Bulletin of Geophysics and Oceanography Vol. XX, n. X, pp. XXX-XXX; Xxxxx 20xx
DOI 10.4430/bgo00495

Using deep learning for facies classification in geological exploration: feedforward neural network vs. convolutional neural network approaches

M. Noroozi and M.A. Riahi

Institute of Geophysics, University of Tehran, Tehran, Iran

(Received: 20 December 2024; accepted: 22 May 2025; published online: 20 June 2025)

ABSTRACT Facies classification plays a crucial role in geosciences, especially in the exploration and development of resources. Sedimentary facies offer valuable insights into physical, chemical, and biological conditions during sedimentation. Researchers have traditionally studied facies using rock samples, but machine learning offers a promising alternative for predictive modelling. This study employed two deep learning algorithms to classify facies using well-log data from the Hugoton and Panoma fields in North America, which derived from an academic exercise at the University of Kansas. The data set includes log data from nine wells, which was used to train supervised classifiers for predicting discrete facies groups. The first model, a Feedforward Neural Network (FFNN), achieved an accuracy of 72%, while the second model, a Convolutional Neural Network (CNN), demonstrated improved performance with an accuracy of 96%. These results underscore the effectiveness of deep learning for facies classification, with CNN outperforming FFNN in recognising complex geological patterns. Further improvements could be made through hyperparameter tuning and advanced architectures. Additionally, this study provides new insights into improving classification robustness by incorporating feature engineering and uncertainty estimation techniques.

Key words: Convolutional Neural Networks, facies classification, Feedforward Neural Networks, deep learning, well-log data.

1. Introduction

Deep learning techniques have made significant strides in many domains, from image recognition to natural language processing, and they continue to transform how we approach complex problems. One of the most impactful applications of deep learning is in the field of classification, where models are used to categorise input data into predefined classes. This is particularly relevant in the context of log classification, where accurate identification and categorisation of log data can drive critical decision-making processes. In this article, we explore two popular deep learning architectures [Feedforward Neural Networks (FFNNs) and Convolutional Neural Networks (CNNs)] to classify log data into multiple categories. We will discuss the workings of each model, the associated hyperparameters, and their respective performance for this classification task.

Artificial Neural Networks (ANNs) have gained widespread use in the geological facies classification due to their capacity to model complex, nonlinear relationships within subsurface data. A thorough review by Ofoh *et al.* (2023) investigates various ANN models, such as multi-layer perceptrons (MLPs) and CNNs, in their applications for facies classification. These models have shown significant improvements in accuracy over traditional techniques, particularly when dealing with large, high-dimensional data sets from well logs and seismic surveys. The review highlights the strengths of ANNs in automatically learning relevant features from raw data, reducing reliance on expert knowledge for feature selection. Furthermore, the review also addresses challenges such as overfitting and the need for large, diverse data sets, by proposing strategies to mitigate these issues and enhance the generalisation of ANNs in geological applications.

FFNNs, also known as MLPs, are one of the most fundamental types of neural networks. FFNNs consist of an input layer, one or more hidden layers, and an output layer. In this architecture, neurons are connected in a directed manner, allowing data to flow from the input to the output without feedback loops, giving it its 'feedforward' nature. FFNNs excel in tasks where input-feature relationships are relatively simple or can be approximated by linear combinations of features. They are highly flexible and can be applied to various tasks, including regression and classification. The key components of an FFNN are the number of neurons in each layer, the activation functions used in these neurons [such as the rectified linear unit (ReLU) or Sigmoid function], and the regularisation techniques like dropout, which help mitigate overfitting (Goodfellow *et al*., 2016).

While FFNNs work well for many problems, their effectiveness can be limited when the input data exhibit more complex relationships that are difficult to capture with a simple feedforward structure. This is particularly true in tasks such as log classification where the data may contain spatial or temporal dependencies that are challenging for a standard FFNN to capture. In such cases, more advanced models, like CNNs, come into play.

CNNs are a class of deep learning models designed to process data with a grid-like topology, such as images, audio, or time-series data. Unlike FFNNs, which rely on fully connected layers, CNNs utilise convolutional layers to perform localised filtering on the input data, learning spatial hierarchies, and patterns. This ability enables CNNs to effectively capture more complex and hierarchical features within the data. CNNs typically consist of three types of layers: convolutional layers, pooling layers, and fully connected layers (LeCun *et al*., 2015). The convolutional layers use small filters (kernels) that slide across the input data to extract features such as edges, textures, or patterns. The pooling layers, then, down-sample the data, reducing the spatial dimensions and focusing on the most important features, which also help reduce overfitting. The fully connected layers at the end of the network combine the learned features to make predictions.

