







Hybrid Modeling of Deep Neural Networks and Ordinary Logistic Regression for Predicting Groundwater Quality

Wigbertus Ngabu^{1*}, Henny Pramoddyo², Atiek Iriany², Rahma Fitriani²

¹ Department of Mathematics, Faculty of Natural Sciences, Brawijaya University, Malang 65141, Indonesia

² Department of Statistics, Faculty of Natural Sciences, Brawijaya University, Malang 65141, Indonesia

Corresponding Author Email: bertongabu@gmail.com

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/mmep.120304>

ABSTRACT

Received: 23 December 2024

Revised: 8 February 2025

Accepted: 14 February 2025

Available online: 31 March 2025

Keywords:

groundwater quality, DNN, OLR, hybrid modeling

Groundwater quality plays a crucial role in daily life and public health, particularly in regions that rely on groundwater resources for domestic, agricultural, and industrial needs. Predictive modeling of groundwater quality is essential for managing and protecting water resources sustainably. This study proposes a hybrid approach combining deep neural networks (DNNs) and ordinary logistic regression (OLR) to enhance the accuracy of groundwater quality predictions. This hybrid approach leverages the capability of DNNs to capture complex non-linear patterns, while OLR is utilized for simpler and more structured coefficient interpretation of factors influencing water quality. The data used in the study includes various environmental and hydrogeological variables affecting groundwater quality, such as pH, heavy metal content, and other minerals. The results indicate that the hybrid DNN-OLR model achieves higher predictive accuracy, at 92.23%, compared to using DNN or OLR individually, which yield accuracies of 58.80%. The integration of these two methods also offers advantages in result interpretation, with OLR providing more transparent insights into the influence of independent variables, while DNN delivers stronger predictive capabilities through non-linear data processing. Therefore, this hybrid model has the potential to be applied for real-time groundwater quality monitoring and as a decision-support tool in water resource management.

1. INTRODUCTION

In reality, groundwater is responsible for the necessary clean water sustaining most human life, especially in rural areas without access to large networks of water distribution systems. Groundwater is of utmost importance, and this topic is key in providing water security in the case of low infrastructure regions. For example, groundwater is a vital source of water for over 60% of the rural population in South Asia for daily requirements ranging from household consumption to agricultural irrigation [1]. Groundwater is the lifeblood of Sub-Saharan Africa, where surface water distribution systems rarely penetrate isolated rural areas. Groundwater plays a pivotal role in the sustenance of human life, as the United Nations World Water Development Report indicates that about 50% of the global population as well as 70% of the water requirements of the agricultural sector are dependent on groundwater.

Unfortunately, groundwater quality is severely threatened by human activities, climate change, and local geology [2]. This degradation of freshwater resources in the region has been aggravated by the phenomenon of saltwater intrusion due to the over-extraction of groundwater in the Middle East, compromising irrigation and drinking water quality and potentially questioning the sustainability of water sources in

this region. Increased mining activity, especially in South America, has led to pollution, including heavy metals such as arsenic and mercury in groundwater from the Andes, with serious health consequences [3]. Another example can be found in Southeast Asia and Kalimantan, where mining activities have resulted in extremely high levels of heavy metals in groundwater that negatively impact both local communities and ecosystems [4]. These challenges suggest that sustainable groundwater management is not only a regional issue but a global imperative.

OLR is the most commonly used predictive method for groundwater quality (distribution). One reason OLR is commonly employed is its relative simplicity, as it allows for the modeling of linear relationships between independent and dependent variables. OLR has already been successfully used, for example, in the assessment of the impact of water quality on public health [5] and in environmental risk assessments [6]. The most important benefit of this approach is easy interpretation, which makes it comprehensible for policymakers and other stakeholders. As a result, this is a very limited method. When relationships are non-linear or involve complex interactions, OLR struggles to explain relationships, as shown in interactions between heavy metals, organic materials, and hydrogeological parameters [7].

Recently, DNN arises as a better option to create and

optimize models that deal with complex datasets that are associated with variables [8]. DNN works best in identifying complex linear relationships in multi-dimensional data, such as the association between pH, heavy metals, and organic substances in the quality of groundwater [9]. On the other hand, DNN is one of the most accurate predictive models, justifying its efficiency but making the interpretability of its results an increasingly challenging task. In the environmental domain, the results should be clear and actionable for stakeholders like environmental regulators and policymakers, who need to know what can be done [10]. It is thus necessary to combine the advantages of DNN with existing methods such as OLR to close the trade-off between predictive validity and interpretability/clarity.

DNN has been widely used in many areas with great potential in existing studies. For instance, employed DNN-based STR for victim identification in mass casualty incidents, illustrating the versatility of DNN in dealing with complex data [11]. Similarly, Convolutional Neural Networks (CNNs) for fault diagnosis in three-phase induction motors, where technical data was very complex as well [12]. The demonstrated the satisfying ability of DNN to tackle difficulties with interacting variables by utilizing Spike Neural Networks to classify IoT device traffic [13]. Although these studies demonstrate the potential of DNN to solve complex problems, it has rarely been applied in groundwater quality contexts [14].

Combining DNN and OLR is motivated by their complementarity. DNN shines in finding complex non-linear relations, but OLR is interpretable. This has already been applied in other fields, e.g., MEDeep is a news-based deep learning-based model developed for emotion analysis [15]. The concept demonstrated in this study that integrating DNN with other methods can produce data analysis that is more resilient and applicable, providing inspiration for the use of DNN in groundwater quality management.

Additionally, the developed a tree-structured deep learning model with self-adaptation and self-learning capabilities to enhance data classification [16]. Their hybrid-based approach shows that integrating deep learning with traditional techniques can improve model accuracy while maintaining flexibility and scalability. In the context of groundwater quality, such an approach is highly relevant to address the challenges of modeling complex patterns.

However, although there have been efforts to combine deep learning methods with traditional approaches in other fields, very few studies have explicitly applied this approach to groundwater quality management. For example, demonstrated the successful application of DNN for analyzing technical data, but the context remains distant from environmental issues [17]. Therefore, this research bridges the gap by developing a DNN-OLR hybrid model specifically for groundwater quality.

The novelty of this research lies in developing a hybrid model that combines the strengths of OLR and DNN. This state-of-the-art approach integrates two different methodologies: OLR provides easily interpretable results, while DNN handles data complexity and non-linear patterns [18]. The hybrid model aims to enhance prediction accuracy without sacrificing interpretability. An innovative aspect of this model is its use of DNN to extract complex non-linear features, which are then simplified by OLR to produce more understandable interpretations. Technically, this integration involves feeding the output of the DNN's final hidden layer as

input to the OLR model. This process allows DNN to capture complex data patterns, while OLR models the linear relationships of enriched features. This method not only improves prediction accuracy but also facilitates deeper analysis of significant variables. This hybrid approach addresses the limitations of traditional models that struggle to effectively capture non-linear relationships while delivering results directly relevant to decision-making. Although this combination has been applied in fields like economics and healthcare, its specific application in groundwater quality prediction remains rare, offering a significant contribution to the literature [19].

In conclusion, the integration of DNN and OLR offers a unique opportunity to enhance predictive accuracy without compromising the interpretability of results. By leveraging DNN's capability to detect complex patterns and OLR's flexibility in providing simple interpretations, this approach has the potential to be a transformational solution to groundwater quality management challenges. This study makes a significant contribution to the development of innovative, applicable, and data-driven solutions in the environmental sector. In the future, this approach can be expanded to include additional environmental variables or applied to other resource management issues, such as water distribution systems in urban areas under pressure from rapid urbanization. Visualizations, such as flowcharts, can also help improve understanding of this hybrid model concept.

2. METHODS

2.1 Data

This research focuses on the analysis of groundwater quality in Yogyakarta City. The data used in this study were collected from groundwater samples taken from various locations across Yogyakarta City. Each sample was then associated with its geographical location using accurate GPS coordinates. The objective of this research is to develop a detailed mapping of groundwater quality in the Yogyakarta area, covering various chemical and physical parameters.

Table 1. Summary of research variables

Variables	Variable Name	Measurement Scale	Information
Y	Groundwater Quality Index	Nominal (Ordinal)	Response
X_1	pH	ratio	Predictor
X_2	Turbidity (NTU)	ratio	Predictor
X_3	Total Dissolved Solid (TDS) (mg/L)	ratio	Predictor
X_4	Nitrate (NO ₃) - N (mg/L)	ratio	Predictor
X_5	Nitrite (NO ₂) - N (mg/L)	ratio	Predictor
X_6	Iron (mg/L)	ratio	Predictor
X_7	Manganese (mg/L)	ratio	Predictor
X_8	Cyanide (mg/L)	ratio	Predictor
X_9	Fluoride (mg/L)	ratio	Predictor
X_{10}	Total coliform (CFU/100 ml)	ratio	Predictor

The variables observed in this study as shown in Table 1 include the chemical quality of groundwater, such as pH, heavy metal content, such as lead and cadmium, nitrates, phosphates, and organic matter. Additionally, physical properties of groundwater, such as turbidity, color, and odor, were also considered. This research also includes microbiological quality, including the presence of coliform bacteria and other pathogens. Sampling was systematically conducted at various locations to obtain a broad representation of groundwater conditions in Yogyakarta City. Each groundwater sample was labeled with accurate GPS coordinates, allowing for the use of geographical approaches in its analysis.

