

Detecting water contaminants using an electrochemical sensor based on improved Transformer Encoder with Genetic Algorithm

Rizqy Ahsana Putri ^{a, ID}, Riyanarto Sarno ^{a, ID, *}, Wahyu Prasetyo Utomo ^{b, ID}, Fadlilatul Taufany ^{c, ID, *}, Dwi Sunaryono ^{a, ID}, Shoffi Izza Sabilla ^{d, ID}, Sang-Seok Lee ^{e, ID}, Kelly Rossa Sungkono ^{a, ID}, Taufiq Choirul Amri ^{a, ID}, Ruri Agung Wahyuono ^{f, ID}, Rini Handayani ^{a, g}, Mohamad Almas Prakasa ^{h, ID}

^a Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia

^b Department of Chemistry, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia

^c Department of Chemical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia

^d Department of Medical Technology, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia

^e Graduate School of Engineering, Tottori University, Tottori, 680-8552, Japan

^f Department of Physics Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia

^g School of Applied Science, Telkom University, Bandung, 40257, Indonesia

^h Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia

ARTICLE INFO

Editor: Guangming Jiang

Keywords:

Electrochemical sensor
Genetic algorithm
TOPSIS
Transformer encoder
Water pollution

ABSTRACT

Detecting water contaminants is important to mitigate environmental and health problems. Currently, traditional techniques for detecting water contaminants are costly and time-consuming. Thus, the development of a cost-effective, rapid, and AI-based alternative is needed. This paper proposes a novel integrated and systematic framework for detecting water contaminants from electrochemical sensor data using an improved Transformer Encoder optimized with a Genetic Algorithm (TE-GA). The electrochemical sensor is employed to capture voltammetric signals from water contaminant samples. Then, the Transformer Encoder processes these signals to capture patterns for identifying contaminants. A Genetic Algorithm (GA) is applied to optimize the hyperparameters of the Transformer Encoder. Furthermore, this study compares the effectiveness of GA in improving Transformer Encoder performance with other optimization methods, including Grid Search, Random Search, Particle Swarm Optimization, Bayesian Optimization, and Tree-structured Parzen Estimator. The Technique for Order Preference by Similarity to Ideal Solution ranking confirmed that the TE-GA achieved the optimal balance, delivering the highest accuracy (99.17%) while completing the optimization process in 965.78 s, thereby reducing the computation time by up to 57% compared to other computationally heavy methods.

1. Introduction

Heavy metal contamination, particularly lead and cadmium, poses significant environmental and health risks, necessitating effective monitoring [1]. Standard analytical techniques, such as Atomic Absorption Spectroscopy (AAS) and Inductively Coupled Plasma Mass Spectrometry (ICP-MS), remain the predominant methods for heavy metal detection [2]. However, these methods are limited by high operational costs (approx. 33 USD per test), long processing times (up to 7 days), and the need for expert operators [3–6]. Thus, the use of sensor-driven methods is increasing as a promising alternative to conventional techniques, such as AAS and ICP-MS, offering faster, more affordable, and more

accessible solutions that can be applied by non-experts [7,8]. Among the various sensing technologies, electrochemical sensors have emerged as a particularly potential modality for heavy metal detection.

An electrochemical sensor is a device that converts chemical information into an electrical signal for analytical purposes. In an electrochemical sensor, the input involves a liquid sample containing ions that interact with the surface of the working electrode [9]. When a potential is applied, these ions participate in oxidation or reduction reactions at the electrode interface. The process is controlled by voltammetry techniques, which systematically vary the applied potential to analyze the redox behavior of the ions. The resulting output is a voltammetric

* Corresponding authors.

E-mail addresses: riyanarto@its.ac.id (R. Sarno), f_taufany@chem-eng.its.ac.id (F. Taufany).

<https://doi.org/10.1016/j.jwpe.2026.110111>

Received 18 November 2025; Received in revised form 11 April 2026; Accepted 18 April 2026

Available online 23 April 2026

2214-7144/© 2026 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

signal, commonly referred to as a voltammogram, which is a continuous signal representing current as a function of potential [10]. This signal contains complex patterns that can be used to identify various heavy metals present in water samples [11]. However, interpreting complex signals such as voltammograms is particularly difficult and requires expertise [12]. As an alternative that simplifies the interpretation process, Artificial Intelligence (AI) techniques, including Machine Learning (ML) and Deep Learning (DL), have been applied to voltammogram data for heavy-metal detection [13,14].