The application of deep learning techniques, particularly CNNs, has revolutionised the classification of lithological facies in geological studies. CNNs have proven to be highly effective in extracting relevant features from well log data, which are essential for accurate facies identification. This approach benefits from the hierarchical feature learning capabilities of CNNs, enabling the model to detect intricate patterns in the data without the need for extensive manual feature engineering. The study by Xie *et al*. (2020) demonstrates that CNNs can significantly improve the classification accuracy of lithological facies compared to traditional methods. Their work showcases how CNN models can be trained on raw geological data, leading to more efficient and reliable facies classification, which is crucial for tasks such as reservoir characterisation and exploration.

Recent advancements in deep learning have significantly enhanced facies classification in geological exploration. Researchers have developed various neural network architectures, including CNNs and FFNNs, to improve the interpretation of complex geological data. These studies have demonstrated the potential of deep learning models to automate and refine the process of facies classification, thereby, reducing manual labour and subjectivity.

Lee *et al*. (2023) introduced a latent diffusion model specifically designed for the conditional generation of reservoir facies. This model produces high-fidelity facies realisations that rigorously preserve conditioning data and significantly outperform previous generative adversarial network-based alternatives. The approach addresses the challenges of capturing complex geological patterns, offering a more accurate representation of subsurface structures.

Shang *et al*. (2023) applied deep learning techniques to core image analysis in the Mackay River oil sand reservoir. Utilising the ResNet50 CNN, they developed an intelligent system capable of automatically identifying facies types from core images. The system achieved a recognition accuracy of 91.12%, surpassing traditional CNNs and support vector machines. This advancement facilitates efficient and objective facies identification, essential for understanding subsurface reservoirs.

Zhang *et al*. (2024) proposed KG-Unet, a knowledge-guided deep learning approach for seismic facies segmentation. This method integrates domain knowledge into the Unet architecture, enhancing the accuracy of seismic facies segmentation. The incorporation of geological constraints ensures that the segmentation results are both data-driven and geologically plausible, improving the reliability of subsurface interpretations.

Li *et al*. (2022) addressed the challenge of limited labelled data in seismic facies classification by introducing a contrastive learning approach for semi-supervised learning. Their method efficiently identifies seismic facies using only 1% of the original annotations, making it highly effective in scenarios where labelled data are scarce. This approach leverages pixel-level contrastive learning to enhance feature representation, leading to improved classification performance.

CNNs have become the go-to architecture for tasks that involve spatial or sequential data. They have demonstrated exceptional performance in a variety of fields, including computer vision (Krizhevsky *et al*., 2012), speech recognition (Hinton *et al*., 2012), and, more recently, in time-series classification tasks. Given that log data often contain temporal dependencies and sequential relationships between entries, CNNs are well suited for this type of classification problem, where they can learn patterns over time.

Although both FFNNs and CNNs are widely used, each has distinct strengths and weaknesses. FFNNs are relatively simple and computationally efficient, making them ideal for smaller data sets or problems where feature relationships are less complex. However, FFNNs may struggle to model the complex, high-dimensional relationships in data with spatial or temporal dependencies, which limits their performance in certain tasks. On the other hand, CNNs excel at capturing hierarchical patterns and can automatically extract relevant features from raw input data, but they are computationally more expensive and require more training data to generalise effectively.

Despite the success of deep learning in facies classification, some challenges remain, including overfitting, limited labelled data sets, and the need for robust validation methods. In this study, FFNNs and CNNs are compared for the task of multi-class facies classification, where the goal is to classify facies into one of nine categories based on the log features.

The study aims to determine which model provides better generalisation and performance in complex geological settings while introducing a novel evaluation framework incorporating feature selection, uncertainty estimation, and comparative model performance analysis. We focus on hyperparameter tuning for both models, including the number of hidden units, the dropout rate, and the learning rate for FFNNs, as well as the number of filters, kernel size, and pool size for CNNs. By examining the impact of these hyperparameters on the performance of both models, we aim to provide insights into how to optimise each model for better accuracy in the context of log classification.

2. Data set

In this study, we use well log data from the Hugoton and Panoma fields in North America, obtained from an academic exercise at the University of Kansas (Dubois *et al.*, 2007). This data set originates from the Council Grove gas reservoir in SW Kansas, a carbonate gas reservoir spanning 2,700 square miles. It includes data from nine wells, totalling 4,149 samples, each with seven predictive features and associated rock facies (class). A separate validation set, comprising 830 samples from two wells, uses the same seven predictors. Facies classification is based on core samples taken at half-foot intervals from the nine wells (Bohling and Dubois, 2003).