2.2 OLR

Binary logistic regression is extended by OLR [20]. OLR is a statistical technique for analyzing data in which the predictor variables are factors (if using nominal or ordinal scales) or covariates (if using interval or ratio scales) and the response variable is ordinal, with three or more categories [21].

The logit model, more especially the cumulative logit model, is the model most frequently employed for OLR [22]. Cumulative probabilities in this model represent the response Y ordinal nature. The cumulative logit model compares the cumulative probability, that is, the probability of being less than or equal to the G -th response category, $P(Y \leq g|\mathbf{x}_i)$, with the probability of being greater than the g -th response category, $P(Y > g|\mathbf{x}_i)$, based on p predictor variables represented as a vector x [20].

Use a cumulative logit model to represent OLR. Cumulative probabilities are used in this logit model to describe the response Y ordinal nature [23]. $P(Y = 1|\mathbf{x})$ can be expressed as $\pi(\mathbf{x})$, which is represented as follows, to create logistic regression:

$$\pi_g(\mathbf{x}_i) = \frac{\exp(\alpha_g + \mathbf{X}_i^T \boldsymbol{\beta})}{1 - \exp(\alpha_g + \mathbf{X}_i^T \boldsymbol{\beta})} \quad (1)$$

where, \mathbf{X}_i^T : The independent variables, with $(i=1, 2, \dots, n)$, where n represents the total number of samples; $\pi(\mathbf{x})$: The probability of a successful event; α_g : The constant (intercept); $\boldsymbol{\beta}_k$: The value of the k -th parameter ($k=1, 2, \dots, n$); g : The response categories, with $g=1, 2, \dots, G-1$.

Logistic regression is a part of generalized linear models. For OLR, the model used is the cumulative logit model. Suppose the response variable Y has G ordinal categories, and x_i represents the predictor variable vector for the i -th observation, expressed as:

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{bmatrix}^T$$

where, $i=1, 2, 3, \dots, n$, then the OLR model can be written as [24]:

$$\text{Logit}[P(Y_i \leq g|\mathbf{x}_i)] = \ln \left[\frac{P(Y_i \leq g|\mathbf{x}_i)}{1 - P(Y_i \leq g|\mathbf{x}_i)} \right] = \alpha_g + \mathbf{x}_i^T \boldsymbol{\beta}$$

where, $P(Y_i=g|\mathbf{x}_i)$ represents the probability that the response variable in the i -th observation falls within category g . Suppose $\pi_g(\mathbf{x}_i)=[P(Y_i=g|\mathbf{x}_i)]$, then:

$$\begin{aligned} P(Y_i \leq g|\mathbf{x}_i) &= P(Y_i = 1|\mathbf{x}_i) + P(Y_i = 2|\mathbf{x}_i) + \dots \\ &\quad + P(Y_i = g|\mathbf{x}_i) \\ &= \pi_1(\mathbf{x}_i) + \pi_2(\mathbf{x}_i) + \dots + \pi_g(\mathbf{x}_i) \end{aligned} \quad (2)$$

Thus, the probability for each response category can be expressed as:

$$\pi_g(\mathbf{x}_i) = P(Y_i = g|\mathbf{x}_i) = P(Y_i \leq g|\mathbf{x}_i) - P(Y_i \leq g-1|\mathbf{x}_i)$$

If the response variable has four categories ($G=4$), then the OLR model follows the formulation [24]:

$$\text{logit}[P(Y_i \leq 1|\mathbf{x}_i)] = \ln \left[\frac{P(Y_i \leq 1|\mathbf{x}_i)}{1 - P(Y_i \leq 1|\mathbf{x}_i)} \right] = \alpha_1 + \mathbf{x}_i^T \boldsymbol{\beta}$$

$$\text{logit}[P(Y_i \leq 2|\mathbf{x}_i)] = \ln \left[\frac{P(Y_i \leq 2|\mathbf{x}_i)}{1 - P(Y_i \leq 2|\mathbf{x}_i)} \right] = \alpha_2 + \mathbf{x}_i^T \boldsymbol{\beta}$$

$$\text{logit}[P(Y_i \leq 3|\mathbf{x}_i)] = \ln \left[\frac{P(Y_i \leq 3|\mathbf{x}_i)}{1 - P(Y_i \leq 3|\mathbf{x}_i)} \right] = \alpha_3 + \mathbf{x}_i^T \boldsymbol{\beta}$$

$$\text{logit}[P(Y_i \leq 4|\mathbf{x}_i)] = \ln \left[\frac{P(Y_i \leq 4|\mathbf{x}_i)}{1 - P(Y_i \leq 4|\mathbf{x}_i)} \right] = \alpha_4 + \mathbf{x}_i^T \boldsymbol{\beta}$$

where:

$$P(Y_i \leq 1|\mathbf{x}_i) = \frac{\exp(\alpha_1 + \mathbf{X}_i^T \boldsymbol{\beta})}{1 + \exp(\alpha_1 + \mathbf{X}_i^T \boldsymbol{\beta})} \quad (3)$$

$$P(Y_i \leq 2|\mathbf{x}_i) = \frac{\exp(\alpha_2 + \mathbf{X}_i^T \boldsymbol{\beta})}{1 + \exp(\alpha_2 + \mathbf{X}_i^T \boldsymbol{\beta})} \quad (4)$$

$$P(Y_i \leq 3|\mathbf{x}_i) = \frac{\exp(\alpha_3 + \mathbf{X}_i^T \boldsymbol{\beta})}{1 + \exp(\alpha_3 + \mathbf{X}_i^T \boldsymbol{\beta})} \quad (5)$$

2.3 DNN

A DNN consists of multiple layers of interconnected nodes (neurons). The basic unit of a DNN is the neuron, which performs a weighted summation of the inputs and passes it through an activation function. Mathematically, a neuron can be represented as [25]:

$$z = \mathbf{W}^T \mathbf{X} + b \quad (6)$$

where, X is the input vector, W is the weight, and b is the bias. The result is then passed through the activation function σ to produce the output:

$$\alpha = \sigma(z) \quad (7)$$

2.4 Hyperparameters in DNN

Hyperparameters are settings that are set before the model is trained and don't change while it is being trained. The performance of the model and its capacity to generalize to new data are greatly impacted by hyperparameters in the context of DNN [26]. Hyperparameters need to be manually adjusted or optimized, in contrast to model parameters like weights and biases, which are modified during training using optimization procedures.

Hyperparameters in DNN encompass various aspects, including network architecture, training strategies, and regularization techniques. Key hyperparameters include:

(1) Layer

The number of layers in a DNN determines the complexity of patterns that the model can learn. Additional hidden layers

enable the network to capture non-linear patterns in data, which is often essential for complex tasks. However, an excessive number of layers increases the risk of overfitting, where the model becomes overly specialized to the training data and performs poorly on unseen data. Techniques such as regularization (e.g., L1 and L2) and early stopping can mitigate this risk by halting training when validation performance starts to decline [27].

(2) Units

Each hidden layer consists of multiple units (neurons), which define the representational capacity of the network. A higher number of units allows the model to capture more intricate patterns, but it also demands greater computational resources and raises the risk of overfitting. Experimental approaches, such as grid search and random search, are commonly employed to determine the optimal number of units. Through these methods, an appropriate configuration can be identified based on the nature of the data and the complexity of the task [28].

(3) Activation function

The activation function enables the model to learn non-linear relationships. Functions such as Rectified Linear Unit (ReLU) are widely used due to their efficiency and ability to mitigate the vanishing gradient problem. In contrast, the sigmoid function is suitable for probability-based tasks but often suffers from saturation, while the tanh function is beneficial for data centered around zero. The selection of an appropriate activation function depends on the specific problem at hand, as each function has its own advantages and limitations [29].

(4) Epoch

The number of epochs defines how many times the model iterates over the entire dataset during training. A higher number of epochs allows the model to learn more complex patterns, but it also increases the risk of overfitting, where the model memorizes the training data rather than generalizing to new data. To mitigate this risk, early stopping is highly recommended, as it halts training when validation performance ceases to improve, ensuring better generalization on unseen data [30].