There are various approaches that have been applied to interpret voltammograms. ML has been used to support voltammogram interpretation, particularly in classifying types of water contaminants [13, 15], while DL has been utilized for more advanced analytical approaches [14,16,17]. These developments provided the initial motivation for this study to employ a Transformer Encoder, as part of DL. The Transformer Encoder provides an advantage in capturing intricate relationships and interactions among multiple features in the dataset, supporting effective classification [18]. Despite its advantages, Transformer Encoder has several hyperparameters that must be tuned to achieve optimal performance.

At present, hyperparameter selection is often performed manually by trial and error or by applying conventional optimization methods such as grid search [13]. These approaches are time-consuming and computationally expensive, whereas the metaheuristic algorithm like Genetic Algorithm (GA) offers an efficient alternative by exploring the hyperparameter space through evolutionary search [19]. GA allows the model to reach near-optimal configurations through an efficient evolutionary search [20].

Additionally, a comparative evaluation among several hyperparameter optimization techniques is needed to validate that the GA provides superior performance in terms of both accuracy and execution time. Models that achieve high accuracy may require longer computation times [21,22], which makes it challenging to select the model that offers the optimal balance between accuracy and execution time. This condition highlights the need for a systematic Multi-Criteria Decision-Making (MCDM) approach, such as the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS).

Based on the reviewed literature, three critical research gaps have been identified. First, regarding the sensor-model integration gap, there is a lack of comprehensive frameworks that effectively couple low-cost electrochemical sensors with automated ML/DL interpretation. Existing studies often focus on sensor development while relying on manual data analysis. Second, regarding the data availability gap, the absence of a voltammogram-based dataset for contaminant detection hinders the development and validation of reliable detection models. Third, regarding the model optimization and trade-off gap, while Transformer Encoders demonstrate superior capability in capturing complex feature interactions, their performance is heavily dependent on hyperparameter configuration. Moreover, existing studies often focus solely on maximizing accuracy, neglecting the trade-off with computation time which is critical for rapid deployment. Addressing these three distinct gaps, this research proposes the following contributions:

1. Proposing an integrated and systematic framework that leverages electrochemical sensors, ML/DL models, hyperparameter optimization techniques, and MCDM analysis for detecting water contaminants.
2. Acquiring and utilizing a primary dataset of raw voltammograms representing three distinct classes: Lead (Pb) and Cadmium (Cd) within the 50–250 ppm range, and non-heavy metal baselines [23].
3. Proposing an Improved Transformer Encoder optimized with GA (TE-GA) architecture, where the Transformer Encoder is used for contaminant classification and the GA is used for hyperparameter optimization. Then, the optimization is evaluated using TOPSIS, enabling balanced selection based on both model accuracy and execution time.

This paper is organized as follows: Section 2 reviews related work on electrochemical sensors for water monitoring, the application of ML/DL models to voltammogram analysis. Section 3 presents the proposed method, including data acquisition, data preprocessing, development of TE-GA, model evaluation, and model ranking using TOPSIS. Section 4 presents the results, compares different classifiers, and discusses optimization and TOPSIS outcomes. Section 5 concludes the main findings and highlights future research opportunities.

2. Related works

This section reviews previous work on electrochemical water monitoring sensors, the application of ML/DL models to process voltammograms, and studies on Transformer Encoder architectures and GA across various domains. This section also discusses research that incorporates decision-making methods for model evaluation.

2.1. Electrochemical sensors for detecting water contaminants

Electrochemical sensors offer several advantages in environmental monitoring, including high sensitivity, low cost, simple fabrication, and rapid response. Electrochemical sensors can be used to detect contaminants, including organic contaminants (such as bacteria [24]) and inorganic contaminants (such as heavy metals like Pb [25,26] and Cd [27,28]). Advanced approaches have focused on the simultaneous detection of multiple heavy metals. For instance, previous studies have utilized electrochemical sensors to detect both Pb and Cd in water samples [29–31]. Other work has also developed systems capable of simultaneously detecting Pb and mercury (Hg) [32].