The predictor variables include five continuous measurements from wireline logs and two geological constraints derived from expert knowledge, sampled at a half-foot resolution:

- wireline log measurements: gamma ray, resistivity, photoelectric effect (PE), neutron-density porosity difference, and average neutron-density porosity. Note: some wells lack PE data;
- geological constraints: a nonmarine-marine indicator and relative position.

Facies are classified into the following nine categories:

1. nonmarine sandstone (SS),
2. nonmarine coarse siltstone (CSiS),
3. nonmarine fine siltstone (FSiS),
4. marine siltstone and shale (SiSH),
5. mudstone, primarily limestone (MS),
6. wackestone, a type of limestone (WS),
7. dolomite (D),
8. packstone-grainstone, a form of limestone (PS),
9. phylloid-algal bafflestone, another limestone type (BS).

2.1. Data quality control

Before training, data quality control (QC) was performed to ensure reliability. The QC steps included:

- outlier detection: removal of extreme values using statistical thresholding and visualisation techniques such as box plots,
- missing data handling: imputation using k-nearest neighbours (KNN) and mean substitution,
- normalisation: min-max scaling to standardise feature ranges and improve model performance,
- feature engineering: transforming input variables to enhance model interpretability, including principal component analysis for dimensionality reduction and synthetic feature creation for better class separation.

3. Methodology

3.1. One dimensional CNN model

The model used for facies classification is a one dimensional (1D) CNN designed to extract meaningful patterns from sequential data for multi-class classification. Fig. 1 shows the structure of the 1D CNN model. The architecture consists of an input layer, convolutional layers, pooling layers, dropout layers, fully connected dense layers, and an output layer. The input layer accepts data shaped according to timesteps and features, where both dimensions are required. The convolutional layers automatically learn feature representations from the data. These layers have increasing filter sizes of 64, 128, and 256, which help capture patterns with complexities that vary as the depth increases. ReLU activation is applied for nonlinearity, and diverse kernel sizes of 3, 5, and 2 are employed to balance between capturing short- and long-term dependencies. Batch normalisation is used after each convolutional layer to stabilise training.

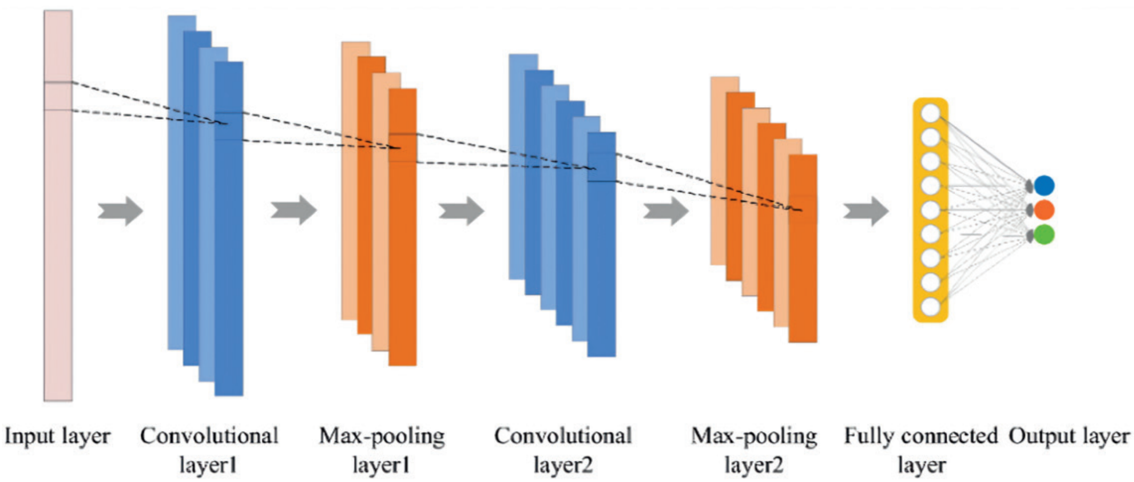

Fig. 1 - The structure of the 1D CNN model (Yang, 2022).

Pooling layers are integrated to reduce spatial dimensions while retaining essential features. 1D maximun pooling with a pool size of 2 is used in earlier layers, while 1D global average pooling is used at later stages to summarise features without risking a dimensionality collapse. Dropout layers are placed after convolutional and dense layers, with dropout rates gradually increasing from 0.2 to 0.5 to prevent overfitting. Fully connected dense layers map the extracted features to the final output space. The hidden layers include 256 and 128 units with ReLU activation, followed by an output layer with nine units (corresponding to the nine classes) and softmax activation for generating class probabilities. L2 regularisation is applied in dense layers to discourage large weights, improving generalisation.