2.5 Basic DNN architecture

Many neurons make up the network that is the biological nervous system. Similarly, the fundamental processing units of artificial neural networks are neurons. As seen in Figure 1 [31], the operating premise is that a number of input values are mathematically transformed to produce an output value.

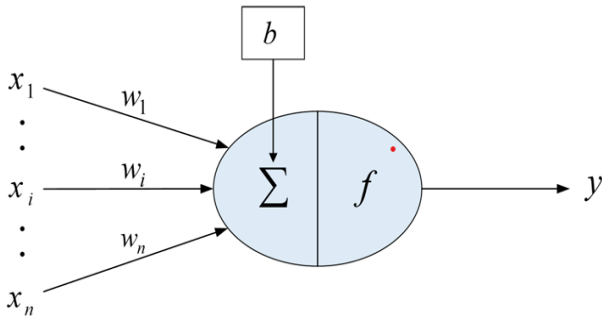


Figure 1. Illustration of a DNN

The mathematical transformation relationship between the

input signals and the output value is as follows:

$$f(b + \sum_{i=1}^n (x_i \times w_i)) \quad (8)$$

where, $f(\cdot)$ represents the activation function, and there are many types of activation functions, such as ReLU, Sigmoid, Tanh, among others.

In a DNN, a neuron receives input $\mathbf{x} = [x_1, x_2, x_3, \dots, x_n]$, which is a vector from the neurons in the previous layer. Each input is multiplied by its corresponding weight $\mathbf{W} = [w_1, w_2, \dots, w_n]$, where, \mathbf{W} is the weight matrix that determines the strength of the connections between neurons. A bias (\mathbf{b}) is added to adjust the output and increase the model's flexibility [32].

$$z = \mathbf{W} \cdot \mathbf{x} + \mathbf{b} \quad (9)$$

here, z is the result of the linear combination of inputs. To introduce non-linearity, the activation function $\phi(z)$ is applied to the result of this linear combination [33]:

$$y = \phi(z) \quad (10)$$

In the hidden layers of a DNN, this operation is repeated at each layer, with the output y from the previous layer becoming the input x for the next layer [34]. Mathematically, for the l -th layer in a DNN, the output of a neuron in that layer can be expressed as:

$$y^l = \phi(\mathbf{W}^l y^{l-1} + \mathbf{b}^l) \quad (11)$$

Activation functions

Activation functions are a critical component in DNNs as they introduce non-linearity into the network. Without activation functions, a DNN would simply be a series of linear operations, limiting its ability to learn non-linear relationships in the data. Below are some commonly used activation functions:

1) Sigmoid

The sigmoid activation function converts the input into a value within the range of 0 to 1, which is highly useful in the context of probabilities. The sigmoid function is often used in the output layer for binary classification tasks.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (12)$$

This function transforms the input into values within a categorical range [35].

2) ReLU

ReLU is one of the most popular activation functions in modern neural networks. It activates neurons only if the input is positive, and the output is equal to the input; if the input is negative, the output is zero.

$$f(x) = \max(0, x) \quad (13)$$

This function activates units only when the input is positive, making it simple and fast to compute. The main advantages of ReLU are its simplicity, computational efficiency, and its ability to mitigate the vanishing gradient problem often seen with sigmoid and tanh functions [36].

2.6 DNN ordinary logistic regression

The hybrid DNN-OLR model leverages the strengths of both approaches. DNN is utilized to capture complex patterns and features from the data through its hidden layers. This is especially useful in situations where the data contains many interacting non-linear features [37]. Meanwhile, OLR is applied in the output layer to provide well-interpreted probabilities for binary outcomes. In this way, the hybrid DNN-OLR model delivers results that are both more accurate and easier to interpret.

To create a hybrid model, the output from logistic regression can be combined with a DNN. A common approach is to use DNN to learn complex features from the input data and then feed these features into a logistic regression model for the final prediction.

In a hybrid model that integrates DNN and logistic regression, the feature extraction process plays a crucial role. Before the model makes a final classification decision, it first learns and extracts meaningful patterns from the input data using a DNN. The extracted features are then used as input for logistic regression to perform classification. The primary goal of feature extraction is to transform raw data into a more representative and informative form, thereby improving accuracy and efficiency in the classification process.

1) Feature extraction process using DNN

A DNN consists of multiple hidden layers, each designed to gradually transform and filter the input data. Instead of directly using raw data for classification, DNN allows the model to capture complex relationships and hidden structures within the data [38]. Each layer in the DNN performs a mathematical transformation on the input data, producing an output that is then passed to the next layer.

Mathematically, the transformation performed by each hidden layer can be represented as:

$$H_i = \sigma(\mathbf{W}_i^T \mathbf{H}_{i-1} + \mathbf{b}_i) \quad (14)$$

where, \mathbf{H}_i is the output of the i^{th} hidden layer, representing the learned features at that stage; \mathbf{W}_i is the weight matrix, which is optimized during the training process; \mathbf{b}_i is the bias for the i^{th} layer; σ is the activation function, such as ReLU or Sigmoid.

2) Stages in the feature extraction process

Feature extraction is one of the crucial stages in the machine learning process, especially in the context of a hybrid model that combines DNN and logistic regression. The DNN functions as a tool to construct richer and more meaningful feature representations by transforming raw data into a more informative form [39]. This process occurs through a series of layers that gradually abstract information from the initial data.

The initial stage of feature transformation begins with the input layer, which receives data in the form of a feature vector. Mathematically, the input data can be represented as:

$$\mathbf{X} = (x_1, x_2, \dots, x_n)$$

where, each component x_i represents a specific feature obtained from the original dataset. This feature can be either raw data or the result of a pre-processing step, such as normalization, statistical attribute extraction, or domain-specific transformations.

After passing through the input layer, the data is then processed by the first hidden layer in the DNN. This process

involves a linear operation, consisting of matrix multiplication between the network weights and the input vector, added with bias, and then passed through a non-linear activation function. This transformation can be expressed as:

$$H_1 = \sigma(\mathbf{W}_1^T \mathbf{X} + \mathbf{b}_1) \quad (15)$$

where, \mathbf{H}_1 is the transformed output at the first hidden layer, representing learned features at this stage; \mathbf{W}_i represents the weights connecting the input layer to the first hidden layer.

The result of this transformation process is a new feature vector, which is more meaningful compared to the original raw input data. The first hidden layer acts as an initial filter, extracting fundamental patterns from the data. However, to capture more complex and abstract patterns, the data needs to be forwarded to the next hidden layers.

Deeper feature extraction is performed gradually through a stacked series of hidden layers. The transformation of features at the k^{th} layer is formulated as:

$$H_2 = \sigma(\mathbf{W}_2^T \mathbf{H}_1 + \mathbf{b}_2) \quad (16)$$

$$H_3 = \sigma(\mathbf{W}_3^T \mathbf{H}_2 + \mathbf{b}_3) \quad (17)$$

⋮

$$H_k = \sigma(\mathbf{W}_k^T \mathbf{H}_{k-1} + \mathbf{b}_k) \quad (18)$$

Each hidden layer in the DNN functions as a filter, combining and forming new abstract features compared to the previous layers. The deeper the network, the more complex patterns it can capture. Thus, the number of hidden layers and the size of each layer significantly affect the DNN's ability to extract features.

At the final transformation stage, the last hidden layer H_k contains a set of optimally extracted features, representing the best data representation for classification. These features provide richer and more informative data than the original input. These extracted features are then used as input for logistic regression, which acts as a classifier to generate the final prediction.

The model follows this structure:

$$H_k = \sigma_k(\mathbf{W}_k^T (\sigma_{k-1}(\mathbf{W}_{k-1}^T (\dots \sigma_1(\mathbf{W}_1^T \mathbf{X} + \mathbf{b}_1) \dots) + \mathbf{b}_{k-1}) + \mathbf{b}_k)) \quad (19)$$

With this feature extraction process, the model can reduce the dimensionality of less relevant features while retaining and highlighting the most important information for the classification task. The combination of DNN's capability to capture non-linear patterns and the interpretability of Logistic Regression makes this approach highly effective in various machine learning applications, including binary classification, predictive analytics, and pattern recognition.

3) The role of the activation function in feature extraction

The activation function plays a critical role in feature extraction within a DNN by introducing non-linearity into the model. This non-linearity enables the network to capture and represent complex relationships in data, which cannot be captured by purely linear models.

Without an activation function, neural networks would only perform linear operations, making them incapable of recognizing more abstract patterns or handling non-linear feature relationships.

In this study, the activation function used is sigmoid, which is commonly applied in binary classification tasks. The sigmoid function is mathematically defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (20)$$

The sigmoid function is used in logistic regression, which serves as the final classification component in this hybrid model. Using sigmoid in the last DNN layer ensures consistency between the DNN output and the final classification model, making the transition seamless.

