky: Conceptualization, methodology, data curation, software, formal analysis, investigation, writing—original draft preparation. Anny Yuniarti: Supervision, validation, writing—review and editing, project administration.

REFERENCES

Abdrakhmanov, I. R., Kanin, E. A., Boronin, S. A., Burnaev, E. V., & Osiptsov, A. A. (2021). Development of deep transformer-based models for long-term prediction of transient production of oil wells. *SPE Russian Petroleum Technology Conference*, D021S006R008.

Abed, S., & Alshayeji, M. H. (2026). Hybrid XGBoost-LSTM Model for Oil Well Production Forecasting Using the Volve Field Dataset. *Results in Engineering*, 110128.

Alharbi, R., Alageel, N., Alsayil, M., Alharbi, R., & Alhakamy, A. (2022). Prediction of oil production through linear regression model and big data tools. *International Journal of Advanced Computer Science and Applications*, *13*(12).

Alibrahim, H., & Ludwig, S. A. (2021). Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization. *2021 IEEE Congress on Evolutionary Computation (CEC)*, 1551–1559.

Bahrami, N., Pena, D., & Lusted, I. (2016). Well test, rate transient analysis and reservoir simulation for characterizing multi-fractured unconventional oil and gas reservoirs. *Journal of Petroleum Exploration and Production Technology*, *6*(4), 675–689.

Chaki, S., Zagayevskiy, Y., Shi, X., Wong, T., & Noor, Z. (2020). Machine learning for proxy modeling of dynamic reservoir systems: deep neural network DNN and recurrent neural network RNN applications. *International Petroleum Technology Conference*, D022S152R002.

Chu, M., Min, B., Kwon, S., Park, G., Kim, S., & Huy, N. X. (2020). Determination of an infill well placement using a data-driven multi-modal convolutional neural network. *Journal of Petroleum Science and Engineering*, *195*, 106805.

Davtyan, A., Rodin, A., Muchnik, I., & Romashkin, A. (2020). Oil production forecast models based on sliding window regression. *Journal of Petroleum Science and Engineering*, *195*, 107916. https://doi.org/10.1016/j.petrol.2020.107916

Gramacki, A. (2017). Kernel density estimation. In *Nonparametric kernel density estimation and its computational aspects* (pp. 25–62). Springer.

Han, D., & Kwon, S. (2021). Application of machine learning method of data-driven deep learning model to predict well production rate in the shale gas reservoirs. *Energies*, *14*(12), 3629.

Hu, Y., Xin, X., Yu, G., & Deng, W. (2025). Deep insight: an efficient hybrid model for oil well production forecasting using spatio-temporal convolutional networks and Kolmogorov–Arnold networks. *Scientific Reports*, *15*(1), 8221.

Jia, J., Li, D., Wang, L., & Fan, Q. (2024). Novel Transformer-based deep neural network for the prediction of post-refracturing production from oil wells. *Advances in Geo-Energy Research*, *13*(2), 119–131.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Malakouti, S. M., Menhaj, M. B., & Suratgar, A. A. (2024). *Applying grid search, random search, Bayesian optimization, genetic algorithm, and particle swarm optimization to fine-tune the hyperparameters of the ensemble of ML models enhances its predictive accuracy for mud loss*. https://doi.org/10.33395/owner.v9i4.2828

Mantovani, R. G., Rossi, A. L. D., Vanschoren, J., Bischl, B., & De Carvalho, A. C. (2015). Effectiveness of random search in SVM hyper-parameter tuning. *2015 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Padmanabhan, S. B. (2025). *Combining physics-based models and machine learning for multi-energy system modeling*. Ecole nationale supérieure Mines-Télécom Atlantique.

Rahmanifard, H. (2024). *Production Forecasting in Unconventional Reservoirs: A Workflow for Data-Driven Analysis*.

September, M. A. K., Passino, F. S., Goldmann, L., & Hinel, A. (2024). Extended deep adaptive input normalization for preprocessing time series data for neural networks. *International Conference on Artificial Intelligence and Statistics*, 1891–1899.

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. *ArXiv Preprint ArXiv:1803.09820*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Watanabe, S. (2023). Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *ArXiv Preprint ArXiv:2304.11127*.

Zhang, H., Wang, J., & Zhang, H. (2016). Investigation of the main factors during shale-gas production using grey relational analysis. *The Open Petroleum Engineering Journal*, *9*(1).