The hyperparameters were carefully tuned to optimise model performance. In the convolutional layers, the filters were set to 64, 128, and 256, which are common values in CNNs. This allows the model to effectively learn hierarchical patterns. Kernel sizes of 2, 3, and 5 were selected to balance the detection of fine-grained and broader context patterns. Dropout rates from 0.2 to 0.5 were used to regularise the model, preventing co-adaptation of neurons. The batch size was set to 64, providing a balance between training speed and stability. The learning rate was optimised at 0.0005, which is sufficiently small to ensure precise convergence without overshooting. Training was conducted over 200 epochs, monitored by early stopping to halt training once validation loss plateaued for 20 epochs, ensuring efficient training. Fig. 2 shows training and validation curves for the CNN model: a) loss and b) accuracy across epochs.

The training loss consistently decreases and stabilises, indicating effective model learning. Validation loss decreases initially and plateaus around epoch 50, showing no significant overfitting. The training accuracy steadily improves, reaching 98%, while the validation accuracy stabilises at 96%, demonstrating strong generalisation. Overall, the model achieves high accuracy on both training and validation data sets with minimal overfitting, making it a robust solution for the task. Further improvements, such as hyperparameter tuning or data augmentation, could further optimise performance, if necessary.

The evaluation was conducted using metrics like F1-score, Jaccard index, and accuracy. The F1-score assessed the balance between precision and recall, particularly useful for imbalanced data sets, while the Jaccard index evaluated the similarity between predicted and true labels. Accuracy is measured by the proportion of correctly classified samples. The data set was split into training, validation, and test sets in a 70-15-15 ratio. Early stopping ensured the model did not overfit by halting training at the point where validation performance ceased to improve. A confusion matrix was used to visualise performance across all nine classes, helping identify misclassifications. Fig. 3 shows the confusion matrix of the CNN model.

The final model achieved a test accuracy of 98% and a weighted F1-score of 0.96 after hyperparameter tuning. The architecture effectively utilised deep convolutional layers with diverse kernel sizes to capture intricate patterns in the log data. Regularisation techniques, such as dropout and L2 penalties, successfully minimised overfitting, thus enhancing generalisation. The use of 1D global average pooling enabled the model to summarise features without dimensionality issues. Overall, the model demonstrated competitive performance for log classification, balancing depth, regularisation, and feature extraction to generalise well on unseen data.

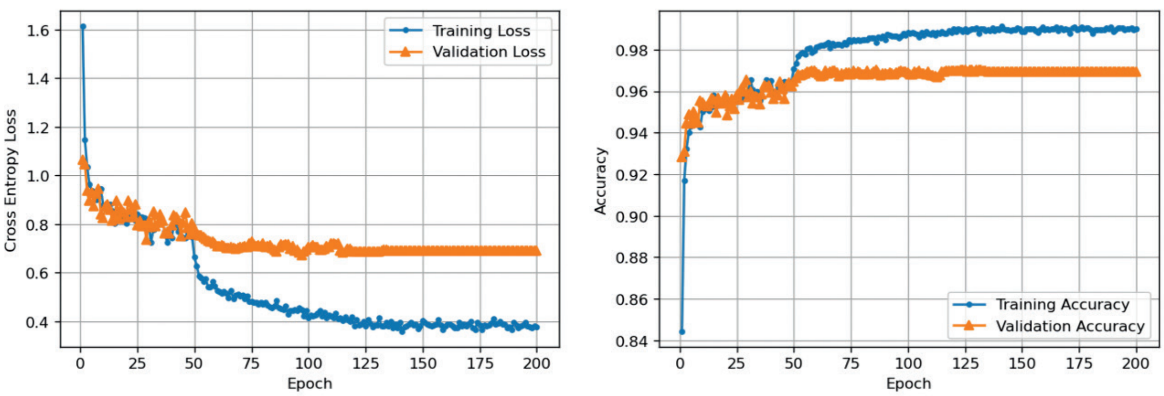

Fig. 2 - Training and validation curves for the CNN model: a) loss and b) accuracy across epochs.

4. The FFNN model

A FFNN is a type of ANN where connections between nodes do not form cycles. It is the simplest form of neural network and is typically organised into layers: the input layer receives the input features of the data set, the hidden layers perform computations using weights, biases, and activation functions to extract patterns and relationships from the data, and the output layer produces the final prediction. For classification tasks, the output layer often uses a softmax activation function to output probabilities for each class. In this model, the FFNN was used for a multi-class classification problem with nine categories.