4) Utilizing extracted features in logistic regression

After the DNN processes the input and generates the final feature representation in the last layer, these features \mathbf{H}_k are used as input for logistic regression to make the final classification decision.

The probability of a class y given the extracted features \mathbf{H}_k is computed as:

$$P(y = 1|\mathbf{H}_k) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 H_{k1} + \beta_2 H_{k2} + \dots + \beta_m H_{km})}} \quad (21)$$

where, $\mathbf{H}_k = (H_{k1}, H_{k2}, \dots, H_{km})$ are the features from the final DNN layer; $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_m)$ are the coefficients of the logistic regression.

The feature extraction process in this hybrid model utilizes DNN to construct more representative features from the raw data. The layers in the DNN progressively transform the input data into more abstract and meaningful features, which are then used by logistic regression for the final prediction. With this approach, we leverage the DNN's ability to capture complex patterns while benefiting from the speed and interpretability of logistic regression.

The hybrid approach, which combines DNN with logistic regression, offers several key advantages in the classification process. These benefits primarily stem from the DNN's ability to extract richer and more complex features before being fed into logistic regression for final decision-making.

One of the major advantages is the DNN's ability to extract more informative features compared to the raw features from the initial data. With multiple hidden layers, the DNN performs gradual transformations on the data, resulting in more meaningful and structured feature representations for classification tasks.

Additionally, DNN has the capability to capture non-linear relationships in the data, which are difficult for purely linear models like logistic regression to handle. Through activation functions such as ReLU, Sigmoid, or Tanh, DNN can model more complex patterns, allowing the model to perform more effectively in scenarios where relationships between features are non-linear.

Another advantage is DNN's ability to automatically reduce data dimensionality. By passing through multiple transformation layers, the DNN efficiently selects the most relevant features for final classification. This helps improve model accuracy while reducing noise from less informative features, ultimately making logistic regression more efficient in processing the enriched data.

2.7 Evaluation criteria for hybrid model performance

To comprehensively assess the performance of the deep neural network with optimized logistic regression (DNNOLR)

hybrid model compared to standalone models, multiple evaluation metrics are employed. While accuracy serves as a fundamental metric, it alone does not provide sufficient insight into the model's effectiveness, particularly in imbalanced datasets. Therefore, additional metrics, such as Accuracy, precision, recall, and F1-score, are considered to offer a more holistic evaluation [40].

1) Accuracy

Accuracy represents the ratio of correct predictions to the overall number of predictions made. The accuracy metric offers a general performance evaluation but becomes unreliable when class imbalance exists because a model can reach high accuracy by predominantly predicting the majority class.

2) Precision

The positive predictive value, or precision, quantifies the fraction of true positive results within all positive classifications. Applications that require minimizing false positives rely heavily on precision particularly in fields like fraud detection and medical diagnosis. The precision measure reflects the number of false positives which means higher precision values lead to more trustworthy positive predictions.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (22)$$

3) Recall

Recall, also referred to as sensitivity or the true positive rate, measures the proportion of actual positive instances that are correctly identified by the model [41]. It is a critical metric in scenarios where false negatives need to be minimized, such as in disease detection, where missing a positive case can have serious consequences.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (23)$$

4) F1-score

The F1-score represents the harmonic mean of precision and recall, providing a single metric that balances the trade-off between the two. It is particularly useful when both false positives and false negatives are costly, ensuring an optimal balance between the two errors.

$$\text{Precision} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (24)$$

2.8 Illustration diagram of the hybrid DNN-OLR model

In this study, a hybrid model combining DNN and OLR was developed to enhance accuracy and interpretability in predictions. This hybrid model leverages the ability of DNN to capture complex patterns in data while utilizing the advantage of logistic regression in providing more transparent and interpretable results. With this approach, the model is expected to generate more optimal predictions compared to single-method approaches. The following diagram illustrates the workflow of the hybrid DNN-OLR model, from data preprocessing to the evaluation of the optimized model.

Figure 2 illustrates the key stages in building the hybrid DNN-OLR model. The process begins with inputting raw data, which then undergoes preprocessing and feature selection before being used to train the DNN model. After applying the activation function, the performance of the DNN model is

evaluated. If the model performs satisfactorily, features are extracted from the DNN and used to train the logistic regression model. The predictions from both models are then combined to generate the final prediction before the model is evaluated. However, if the DNN performance is unsatisfactory, hyperparameter tuning and retraining of the DNN model are conducted. This process continues until an optimized hybrid model is obtained as the final output.

2.9 Justification for combining DNN and OLR

The choice of methods in data analysis and prediction highly depends on the complexity of the data and the relationships between variables. Logistic regression (Ordinary, Multinomial, and Independent Models) is often used for analyzing data with linear relationships, while DNN and other deep learning-based methods excel in handling complex and nonlinear relationships.

Table 2 presents several studies that utilize various methods, datasets, and the accuracy levels achieved. The purpose of this analysis is to highlight the strengths and weaknesses of each approach, providing a basis for combining DNN and OLR.

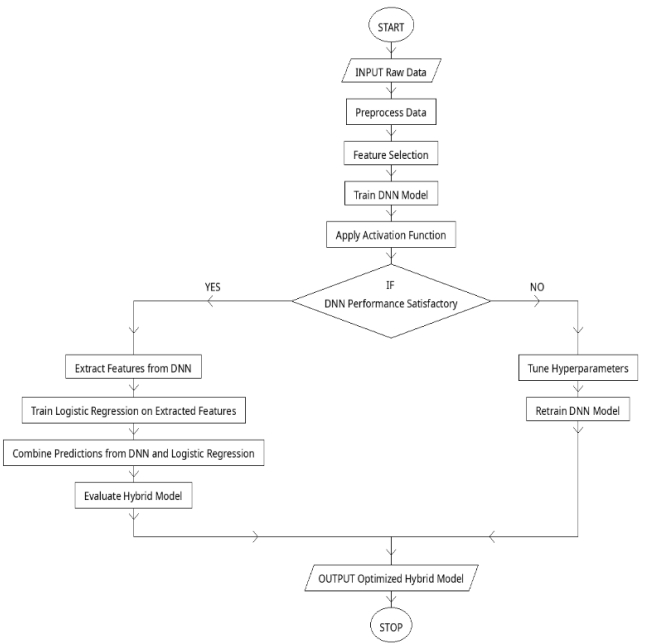


Figure 2. Hybrid DNNOLR flow diagram

Table 2. Justification for combining DNN and OLR

Study	Method	Dataset	Accuracy	Weaknesses
Venkataraman and Uddameri [42]	Logistic regression (Ordinary, multinomial, and independent models) + geographic information system (GIS)	Water quality data from Texas Water Development Board (TWDB), Soil data from SSURGO (1980s-2010)	74%	Cannot handle nonlinear relationships
Müller et al. [43]	Recurrent neural network (RNN), CNN	Groundwater level data from Butte County, California, USA (2010-2018)	79%	Requires extensive training
Alabdulkreem et al. [44]	Stacked LSTM with DNN (SLSTM-DNN), compared with DNN, and LSTM	Water NSW dataset for groundwater monitoring and prediction, incorporating climate and hydrological data	82%	High computation, complex tuning, data dependency, and real-time challenges
Nhu et al. [45]	DNN + GIS	Groundwater spring data from Kon Tum province, Vietnam, including 733 groundwater spring locations and 12 influencing factors processed using ArcGIS Pro	74%	High data reliance, costly computation, and limited generalizability

3. RESULTS AND DISCUSSION

3.1 DNN

In this study, we implemented a DNN model consisting of multiple hidden layers, with input from 10 variables (X_1 - X_{10}) combined to predict the target variable (Y). Each layer in this DNN is fully connected, meaning every neuron in a given layer is connected to all neurons in the next layer through weights that are optimized during the training process.

Figure 3 illustrates the DNN architecture consisting of 10 inputs, 2 hidden layers, and a single output. Each hidden layer functions to capture complex non-linear patterns from the data, where each layer acts as a transformation of the input features toward better prediction in the output layer. The weights connecting the neurons in each layer are adjusted through the backpropagation algorithm with the goal of minimizing prediction errors.

The results from applying DNN in this study indicate that networks with deeper architectures are capable of learning more complex patterns and improving predictive performance.

However, the selection of the number of layers and neurons is crucial, as an overly deep architecture can increase computation time and the risk of overfitting. Therefore, finding an optimal balance between model complexity and generalization is necessary.