Although electrochemical sensors have demonstrated strong sensitivity and selectivity performance, a common limitation lies in the data processing stage. The mentioned studies rely on manual or semi-automated interpretation. This limitation highlights the need for a more robust approach, which can be addressed by integrating ML/DL in electrochemical sensor.

2.2. ML/DL for the processing voltammogram

ML/DL methods are increasingly being applied to voltammogram processing because of their potential to automate data analysis. For instance, established ML algorithms such as Support Vector Machine (SVM) [13] and Random Forest (RF) [15], have been applied to classify voltammograms and have reported notable accuracy in distinguishing different heavy-metal ions. For DL approaches, more advanced architectures have been applied to capture the complex and nonlinear patterns present in voltammograms. Gated Recurrent Units (GRU) [16] have been employed for voltammogram classification, while Convolutional Neural Networks (CNN) [17] have been used to capture local signal patterns. Furthermore, one-dimensional CNNs (1D-CNN) have been applied for the simultaneous detection of multiple heavy metals, such as Cd^{2+} , Pb^{2+} , Cu^{2+} , and Hg^{2+} , in water samples [14].

Previous studies have utilized deeper architectures and explore comparisons across multiple algorithms. However, the adoption of more advanced DL methods, such as attention-based mechanisms, has been limited.

2.3. Transformer encoder for processing complex data

The Transformer Encoder has emerged as a powerful architecture for processing complex and high-dimensional data. Its effectiveness is due to the self-attention mechanism, typically implemented as multi-head attention, which allows the model to dynamically weigh the importance of all features in a sequence, thereby effectively capturing intricate relationships and interactions within the data [18,33].

Transformer Encoders have been applied across various research domains and data types, including time series data [34], images [35],

text [36], and log data [37]. Several studies have implemented Transformer Encoders for signal-based datasets, such as Electrocardiogram (ECG) signals [38], Electroencephalogram (EEG) signals [39], Ultrasonic Spectrograms [40], and Gas Sensor Signals [41]. One study also processed voltammogram data using a transformer-based model known as the Fused Convolutional Transformer (FCvT), which was designed to address the challenge of overlapping signal peaks, although the use of a Transformer Encoder was not explicitly stated [42].

The evaluation results from previous studies indicate that the Transformer Encoder achieves strong performance in various applications, particularly in signal processing tasks. However, research on applying hyperparameter optimization techniques to Transformer Encoder architectures is still limited.

2.4. GA for hyperparameter optimization

Hyperparameter optimization plays a critical role in enhancing the performance of ML/DL models. Parameters such as learning rate, batch size, number of hidden layers, and attention units can significantly influence both model accuracy and generalization ability. Among various optimization strategies, GA has been widely adopted due to its ability to effectively explore complex parameter spaces through evolutionary operations such as selection, crossover, and mutation.

Several studies have demonstrated the ability of GA to improve model performance across various architectures and domains. For instance, GA has been used to optimize an EfficientNetB7 and DenseNet201 ensemble for ultrasound image classification, resulting in up to 99.95% accuracy improvements [43]. Similarly, a study on bankruptcy forecasting applied GA to optimize XGBoost and Artificial Neural Networks, achieving the highest accuracy of 98.3% and outperforming PSO [44]. GA has also been applied to optimize a transformer-based YOLO model for dental anomaly detection, resulting in improved classification accuracy of up to 99.31% [20]. In the domain of fluid mechanics, a hybrid approach integrating SVR with PSO and GA was utilized to predict discharge coefficients, achieving a high determination coefficient (R^2) of up to 0.9732 [45]. These studies demonstrate the utility of GA in improving model performance across diverse domains and architectures. This consistent performance highlights the potential of applying GA to improve the Transformer Encoder for detecting water contaminants using an electrochemical sensor.

2.5. MCDM approaches for model evaluation

In most ML/DL studies for prediction or classification, model evaluation commonly relies on accuracy as the main performance metric. However, execution time is also a critical factor, especially in real-time monitoring applications. A highly accurate model that requires long execution time may not be practical or efficient [21,22]. Ideally, a good model should achieve high accuracy while minimizing execution time [46].