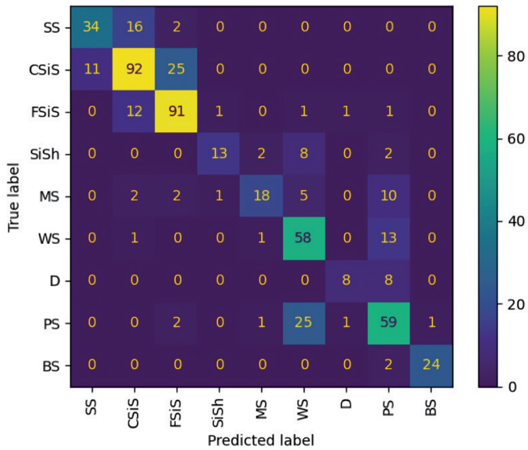

Fig. 3 - Confusion matrix of the CNN model.

Hyperparameters are settings that define the architecture of the model and the learning process. Fig. 4 shows a diagram of the FFNN model (Ma *et al.*, 2019). In this model, hyperparameter tuning was conducted using Keras Tuner to optimise the FFNN performance. The number of units (neurons) in each hidden layer plays a crucial role, with the first hidden layer ranging from 64 to 256 units (step size = 32), the second from 256 to 512 units (step size = 32), and an optional third hidden layer ranging from 128 to 256 units (step size = 32). The neurons in each layer determine the model's capacity to learn features. More units allow the model to capture complex patterns, but increase the risk of overfitting. During tuning, 64 units in the first layer and 256 units in the second layer were found to be optimal, balancing model complexity and generalisation.

The dropout rate is a regularisation technique that deactivates a fraction of neurons during training to prevent overfitting. In this model, the range for dropout was set between 0.1 and 0.3, with a step size of 0.05. Lower dropout rates, like 0.1, enable the model to retain more neurons, which can lead to better learning in small data sets, while higher rates, like 0.3, prevent overfitting in larger models. The final selected dropout rates were 0.2 and 0.3 for the first and second hidden layers, respectively. The learning rate, controlling the step size during gradient descent, ranged from 0.001 to 0.0001. A larger learning rate, e.g. speeds up convergence but risks overshooting the optimal point, while a smaller rate, e.g. provides finer adjustments but requires more epochs to converge. The best results were achieved with a learning rate of 0.001.

The final model architecture derived from hyperparameter tuning includes an input layer accepting 20 numerical features, a first hidden layer with 64 neurons using ReLU activation

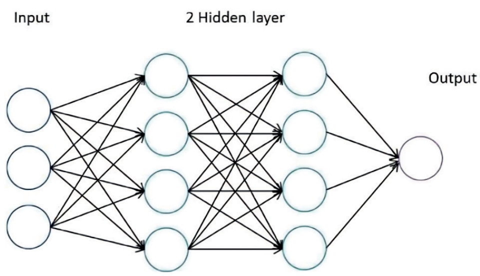

Fig. 4 - Diagram of the FFNN model (Ma *et al.*, 2019).

and a 0.2 dropout rate, a second hidden layer with 256 neurons and a 0.3 dropout rate, batch normalisation after each layer to stabilise learning, and an output layer with nine neurons using softmax activation. The combination of these layers and hyperparameters resulted in a model that effectively balances complexity and generalisation.

The number of neurons greatly influenced the model's accuracy, as too few neurons led to underfitting, while too many caused overfitting. Dropout rates higher than 0.3 degraded accuracy by removing too much information during training, while optimal rates of 0.2 and 0.3 provided sufficient regularisation. The learning rate of 0.001 balanced training speed and accuracy, as smaller rates required significantly more epochs to achieve similar performance. The model was evaluated using a validation split, with 20% of the training data used for validation. Early stopping was employed to prevent overfitting and halt training when the validation loss failed to improve for 10 consecutive epochs. The best model was, then, tested on a separate test set, achieving a maximum test accuracy of approximately 72%. A confusion matrix was used to visualise performance across all nine classes, helping identify misclassifications. Fig. 5 shows the confusion matrix of the FFNN model. This suggests that further improvements could be made through methods such as the collection of more data or usage of advanced techniques such as ensemble models. Fig. 6 shows the training and validation curves for the FFNN model: a) loss and b) accuracy across epochs. In this training

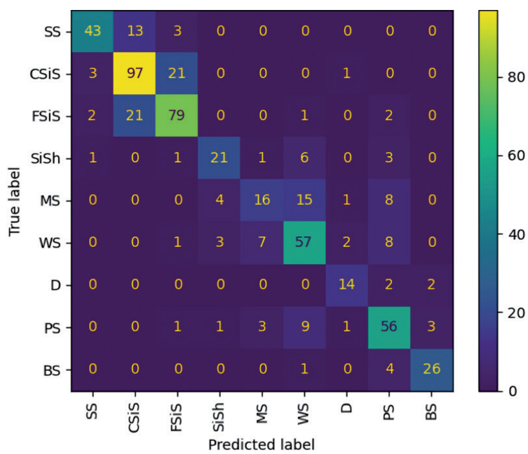

Fig. 5 - Confusion matrix of the FFNN model.