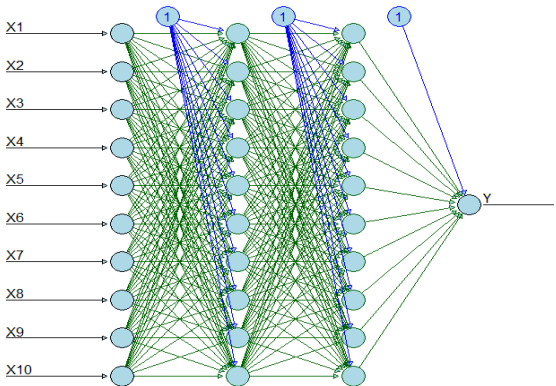


Figure 3. Basic architecture of the DNN for research data

In model validation, the DNN is evaluated based on two key metrics: loss and accuracy. Loss reflects how well the model predicts the expected target, while accuracy indicates the model's success rate in making correct predictions during the training and validation process. By plotting loss and accuracy against the number of epochs, we can analyze the model's learning process, identify any patterns of overfitting, and evaluate the stability of the model's performance.

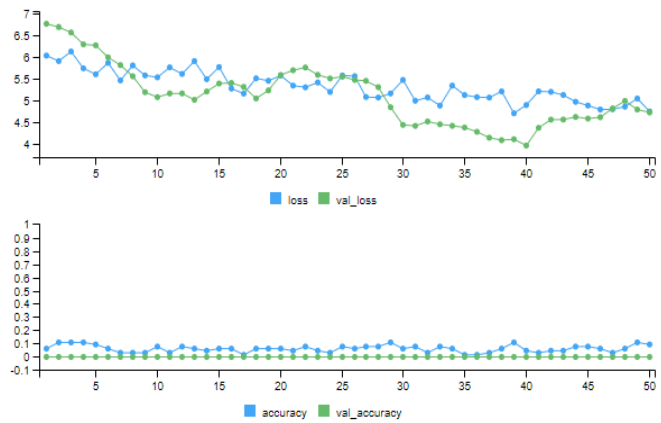


Figure 4. Validation graph of DNNs model

In Figure 4, the first graph shows the training loss and validation loss (val_loss) of the DNN model over 50 epochs. At the beginning of the training, the loss values for both training and validation are relatively high, with an initial loss around 6.5. As the number of epochs increases, the loss decreases significantly, especially up to the 25th epoch. However, from epoch 25 to epoch 50, fluctuations in the validation loss can be observed, which may indicate potential overfitting. Meanwhile, the training loss remains more stable, suggesting that the model is learning the training data better than the validation data.

In the second graph of Figure 4, the movement of accuracy and validation accuracy (val_accuracy) is shown, ranging from 0 to 0.1. The very low accuracy values indicate that the model has not yet achieved adequate predictive performance, although there is slight improvement in some epochs. The low accuracy values and the small difference between training and validation accuracy suggest that the model may be struggling to capture more complex data patterns, leading to underfitting. The model validation results are presented in Table 3.

Table 3. Validation values of the DNN model

Final Epoch	Values
Loss	1.4880
Accuracy	56.52%
Val_Loss	0.5989
Val Accuracy	62.86%
Total Accuracy	57.80%

Based on Table 3, the final result of the DNN training yielded a total accuracy of 57.80%, indicating that the model was able to make correct predictions for 57.8% of the entire dataset. Although this value is not yet optimal, it demonstrates that the model successfully captured a significant portion of relevant patterns from the data. However, there is still room for performance improvement.

Furthermore, the validation accuracy of 62.86% suggests that the model performed better on unseen data during the

training process, indicating a reasonably good potential for generalization. Given these results, further steps should be taken to extend the model into a hybrid approach with an OLR model.

3.2 Feature extraction

Feature extraction is the process of extracting specific information or characteristics from raw data to produce a more meaningful and useful representation for further analysis or machine learning models. This process aims to identify important patterns in the data that can assist in solving problems or making predictions.

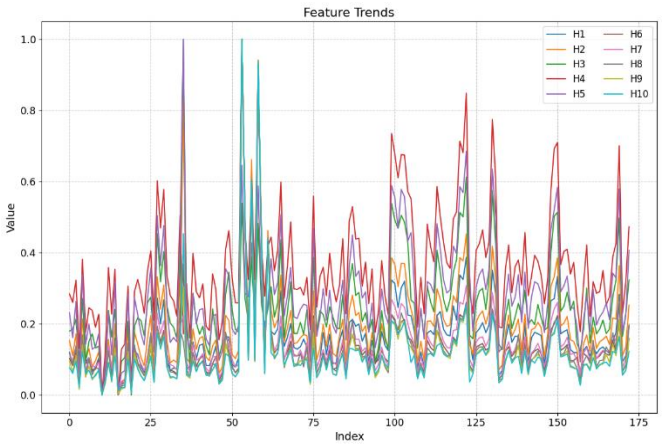


Figure 5. Validation graph of the DNNs model

Figure 5 illustrates the trend of values for ten features represented by labels H_1 to H_{10} on the vertical axis (Value) against the data index on the horizontal axis (Index). The graph shows the fluctuation of values for each feature within a certain range of indices. The primary purpose of this visualization is to understand the patterns or anomalies occurring in the data for each feature, providing insights into significant interactions or changes within the observed dataset.

Figure 4 shows that most features exhibit relatively stable values across the majority of indices, although there are some extreme peaks, particularly around index 50 and a few other points. Feature H_2 appears to have higher values compared to other features on several occasions, indicating a potential imbalance or uniqueness in the data for that feature. Feature extraction was performed to isolate 10 distinct features (H_1 to H_{10}) from the dataset, each demonstrating value dynamics across specific indices.

3.3 Hybrid DNNs ordinary logistic regression

This study introduces a hybrid DNN and OLR approach designed to enhance performance in ordinal classification. This hybrid method combines the strengths of DNN in extracting complex non-linear features with the power of OLR in modeling linear relationships between variables, which aligns with the nature of ordinal data.

In this approach, the DNN model serves as a feature extraction tool, where the hidden layers are used to filter and produce deeper feature representations from the input data. The features extracted by the DNN are then used as inputs for OLR, which subsequently predicts classification outcomes based on the linear structure of the data, now represented more informatively by the DNN. This process allows the OLR

model to work with data enriched by the DNN, thereby improving the overall prediction accuracy. This discussion outlines the significant contribution of DNN-extracted features to the performance of the OLR model and compares the effectiveness of this hybrid approach with standalone DNN and OLR approaches. The results show that the hybrid approach maximizes the model's ability to handle complex relationships, providing better classification accuracy on data with ordinal structure. The results of the hybrid model are presented in Table 4:

Table 4. Results of the hybrid DNNs and ordinary logistic regression model

Parameter	Estimate	Std. Error	z-value	p-value
α_1	4.641	4.331	1.072	-
α_2	5.597	4.346	1.518	-
α_3	7.607	4.358	1.746	-
H_1	0.0348	0.073	0.473	0.6361
H_2	0.446	0.577	0.773	0.4397
H_3	0.401	0.116	0.354	0.7233
H_4	-0.013	0.010	-0.133	0.1826
H_5	0.004	0.001	2.309	0.0209*
H_6	-0.064	0.047	-1.362	0.0173*
H_7	9.410	6.174	1.524	0.1275
H_8	-2.369	15.278	-0.155	0.8767
H_9	2.115	3.099	0.682	0.4951
H_{10}	-0.040	0.087	-0.460	0.5452

Based on the analysis results in Table 4, the p-value indicates that the parameters nitrate and nitrite, specifically H_5 (Nitrate, $P=0.0209$) and H_6 (Nitrite, $P=0.0173$), are statistically significant at the 5% significance level, suggesting that these two parameters contribute significantly to the model. This means that the variables associated with these parameters have a meaningful impact on the prediction outcomes. Conversely, other parameters such as H_1 , H_2 , H_3 , H_4 , H_7 , H_8 , H_9 , and H_{10} have p-values greater than 0.05, indicating they are not statistically significant and do not have enough influence on the model in the context of this analysis.

The analysis of groundwater quality in Yogyakarta City shows that the nitrate and nitrite parameters contribute significantly to the model used in the study. This is evident from the p-values of 0.0209 for nitrate (H_5) and 0.0173 for nitrite (H_6), both of which are below the 5% significance threshold. Therefore, these two parameters are statistically proven to affect groundwater quality in the region, indicating a strong correlation between nitrate and nitrite concentrations and the measured level of groundwater quality.

The significance of nitrate and nitrite on groundwater quality in Yogyakarta City carries important implications for public health. High levels of nitrate and nitrite in groundwater are often associated with contamination from nitrogen-based agricultural fertilizers or from suboptimal waste management systems. Elevated levels of these substances can pose serious health risks, such as methemoglobinemia (blue baby syndrome) in infants and potential cancer risks in adults. Therefore, effective management of pollution sources that increase nitrate and nitrite concentrations should be a key priority for stakeholders in Yogyakarta.