Some studies have considered computational time alongside predictive performance, but the evaluation of these metrics has been done separately, without seeking a balance between the two [47–50]. To address this issue, researchers have applied MCDM approaches to evaluate the balance between multiple performance metrics. MCDM methods, such as MARCOS, TOPSIS, EDAS, CODAS, ELECTRE III, PROMETHEE II, FDOSM, and 2TLN-FDOSM, can identify the most suitable model by considering a comprehensive set of evaluation criteria, including training time, testing time, AUC, classification accuracy, F1 score, precision, recall, LogLoss, and specificity [51–53]. These methods support model selection based on predictive accuracy and overall efficiency, including execution time and other relevant performance indicators.

Previous studies have applied various MCDM techniques, primarily to evaluate and select ML/DL algorithms. In contrast, the application of MCDM for assessing hyperparameter optimization strategies remains limited, which highlights a gap explored in this research.

3. Proposed method

This section presents the detailed framework of detecting water contaminants using an electrochemical sensor based on TE-GA approach. The methodological framework begins with data acquisition, followed by data preprocessing to prepare the voltammetric signals for analysis. Then, the Transformer Encoder model is developed and further enhanced using GA to improve its predictive performance. Then, the models are evaluated using several performance metrics. A model ranking using the TOPSIS method is performed to identify the most effective optimization method by evaluating and balancing both accuracy and execution time. Fig. 1 illustrates the proposed framework, outlining each step from data acquisition to model evaluation and final performance ranking.

3.1. Data acquisition

This section presents the voltammetric data acquisition process using an electrochemical sensor system designed to classify water samples containing heavy metals. The data collection procedure comprises four key stages: solution preparation, electrochemical setup, voltammetric measurement, and signal acquisition.

3.1.1. Solution preparation

First, the supporting electrolyte was prepared by dissolving KCl in distilled water to obtain a 0.5 M KCl solution. This specific concentration was selected to provide optimal ionic conductivity and stabilize the background current, ensuring that the redox peaks of heavy metals are clearly distinguishable and well-defined [54]. During the preparation and homogenization process, the solutions were stirred using a magnetic stirrer at 300 rpm at a controlled room temperature $25 \pm 2^\circ\text{C}$ [14].

Heavy metal solutions were prepared by dissolving analytical grade (PA) Cadmium Nitrate and Lead Nitrate into the electrolyte at concentrations varying from 50 to 250 ppm. This wide range of concentrations was intentionally selected to ensure the robustness of the ML/DL models, ensuring they generalize well across different contamination levels rather than being optimized for a single value. For classification purposes, all concentration levels for a specific metal were pooled into a single representative class. This pooling strategy was adopted to enhance the model's practical deployment capability, focusing on the identification of the metal type regardless of its specific concentration. For non-heavy metal samples, only the electrolyte solution without metal salts was used.

3.1.2. Electrochemical setup

Following the solution preparation, an electrochemical system was assembled. The primary component of this system was an electrochemical sensor utilizing three electrodes: a Glassy Carbon Electrode (GCE) as the working electrode, an Ag/AgCl reference electrode (RE), and a Platinum wire counter electrode (CE) [55]. Before starting each measurement, all electrodes were cleaned using an electrode cleaner solution, rinsed thoroughly with deionized water, and dried with chemical wipes to ensure surface reproducibility and remove any passivation layers. The electrodes were then immersed in a beaker containing the test solution and fixed in a stationary holder. This rigid setup ensured a constant inter-electrode distance and prevented mechanical vibrations or short-circuiting during the recording process. The electrodes were connected to a potentiostat that interfaced with a computer to enable real-time signal recording. Fig. 2 provides a visual overview of the electrochemical setup.

3.1.3. Voltammetric measurement

During voltammetric measurement, the potential applied to the working electrode was varied within a defined range, and the resulting current response was recorded. The applied potential ranged from -2.0 V to $+2.0$ V using a scan rate of 0.1 V/s [56] and a step size of 0.01 V. These parameters were selected to effectively capture the redox behavior of the investigated substances.