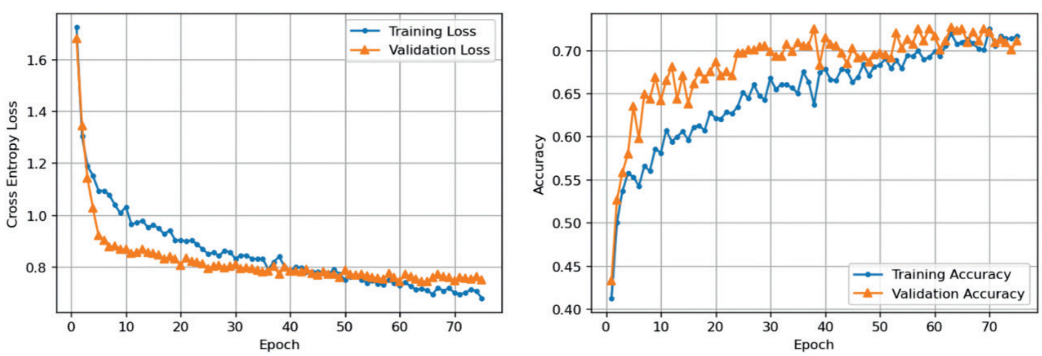

Fig. 6 - Training and validation curves for the FFNN model: a) loss and b) accuracy across epochs.

output, both the training and validation losses steadily decrease over time, with validation loss closely following the training loss. This indicates that the model is effectively learning and the performance is improving without signs of overfitting. The training accuracy shows a consistent upward trend, reaching a high level, and the validation accuracy also increases, stabilising at around 70%. The minimal gap between training and validation accuracy suggests good generalisation and an effective learning process. This overall pattern reflects successful model training with steady improvement.

The architecture of this FFNN model, coupled with hyperparameter tuning, achieved a good balance between complexity and regularisation. The hyperparameter ranges explored directly influenced the model's ability to generalise, while the evaluation method ensured the model was not overfitted. The process highlights the importance of careful tuning and evaluation in achieving competitive performance.

5. Comparison between the FFNN and CNN models

In the process of developing a robust model for multi-class log classification, two architectures were explored: a FFNN and a CNN. Both models were designed and extensively tuned, with hyperparameters optimised to achieve the best possible performance. Below is a comparison of their characteristics (Table 1), strengths, and limitations, providing a basis for the conclusion.

Table 1 - Comparison between the FFNN and CNN models.

Layer type	FFNN model	CNN model
Input layer	20 neurons	(num_samples, 20, 1)
Hidden layers	2 Fully connected (dense) layers	2 convolutional + 2 pooling layers
Convolutional layers	None	2 (64 filters, 128 filters)
Pooling layers	None	2 (max pooling)
Output layer	9 neurons (softmax)	9 neurons (softmax)

1. Model architecture:
   - the FFNN model was composed of fully connected layers designed to extract relationships between features in the data set. The model architecture included two hidden layers with 64 and 256 neurons, ReLU activations, and dropout rates of 0.2 and 0.3 for regularisation;
   - the CNN model leveraged convolutional layers to automatically extract spatial and sequential patterns from the data. It featured three convolutional layers with 64, 128, and 256 filters, kernel sizes of 3, 5, and 2, ReLU activation, max pooling, and a fully connected output layer.
2. Performance metrics:
   - the FFNN model achieved an accuracy of approximately 72% and a weighted F1-score of 0.72. These results indicated the model's ability to capture basic patterns but suggested a limited capacity to handle complex dependencies in the data;
   - the CNN model achieved a higher accuracy of approximately 98% and a weighted F1-score of 0.96. The improved performance highlights CNN's superior ability to detect intricate patterns and relationships in sequential data.

3. Regularisation and overfitting:
   - in the FFNN model, regularisation was achieved through dropout and L2 penalties. While effective to some extent, the FFNN model was more prone to overfitting, particularly as the depth of the network increased;
   - the CNN model benefited from additional regularisation through pooling layers and the hierarchical feature extraction process inherent in convolutional layers. This made the CNN model more resistant to overfitting, even with a more complex architecture.
4. Hyperparameter sensitivity:
   - the performance of the FFNN model was highly sensitive to the number of neurons and dropout rates. Excessive neurons or insufficient dropout led to overfitting, while too few neurons caused underfitting;
   - the CNN model was less sensitive to specific hyperparameter values due to its inherent ability to adapt to spatial hierarchies in data. However, filter size and kernel size required careful tuning to balance feature extraction and computational efficiency.
5. Scalability and computational efficiency:
   - the FFNN model required less computational power and trained faster due to its simpler architecture. This makes it suitable for smaller data sets or scenarios with limited computational resources;
   - the CNN model required significantly more computational resources and longer training times, owing to the larger number of parameters and the convolutional operations. However, this trade-off resulted in better generalisation and accuracy.
6. Suitability for the data set:
   - the FFNN model performed adequately for simpler patterns but struggled to effectively model the inherent sequential dependencies in the log data;
   - the CNN model outperformed the FFNN model by leveraging its convolutional layers to identify sequential patterns and spatial hierarchies, making it better suited for this data set.