On the other hand, the analysis results show that other parameters, such as H_1 , H_2 , H_3 , H_4 , H_7 , H_8 , H_9 , and H_{10} , do not have a significant impact on the groundwater quality prediction model. With p-values above 0.05, these parameters are considered to have insufficient influence in this analysis. However, this does not imply that these factors are

unimportant; rather, the current focus for maintaining groundwater quality in Yogyakarta City should be more directed towards controlling nitrate and nitrite levels.

To evaluate the performance of the DNNOLR model in classifying data, several evaluation metrics are used to assess how well the model distinguishes between positive and negative classes as shown in Table 5. These metrics include accuracy, precision, recall (sensitivity), specificity, and F1-score, each providing a deeper insight into the model's effectiveness in making predictions. Below are the evaluation results of the model based on these metrics:

Table 5. Classification performance metrics DNNOLR

Metric	Value
Accuracy	92.23%
Precision	86.90%
Recall (Sensitivity)	87.64%
Specificity	82.34%
F1-score	88.12%

The DNNOLR model demonstrates outstanding performance in classification tasks, as reflected in various evaluation metrics. With an accuracy rate of 92.23%, the model is capable of correctly classifying the majority of the data, indicating a relatively low error rate in overall predictions. Furthermore, the precision score of 86.90% signifies that a high proportion of positive predictions are indeed correct. This metric is particularly crucial in scenarios where minimizing false positive errors is essential, such as in fraud detection or sensitive medical diagnoses. Additionally, the F1-score of 88.12% reflects a well-balanced trade-off between precision and recall, ensuring that the model not only effectively detects positive cases but also maintains a low error rate. This suggests that the DNNOLR model exhibits a solid and balanced performance in classification tasks.

3.4 Research findings

In this study, the DNN model was implemented to predict the target variable based on ten input variables. The model was designed with two hidden layers, which serve to capture complex non-linear patterns in the data. Each neuron in the hidden layers is fully connected to the neurons in the subsequent layer through weights optimized during training using the backpropagation algorithm. The study results indicate that a deeper DNN architecture allows the model to learn more complex patterns. However, it also increases the risk of overfitting if not properly configured. Therefore, balancing model complexity and generalization ability is a crucial factor in developing an optimal model.

During the validation phase, the model's performance was evaluated using loss and accuracy metrics. The validation graph shows a significant decrease in loss up to the 25th epoch, but after that, fluctuations in validation loss were observed. This phenomenon indicates potential overfitting, where the model learns very well on the training data but is less effective in generalizing to new data. Additionally, the low accuracy values, with a total accuracy of 57.80% and validation accuracy of 62.86%, suggest that the model is still experiencing underfitting. This means the model has not fully captured the complex patterns in the data, leading to suboptimal prediction results.

Furthermore, this study proposes a hybrid DNNOLR approach to improve ordinal classification performance. This

approach combines the advantages of DNN in extracting non-linear features with the strength of OLR in modeling linear relationships between variables. DNN is used as a feature extraction tool, where the hidden layers generate richer feature representations. These extracted features are then used as input for the OLR model, which performs classification based on the linear structure within the data. Through this approach, the hybrid model can capture complex patterns while leveraging linear relationships present in the dataset, thereby improving overall prediction accuracy.

The analysis results of the hybrid model indicate that only two parameters significantly influence the model: H_5 (nitrate) with a p-value of 0.0209 and H_6 (nitrite) with a p-value of 0.0173. These two parameters were statistically proven to have a substantial contribution to groundwater quality. In contrast, other parameters, such as H_1 , H_2 , H_3 , H_4 , H_7 , H_8 , H_9 , and H_{10} , had p-values greater than 0.05, indicating that these variables did not have a statistically significant impact on the predictive model in this study.

The advantage of this hybrid approach lies in its ability to capture complex patterns that were not well-identified in pure regression methods. The analysis results show that the feature transformation performed by DNN allows for better separation between groundwater quality categories. DNN extracts non-linear relationships from the data, which are then processed by OLR to obtain clearer statistical interpretations. This combination enables the identification of more complex patterns in the distribution of nitrate and nitrite levels, as well as their interactions with other factors, thereby improving prediction accuracy.

A deeper analysis of feature interactions reveals that nitrate and nitrite levels not only have individual influences but also exhibit synergistic patterns when combined with other environmental factors. The feature extraction results from DNN uncovered previously unseen patterns, such as the relationship between high nitrate levels and specific soil conditions that contribute to groundwater quality degradation. The model also identified that, in certain cases, nitrite levels could increase significantly under specific conditions that were not detected in simple linear models. Therefore, this approach allows for the identification of more specific risk factors that can be used for more effective groundwater management policies.

The findings of this study also provide critical insights into groundwater quality in Yogyakarta City. The high levels of nitrate and nitrite in groundwater can be associated with the use of nitrogen-based agricultural fertilizers and suboptimal waste management systems. The increase in these substances poses serious health risks, such as methemoglobinemia (blue baby syndrome) in infants and cancer risks in adults due to contaminated water consumption. Therefore, the study underscores that controlling pollution sources, particularly those contributing to elevated nitrate and nitrite levels in groundwater, should be a top priority for stakeholders in Yogyakarta City.

Despite its predictive accuracy advantages, the hybrid approach has several limitations that need to be considered. One of the main challenges is the higher computational resource requirement compared to simple regression methods. Training a DNN requires significant time and computational power, especially as the dataset size increases. If the dataset is limited, the model may struggle to produce stable and accurate predictions. Thus, the success of this approach highly depends on having a sufficiently large dataset to capture more

representative patterns.

Additionally, this model still carries the risk of overfitting if not properly controlled. Although DNN has the capability to capture complex relationships, if the dataset is insufficient or has an imbalanced distribution, the model may generate overly specific predictions for the training data, reducing its effectiveness on new data. Therefore, strategies such as regularization, dropout, or cross-validation should be implemented to ensure that the model retains good generalization capabilities.

Although some parameters were not significant in the predictive model, this does not mean that these factors have no role in influencing groundwater quality. However, in the context of this study, the primary focus in groundwater quality preservation and management efforts should be directed toward controlling nitrate and nitrite levels. Preventive measures such as regular groundwater quality monitoring, regulating nitrogen-based fertilizer use, and improving waste treatment systems are strategic efforts that need to be implemented to mitigate negative health impacts on the community.

Considering the advantages and limitations of this hybrid approach, future research directions could include exploring optimization methods to improve computational efficiency, applying regularization techniques to reduce overfitting risks, and collecting larger datasets to enhance model generalization.

3.5 Model evaluation

To improve prediction accuracy in classification models, this study proposes a hybrid DNNOLR approach. This approach leverages the capability of DNN to extract complex non-linear features from the input data, which are then used as inputs for the OLR model. The results from applying this hybrid model were compared to a traditional DNN model, showing a significant improvement in accuracy. This accuracy improvement reflects the hybrid model's ability to combine the strength of DNN in capturing complex patterns with more stable predictions through OLR. The model evaluation results are presented in Table 6.

Table 6. Model comparison

Model	Accuracy
DNN	57.80%
DNNOLR	92.23%

Table 6 shows the accuracy comparison between the DNN model and the hybrid DNNOLR model. The DNN model achieved an accuracy of 57.80%, indicating that it was able to correctly predict outcomes for approximately 57.80% of the test data. On the other hand, the hybrid DNNOLR model showed a significant accuracy improvement, achieving 92.23%. This indicates that the hybrid approach, which combines the feature extraction capabilities of DNN with logistic regression-based predictions, provides a substantial performance boost compared to the standalone DNN model.

3.6 Generalization of models and adaptation to different conditions

For a model to be widely applicable across various conditions, it is crucial to consider data characteristics, such as numerical or categorical scales, variable distributions (which

may be normal or skewed), and dataset patterns, including seasonal trends in financial data or feature correlations in medical data. Training a model with data that reflects environmental variations enhances its ability to recognize complex patterns and mitigates the risk of overfitting.

Transfer learning serves as an effective strategy to improve model adaptability. For instance, in the medical field, a deep learning model pre-trained on radiology image datasets can be repurposed to detect new disease types by fine-tuning with additional data. Similarly, in the financial industry, a model trained to detect suspicious transactions in one country can be adapted to another country's transaction patterns with minimal adjustments to local data. Using a pre-trained model as a foundation allows for targeted modifications on specific datasets from different conditions. This approach accelerates adaptation, reduces the need for training from scratch, and enhances model performance across various scenarios.

Evaluating model performance on diverse datasets is essential to assess its reliability. Commonly used metrics include accuracy, precision, recall, and F1-score for classification tasks, as well as mean squared error (MSE) and root mean squared error (RMSE) for regression tasks. Additionally, the area under the curve (AUC) metric provides a comprehensive assessment of the model's ability to distinguish between positive and negative classes. The hybrid DNNOLR model, which integrates the feature extraction capabilities of DNN with the predictive stability of OLR, offers advantages over traditional models.