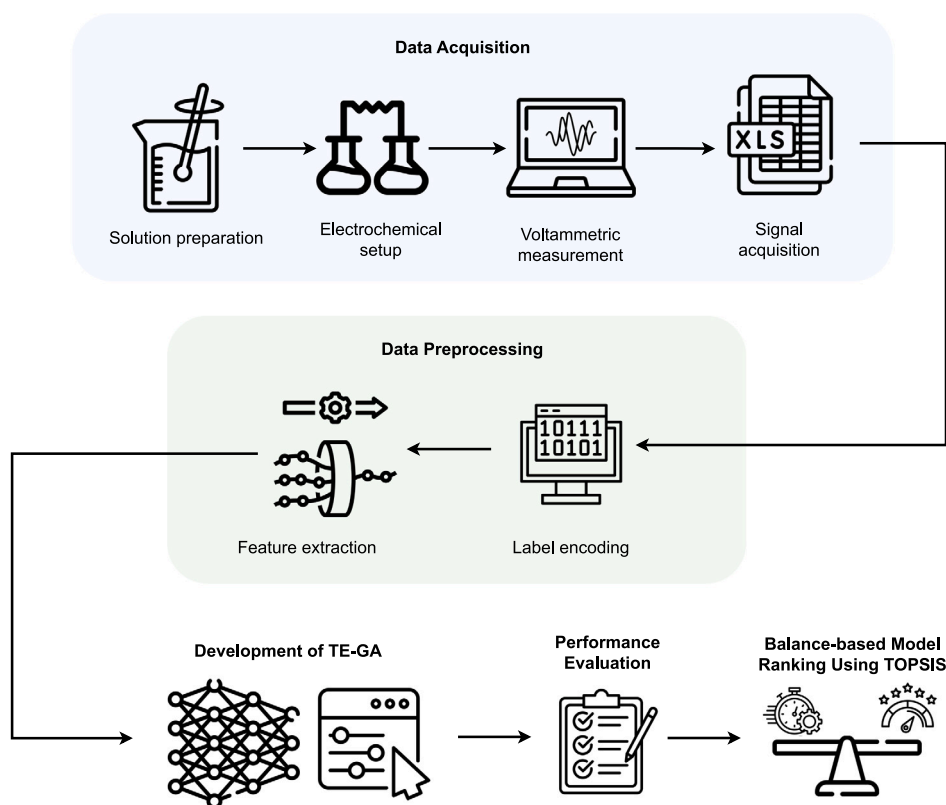


Fig. 1. Overall framework for detecting water contaminants using an electrochemical sensor based on the TE-GA.

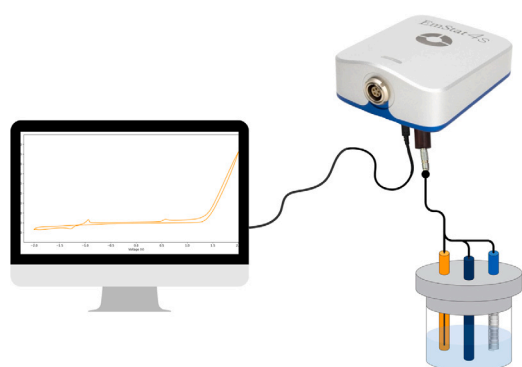


Fig. 2. The electrochemical setup.

3.1.4. Signal acquisition

Each voltammetric test generated one output file in Excel format. The recorded data included a graphical signal known as a voltammogram and two numerical columns: voltage (V) and current (μA). Each concentration variant underwent 45 replicate measurements. These raw signals served as the primary foundation for subsequent data preprocessing and the development of the classification models.

3.2. Data preprocessing

The preprocessing stage began with label encoding, where one-hot encoding was used to convert the class labels (Pb, Cd, and Non-Heavy Metal) into a numerical format. Label encoding is necessary to prepare the data for classification algorithms that require numerical input [57].

The next step is feature extraction. Feature extraction was specifically applied to the SVM and DNN models, which require clear feature

representations to perform classification effectively. Therefore, a statistical approach was employed to condense the voltammograms into a set of representative values while retaining their essential electrochemical characteristics. The raw signal consists of two columns: voltage (V) and current (μA). From these, six statistical features were calculated: mean, standard deviation, skewness, kurtosis, maximum, and minimum [58].

Higher-order time-frequency features (e.g., Fast Fourier Transform components, wavelet coefficients) and explicit peak-based features (e.g., peak area, peak slope) were deliberately excluded from this extraction process. Because the primary objective of this study is qualitative classification rather than quantitative regression, implementing complex peak integration or frequency-domain transformations would introduce unnecessary computational overhead.