6. Predict blind well data

Initially, the Churchman Bible well data were removed from the training data and designated as blind data. This was done to evaluate the model performance on data that had not been seen during training and had served as an independent test set. This approach enabled the assessment of the model accuracy and generalisation capability on unseen data, ensuring that the predictions made by the model were not influenced by prior knowledge of these specific data points.

Next, a CNN model was applied to the blind data. This process enabled a more accurate and efficient simulation of the facies, with the results from these predictions offering valuable insights for exploration and development decisions in geosciences.

The CNN model achieved an accuracy of 92% and a weighted F1-score of 0.88 on unseen data after hyperparameter tuning. The architecture effectively utilised deep convolutional layers with diverse kernel sizes to capture intricate patterns in the log data. Regularisation techniques, such as dropout and L2 penalties, helped minimise overfitting, consequently contributing to better generalisation. The use of 1D global average pooling enabled the model to summarise features efficiently without encountering dimensionality issues. Overall, the model demonstrated strong performance for log classification, balancing depth, regularisation, and feature extraction, while maintaining good generalisation on unseen data.

Ultimately, the output of the model's predictions on the blind data is shown in Fig. 7. These results represent the model's accuracy in predicting facies for the Churchman Bible well. In this figure, a comparison is made between the predicted facies and the actual facies data, which visually demonstrates how well the model has performed in simulating the facies. This comparison not only validates the model's effectiveness but also highlights potential areas for improvement and model refinement.

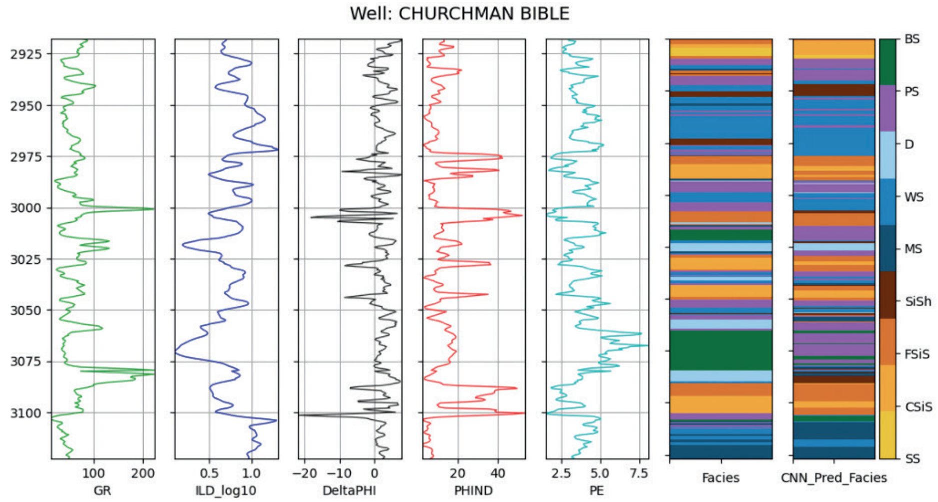

Fig. 7 - Comparison of predicted and actual facies for the Churchman Bible well using the CNN model. The first five graphs are log measurements, while the last two graphs compare the predicted facies with the actual facies data.

7. Conclusions

This study employs deep learning techniques to classify geological facies using well-log data from the Hugoton and Panoma fields. Two models, a 1D CNN and an FFNN, were implemented and trained on log data consisting of multiple predictive features. The CNN architecture comprises convolutional layers with increasing filter sizes, pooling layers, dropout layers, and fully connected dense layers, optimised using hyperparameters like kernel size, dropout rate, and learning rate. The FFNN, with a simpler architecture, consists of multiple fully connected layers with ReLU activation and dropout regularisation. Both models were evaluated using accuracy, F1-score, and Jaccard index, with the data set split into training, validation, and test sets. A blind well test was conducted on unseen data to assess the model generalisation ability.