A major challenge in deploying models across different regions is the variation in data distribution and feature characteristics. For example, in the healthcare sector, urban patient data may reflect disease patterns associated with air pollution, while rural data may be more related to limited access to medical care. Similarly, in the financial sector, digital transaction habits are more prevalent in urban areas, whereas remote regions may still rely on cash-based transactions. The hybrid DNNOLR model can adjust feature weights based on regional differences to improve prediction accuracy by accounting for these variations. For instance, in healthcare, disease patterns are influenced by environmental factors such as pollution and healthcare accessibility, while in finance, transaction behaviors vary by region. The hybrid DNNOLR model optimizes feature weights according to regional characteristics to enhance prediction accuracy.

To improve generalization, additional training with more diverse datasets is necessary, as data variations help models recognize broader patterns and reduce bias toward specific datasets. By exposing the model to a wide range of training scenarios, it becomes more adaptive to real-world conditions and generates more accurate predictions on previously unseen data. Techniques such as transfer learning, data augmentation, ensemble learning, and hyperparameter tuning can aid in model adaptation to different conditions. With this approach, the hybrid DNNOLR model emerges as an adaptive solution capable of delivering accurate predictions across multiple industry sectors.

Model parameter adjustments also play a crucial role in ensuring optimal performance when applied to datasets with different characteristics. In certain cases, parameters in both DNN and OLR may require modification to better align with new data distributions. Hyperparameter tuning techniques can help identify the most suitable parameter configurations for varying environmental conditions. In this study, methods such as Grid Search and Random Search are applied to optimize

parameters, including the number of hidden layers, the number of neurons, and the learning rate in DNN, as well as regularization values in OLR. The impact of these optimizations on model generalization is evident in improved accuracy when tested on datasets with different characteristics, reducing the risk of overfitting and enhancing predictive performance in diverse environmental conditions beyond the initial training data. By integrating these strategies, the DNNOLR model demonstrates significant potential for application across different conditions while maintaining high prediction accuracy.

4. CONCLUSIONS

Based on the research findings, the DNN model successfully captures non-linear patterns in the data; however, it still faces challenges in significantly improving accuracy. Therefore, the hybrid DNN-OLR approach is implemented to address these limitations. The Hybrid DNN-OLR approach demonstrates that the features extracted by DNN can enhance the performance of OLR in ordinal classification. The analysis results indicate that the nitrate (H_5) and nitrite (H_6) parameters have a significant influence on the model, with p-values below 0.05. This finding suggests that these two parameters play a crucial role in determining groundwater quality in Yogyakarta City.

The study results show that the hybrid model significantly improves predictive accuracy compared to using either the standalone DNN or OLR models. The integration of DNN, which excels at capturing non-linear patterns, with OLR, which is reliable in probabilistic interpretation, produces a more comprehensive model for mapping factors affecting groundwater quality. Moreover, this model can be adapted for application in various other environmental quality prediction cases. These findings provide a significant contribution to efforts in groundwater quality mitigation and management amid increasing environmental pressures.

Furthermore, this study offers deeper insights into how machine learning models can be integrated with conventional statistical techniques to enhance prediction accuracy in environmental issues. Through this hybrid approach, governments and stakeholders can better understand the key factors influencing groundwater quality and design more effective mitigation strategies. Future studies may focus on exploring additional features or improving model architecture to further enhance predictive performance and expand the application of this model to other regions with different environmental characteristics.

For future research, integrating spatial aspects into predictive models could be a promising direction, considering the geographical distribution of environmental parameters and other spatial factors such as land use patterns and pollution sources in various areas. The use of GIS could facilitate visualization and further analysis of groundwater quality distribution, enabling policymakers to take more targeted actions based on specific locations affected by contamination. Additionally, this hybrid approach can be applied in other predictive contexts, such as air pollution monitoring or assessing the impact of climate change on water resources. Optimizing model architecture through techniques such as hyperparameter fine-tuning and advanced regularization methods could further enhance accuracy and reduce the risk of overfitting.

ACKNOWLEDGMENTS

The author would like to express heartfelt gratitude to the Directorate of Research, Technology, and Community Service (DRTPM) of DIKTI for the research funding support that enabled the completion of this study. The author also extends appreciation to the Doctoral Program in Mathematics at Universitas Brawijaya for the facilities, academic support, and guidance provided throughout the preparation of this journal. May the contributions and support offered bring significant benefits to the advancement of science in the future.

REFERENCES

- [1] Al-Aizari, H.S., Aslaou, F., Al-Aizari, A.R., Al-Odayni, A.B., Al-Aizari, A.J.M. (2023). Evaluation of groundwater quality and contamination using the groundwater pollution index (GPI), nitrate pollution index (NPI), and GIS. *Water*, 15(20): 3701. <https://doi.org/10.3390/w1520370>
- [2] Burri, N.M., Weatherl, R., Moeck, C., Schirmer, M. (2019). A review of threats to groundwater quality in the anthropocene. *Science of the Total Environment*, 684: 136-154. <https://doi.org/10.1016/j.scitotenv.2019.05.236>
- [3] Custodio, M., Cuadrado, W., Peñaloza, R., Montalvo, R., Ochoa, S., Quispe, J. (2020). Human risk from exposure to heavy metals and arsenic in water from rivers with mining influence in the Central Andes of Peru. *Water*, 12(7): 1946. <https://doi.org/10.3390/w12071946>
- [4] El Mountassir, O., Bahir, M., Ouazar, D., Chehbouni, A., Carreira, P. M. (2022). Temporal and spatial assessment of groundwater contamination with nitrate using nitrate pollution index (NPI), groundwater pollution index (GPI), and GIS (case study: Essaouira basin, Morocco). *Environmental Science and Pollution Research*, 1-18. <https://doi.org/10.1007/s11356-021-16922-8>
- [5] Christodoulou, E., Ma, J., Collins, G.S., Steyerberg, E.W., Verbakel, J.Y., Van Calster, B. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *Journal of Clinical Epidemiology*, 110: 12-22. <https://doi.org/10.1016/j.jclinepi.2019.02.004>
- [6] Latchmore, T., Hynds, P., Brown, R.S., Schuster-Wallace, C., Dickson-Anderson, S., McDermott, K., Majury, A. (2020). Analysis of a large spatiotemporal groundwater quality dataset, Ontario 2010–2017: Informing human health risk assessment and testing guidance for private drinking water wells. *Science of the Total Environment*, 738: 140382. <https://doi.org/10.1016/j.scitotenv.2020.140382>
- [7] Fullerton, A.S. (2009). A conceptual framework for ordered logistic regression models. *Sociological Methods & Research*, 38(2): 306-347. <https://doi.org/10.1177/0049124109346162>
- [8] Fischetti, M., Jo, J. (2018). Deep neural networks and mixed integer linear optimization. *Constraints*, 23(3): 296-309. <https://doi.org/10.1007/s10601-018-9285-6>
- [9] Bengio, Y., Courville, A., Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8): 1798-1828. <https://doi.org/10.1109/TPAMI.2013.50>
- [10] Katz, G., Huang, D.A., Ibeling, D., Julian, K., Lazarus, C., Lim, R., Shah, P., Thakoor, S., Wu, H., Zeljić, A., Dill, D.L., Kochenderfer, M.J., Barrett, C. (2019). The marabou framework for verification and analysis of deep neural networks. In *Computer Aided Verification: 31st International Conference, CAV 2019, New York, USA*, pp. 443-452. https://doi.org/10.1007/978-3-030-25540-4_26
- [11] Azzaoui, H., Boukhamla, A.Z.E., Arroyo, D., Bensayah, A. (2022). Developing new deep-learning model to enhance network intrusion classification. *Evolving Systems*, 13(1): 17-25. <https://doi.org/10.1007/s12530-020-09364-z>
- [12] Badr, B.E., Altawil, I., Almomani, M., Al-Saadi, M., Alkhurainej, M. (2023). Fault diagnosis of three-phase induction motors using convolutional neural networks. *Mathematical Modelling of Engineering Problems*, 10(5): 1727-1736. <https://doi.org/10.18280/mmep.100523>
- [13] Alrayes, F.S., Zakariah, M., Driss, M., Boulila, W. (2023). Deep Neural Decision Forest (DNDF): A novel approach for enhancing intrusion detection systems in network traffic analysis. *Sensors*, 23(20): 8362. <https://doi.org/10.3390/s23208362>
- [14] Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.R. (2021). Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3): 247-278. <https://doi.org/10.1109/JPROC.2021.3060483>
- [15] Aslam, N., Khan, I.U., Albahussain, T.I., Almousa, N.F., Alolayan, M.O., Almousa, S.A., Alwhebi, M.E. (2022). MEDeep: A deep learning based model for memotion analysis. *Mathematical Modelling of Engineering Problems*, 9(2): 533-538. <https://doi.org/10.18280/mmep.090232>
- [16] Veluswamy, N., Boopathy, J. (2024). A tree-structured deep learning model for improving classification with self-adaption and self-learning. *Mathematical Modelling of Engineering Problems*, 11(10): 2801-2808. <https://doi.org/10.18280/mmep.111022>
- [17] Chen, Q.J., Xie, Y.X., Ao, Y., Li, T.G., Chen, G.R., Ren, S.F., Wang, C., Li, S.F. (2021). A deep neural network inverse solution to recover pre-crash impact data of car collisions. *Transportation Research Part C: Emerging Technologies*, 126: 103009. <https://doi.org/10.1016/j.trc.2021.103009>
- [18] Afzaal, H., Farooque, A.A., Abbas, F., Acharya, B., Esau, T. (2019). Groundwater estimation from major physical hydrology components using artificial neural networks and deep learning. *Water*, 12(1): 5. <https://doi.org/10.3390/w12010005>
- [19] Müller, J., Park, J., Sahu, R., Varadharajan, C., Arora, B., Faybishenko, B., Agarwal, D. (2021). Surrogate optimization of deep neural networks for groundwater predictions. *Journal of Global Optimization*, 81: 203-231. <https://doi.org/10.1007/s10898-020-00912-0>
- [20] Liang, J., Bi, G., Zhan, C. (2020). Multinomial and ordinal Logistic regression analyses with multi-categorical variables using R. *Annals of Translational Medicine*, 8(16): 982. <https://doi.org/10.21037/atm-2020-57>
- [21] Tutz, G. (2022). Ordinal regression: A review and a taxonomy of models. *Wiley Interdisciplinary Reviews: Computational Statistics*, 14(2): e1545.