In contrast, the Transformer Encoder does not require this manual feature extraction step. Its architecture, driven by the self-attention mechanism, is designed to learn complex dependencies directly from the raw sequential data. Applying statistical reduction to the Transformer would be redundant and could potentially lead to information loss, limiting the model's ability to extract fine-grained patterns from the full signal waveform.

3.3. Development of TE-GA

TE-GA is a method that combines the power of a Transformer Encoder for data processing with a Genetic Algorithm (GA) for optimizing hyperparameters. This approach is designed to automatically find the best combination of hyperparameters, which improves the performance of the Transformer Encoder. The Transformer Encoder is a deep learning architecture designed to process data with multiple features, while the GA is a search method inspired by biological evolution that helps identify optimal solutions within a large search space [59]. The Transformer Encoder architecture is constructed with a fixed depth of 2 encoder layers. The embedding dimension (equivalent to the `key_dims`) is optimized within a range of [16, 32]. To mitigate

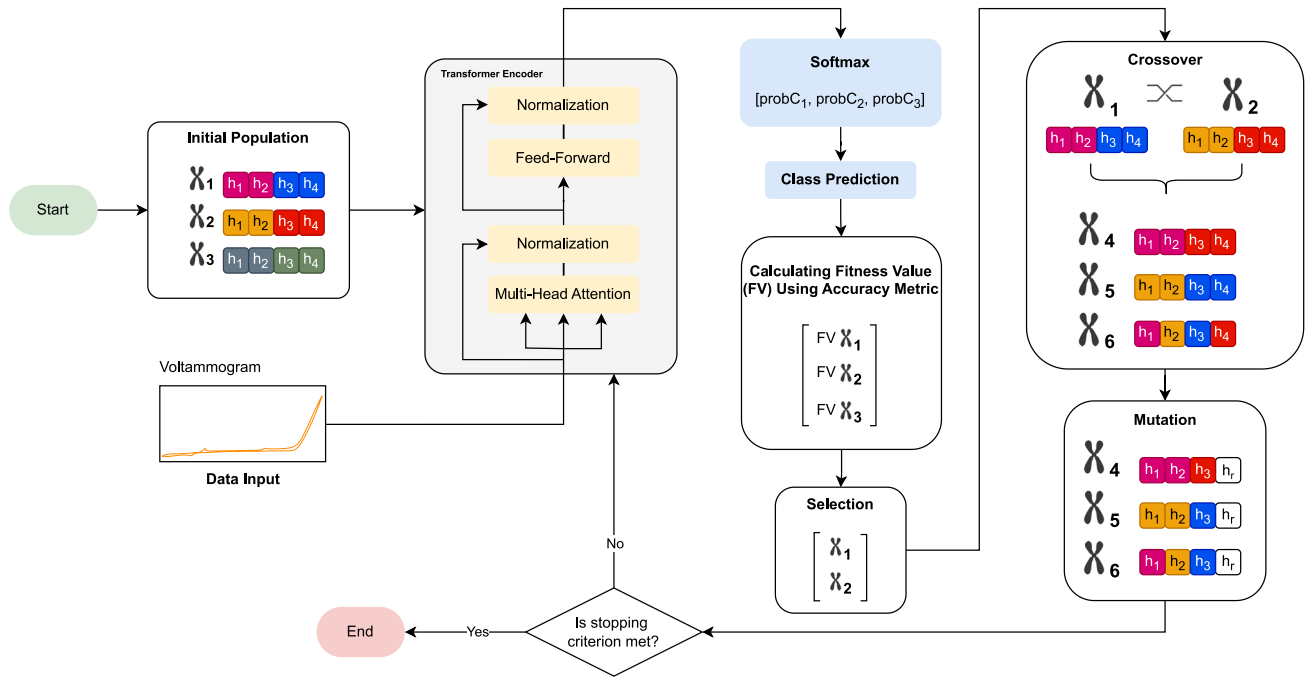


Fig. 3. Architecture of TE-GA.

Table 1
Summary of fixed and optimized hyperparameters for the Transformer Encoder.