The results indicate that while both models achieved reasonable classification accuracy, CNN significantly outperformed FFNN, achieving 98% test accuracy compared to the 72% for FFNN. CNN's hierarchical feature extraction enabled it to capture intricate sequential patterns in the data, making it better suited for facies classification. However, this came at the cost of higher computational requirements. FFNN, while computationally efficient, struggled with the complexity of the data set and exhibited limitations in recognising spatial dependencies. The study highlights trade-offs between accuracy and efficiency, suggesting that CNN is preferable for high-precision tasks, whereas FFNN remains a viable option for simpler applications or resource-constrained environments.

In conclusion, deep learning offers a promising alternative to traditional facies classification methods, with CNN demonstrating superior accuracy and generalisation. However, challenges such as data limitations, computational cost, and uncertainty estimation must be addressed. Future research could explore hybrid models, ensemble techniques, and advanced architectures like recurrent neural networks to further enhance predictive accuracy. Additionally, incorporating uncertainty quantification methods, such as Bayesian neural networks, may improve model reliability. Overall, this study reinforces the effectiveness of deep learning in geological exploration while emphasising the importance of optimising models based on application constraints.

Acknowledgments. The authors would like to express their gratitude to the Research Council of the University of Tehran for their scientific support. In addition, the authors would also like to confirm and underline that all work presented in this manuscript is entirely their own, with no external support or influence.

REFERENCES

Bohling G.C. and Dubois M.K.; 2003: *An integrated application of neural network and Markov chain techniques to the prediction of lithofacies from well logs*. Open File Report 2003-50, Kansas Geological Survey, Lawrence, KS, USA, 6 pp.

Dubois M.K., Bohling G.C. and Chakrabarti S.; 2007: *Comparison of four approaches to a rock facies classification problem.* Comput. Geosci., 33, 599-617, doi: 10.1016/j.cageo.2006.08.011.

Goodfellow I., Bengio Y. and Courville A.; 2016: *Deep Learning.* MIT Press, Cambridge, MA, USA, 801 pp.

Hinton G.E., Deng L., Yu D., Dahl G., Mohamed A.-R., Jaitly N., Senior A., Vanhoucke V., Nguyen P., Sainath T. and Kingsbury B.; 2012: *Deep neural networks for acoustic modeling in speech recognition.* IEEE Signal Process. Mag., 29, 82-97.

Krizhevsky A., Sutskever I. and Hinton G.E.; 2012: *Imagenet classification with deep convolutional neural networks*. Adv. Neural Inf. Process. Syst., 25, 1-9.

LeCun Y., Bengio Y. and Hinton G.; 2015: *Deep learning.* Nat., 521(7553), 436-444.

Lee D., Ovanger O., Eidsvik J., Aune E., Skauvold J. and Hauge R.; 2024: *Latent diffusion model for conditional reservoir facies generation*. Comput. Geosci., 194, doi: 10.1016/j.cageo.2024.105750.

Li K., Liu W., Dou Y., Xu Z., Duan H. and Jing R.; 2023: C*ONSS: contrastive learning method for semisupervised seismic facies classification.* IEEE J. Selec. Topics Appl. Earth Obs. Rem. Sens., 16, doi: 10.1109/JSTARS.2023.3308754.

Ma X., Spinner S., Venditti A., Li Z. and Tang S.; 2019: *Initial margin simulation with deep learning.* SSRN Electron. J.,, doi: 10.2139/ssrn.3357626.

Ofoh I.J., Onyekuru S.O., Ikoro D.O., Opara A.I., Njoku I.O., Okereke C.N. and Akakuru O.C.; 2023: *Enhancing facies classification in geological studies through artificial neural networks: a review.* Sustainable Earth Review, 3, 31-45, doi: 10.48308/ser.2024.234950.

Shang H., Cheng L., Huang J., Wang L. and Yin Y.; 2023: *A deep learning method for facies recognition from core images and its application: a case study of Mackay River oil sands reservoir*. Energies, 16, 465, doi: 10.3390/en16010465.

Xie J., Zhang Y. and Li Z.; 2020: *Lithological facies classification using deep convolutional neural network.* J. Pet. Sci. Eng., 195, 107470.

Yang X., Yuan R., Lv Y., Li L. and Song H.; 2022: *A novel multivariate cutting force-based tool wear monitoring method using a one-dimensional convolutional neural network.* Sens., 22, doi: 10.3390/s22218343.

Zhang X.-Y., Wang W.-L., Hu G.-M. and Yao X.-M.; 2024: *KG-Unet: a knowledge-guided deep learning approach for seismic facies segmentation*. Earth Sci. Inf., 17, 1967-1981, doi: 10.1007/s12145-024-00634-6.

Corresponding author: Mohammad Ali Riahi
Institute of Geophysics, University of Tehran
North Kargar Ave., Tehran, Iran
Phone: +98 21 61118219; e-mail: mariahi@ut.ac.ir