- <https://doi.org/10.1002/wics.1545>
- [22] Bender, R., Grouven, U. (1998). Using binary logistic regression models for ordinal data with non-proportional odds. *Journal of Clinical Epidemiology*, 51(10): 809-816. [https://doi.org/10.1016/S0895-4356\(98\)00066-3](https://doi.org/10.1016/S0895-4356(98)00066-3)
- [23] Fagerland, M.W., Hosmer, D.W. (2016). Tests for goodness of fit in ordinal logistic regression models. *Journal of Statistical Computation and Simulation*, 86(17): 3398-3418. <https://doi.org/10.1080/00949655.2016.1156682>
- [24] Pramoddyo, H., Ngabu, W., Riza, S., Iriany, A. (2024). Spatial analysis using geographically weighted ordinary logistic regression (GWOLR) method for prediction of particle-size fraction in soil surface. *IOP Conference Series: Earth and Environmental Science*, 1299(1): 012005. <https://doi.org/10.1088/1755-1315/1299/1/012005>
- [25] Cichy, R.M., Kaiser, D. (2019). Deep neural networks as scientific models. *Trends in Cognitive Sciences*, 23(4): 305-317. <https://doi.org/10.1016/j.tics.2019.01.009>
- [26] Peng, Y., Gong, D., Deng, C., Li, H., Cai, H., Zhang, H. (2022). An automatic hyperparameter optimization DNN model for precipitation prediction. *Applied Intelligence*, 52(3): 2703-2719. <https://doi.org/10.1007/s10489-021-02507-y>
- [27] Cuong-Le, T., Minh, H.L., Sang-To, T., Khatir, S., Mirjalili, S., Wahab, M.A. (2022). A novel version of grey wolf optimizer based on a balance function and its application for hyperparameters optimization in deep neural network (DNN) for structural damage identification. *Engineering Failure Analysis*, 142: 106829. <https://doi.org/10.1016/j.engfailanal.2022.106829>
- [28] Tsai, C.W., Fang, Z.Y. (2021). An effective hyperparameter optimization algorithm for DNN to predict passengers at a metro station. *ACM Transactions on Internet Technology (TOIT)*, 21(2): 1-24. <https://doi.org/10.1145/3410156>
- [29] Alkhouly, A.A., Mohammed, A., Hefny, H.A. (2021). Improving the performance of deep neural networks using two proposed activation functions. *IEEE Access*, 9: 82249-82271. <https://doi.org/10.1109/ACCESS.2021.3085855>
- [30] Choi, D., Cho, H., Rhee, W. (2018). On the difficulty of DNN hyperparameter optimization using learning curve prediction. In *TENCON 2018-2018 IEEE Region 10 Conference*, pp. 651-656. <https://doi.org/10.1109/TENCON.2018.8650070>
- [31] Chen, Y., Xie, Y., Song, L., Chen, F., Tang, T. (2020). A survey of accelerator architectures for deep neural networks. *Engineering*, 6(3): 264-274. <https://doi.org/10.1016/j.eng.2020.01.007>
- [32] Hussain, H., Tamizharasan, P.S., Rahul, C.S. (2022). Design possibilities and challenges of DNN models: A review on the perspective of end devices. *Artificial Intelligence Review*, 1-59. <https://doi.org/10.1007/s10462-022-10138-z>
- [33] Abdou, M.A. (2022). Literature review: Efficient deep neural networks techniques for medical image analysis. *Neural Computing and Applications*, 34(8): 5791-5812. <https://doi.org/10.1007/s00521-022-06960-9>
- [34] Kepner, J., Gadepally, V., Jananthan, H., Milechin, L., Samsi, S. (2018). Sparse deep neural network exact solutions. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, Waltham, MA, USA, pp. 1-8. <https://doi.org/10.1109/HPEC.2018.8547742>
- [35] Chung, H., Lee, S.J., Park, J.G. (2016). Deep neural network using trainable activation functions. In *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, pp. 348-352. <https://doi.org/10.1109/IJCNN.2016.7727219>
- [36] Zhang, C., Woodland, P.C. (2016). DNN speaker adaptation using parameterised sigmoid and ReLU hidden activation functions. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5300-5304. <https://doi.org/10.1109/ICASSP.2016.7472689>
- [37] Al-Dulaimi, A., Zabihi, S., Asif, A., Mohammadi, A. (2019). A multimodal and hybrid deep neural network model for remaining useful life estimation. *Computers in Industry*, 108: 186-196. <https://doi.org/10.1016/j.compind.2019.02.004>
- [38] Jogin, M., Madhulika, M.S., Divya, G.D., Meghana, R.K., Apoorva, S. (2018). Feature extraction using convolution neural networks (CNN) and deep learning. In *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, pp. 2319-2323. <https://doi.org/10.1109/RTEICT42901.2018.9012507>
- [39] Wang, H., Ma, P., Yuan, Y., Liu, Z., Wang, S., Tang, Q., Niu, S., Wu, S. (2022). Enhancing DNN-based binary code function search with low-cost equivalence checking. *IEEE Transactions on Software Engineering*, 49(1): 226-250. <https://doi.org/10.1109/TSE.2022.3149240>
- [40] Yacoub, R., Axman, D. (2020). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP systems*, pp. 79-91. <https://doi.org/10.18653/v1/2020.eval4nlp-1.9>
- [41] Miao, J., Zhu, W. (2022). Precision-recall curve (PRC) classification trees. *Evolutionary Intelligence*, 15(3): 1545-1569. <https://doi.org/10.1007/s12065-021-00565-2>
- [42] Venkataraman, K., Uddameri, V. (2012). Modeling simultaneous exceedance of drinking-water standards of arsenic and nitrate in the Southern Ogallala aquifer using multinomial logistic regression. *Journal of Hydrology*, 458: 16-27. <https://doi.org/10.1016/j.jhydrol.2012.06.028>
- [43] Müller, J., Park, J., Sahu, R., Varadharajan, C., Arora, B., Faybishenko, B., Agarwal, D. (2021). Surrogate optimization of deep neural networks for groundwater predictions. *Journal of Global Optimization*, 81: 203-231. <https://doi.org/10.1007/s10898-020-00912-0>
- [44] Alabdulkreem, E., Alruwais, N., Mahgoub, H., Dutta, A.K., Khalid, M., Marzouk, R., Motwakel, A., Drar, S. (2023). Sustainable groundwater management using stacked LSTM with deep neural network. *Urban Climate*, 49: 101469. <https://doi.org/10.1016/j.uclim.2023.101469>
- [45] Nhu, V.H., Phan, D.C., Hoa, P.V., Vinh, P.T., Bui, D.T. (2024). A new approach based on deep neural networks and multisource geospatial data for spatial prediction of groundwater spring potential. *IEEE Access*, 12: 26344-26363. <https://doi.org/10.1109/ACCESS.2024.3360337>