Hyperparameter	Status	Value
Number of Encoder Layers	Fixed	2
Dropout Rate	Fixed	0.3
Optimizer	Fixed	Adam
Initial Learning Rate	Fixed	0.001
Embedding Dimension (key_dim)	Optimized	[16, 32]
Number of Heads (num_heads)	Optimized	[2, 4, 6]
Dense Units (dense_units)	Optimized	[64, 128]
Batch Size (batch_size)	Optimized	[16, 32, 64]

overfitting, a dropout rate of 0.3 is applied following the dense feed-forward network. During training, the model is optimized using the Adam optimizer with an initial learning rate of 0.001. These fixed parameters were selected based on optimal settings established in prior studies [60,61]. A comprehensive summary of the architectural configuration is provided in Table 1. The architecture of the TE-GA method is illustrated in Fig. 3. The TE-GA workflow begins with the initialization of the population. In GA, the population represents a collection of individuals, where each individual corresponds to a unique set of hyperparameters for the Transformer Encoder. This individual, often referred to as a chromosome, is composed of several genes, with each gene representing a single hyperparameter. In this method, four hyperparameters are optimized: $[h_1, h_2, h_3, h_4]$ with h_1 : number of attention heads, h_2 : embedding dimension size, h_3 : number of dense units, and h_4 : batch size. These values are distinct for each individual and directly influence the operational performance of the Transformer Encoder.

Following initialization, each individual in the population is evaluated. This process involves training the Transformer Encoder model using the hyperparameters specified by the individual. Prior to entering the attention mechanism, the raw voltammogram sequences undergo a specific formatting pipeline to ensure compatibility with the Transformer Encoder architecture. First, truncation is applied based on the voltage range, selecting only the data points between -2.0 V and -0.5 V. This specific range is retained because it contains the most critical electrochemical signal characteristics, while the remaining data

is discarded to reduce noise. Subsequently, zero-padding is applied to the tail of the sequences to ensure a uniform input length across all samples. To stabilize the learning process, a Batch Normalization layer is applied to the input features before they are processed by the encoder.

Once prepared, the voltammogram data is fed into the Multi-Head Attention layer. This layer enables the model to simultaneously attend to different parts of the input sequence. Each head in this mechanism independently captures specific relationships or patterns within the data, allowing the model to learn diverse features from multiple perspectives. The Multi-Head Self-Attention mechanism is mathematically defined in Eqs. (1) and (2) [34].

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (1)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

where Q , K , and V denote the query, key, and value matrices, respectively. QW_i^Q , KW_i^K , and VW_i^V are learnable projection matrices that transform the input into multiple subspaces. The attention function computes the scaled dot-product attention between the projected queries and keys, which is then used to weight the corresponding values. The outputs of all attention heads are concatenated along the channel dimension and passed through a final linear projection using W^O .

Subsequently, the resulting output is normalized and passed through a feed-forward network consisting of dense layers with ReLU activation. The output of the Transformer Encoder is then fed into a Softmax activation function. This function aggregates the extracted sequence features into a suitable classification format, producing a probability distribution for each class, e.g., $[\text{prob}_{C_1}, \text{prob}_{C_2}, \text{prob}_{C_3}]$. Based on these probabilities, class prediction is performed by selecting the class with the highest probability. For example, if the probabilities for a given data point are $[0.2, 0.75, 0.05]$, the predicted class is C_2 as it has the highest probability (0.75). Then, this predicted class is compared with the actual class to determine the accuracy. To ensure the model's generalization capability and prevent overfitting, this accuracy – which serves as the fitness value (GA Objective) for each individual – is computed strictly on the validation folds of each cross-validation split,

not the training folds. Thus, the objective function of the GA is to maximize this validation accuracy.

After individuals have been evaluated, those with the best fitness values are chosen for the selection phase. This ensures that only the top-performing individuals have the opportunity to reproduce. The selected individuals are combined through crossover, where the genes of two parent individuals are mixed to produce new offspring. The number of offspring is typically equal to the initial population size. Next, mutation is applied to each new offspring. Mutation introduces random changes to individual genes, such as altering a specific hyperparameter value, to explore new solutions and prevent the population from converging on a suboptimal solution. For example, a gene h_4 might be mutated to h_r by adding a random value. The newly mutated individuals are then used to configure the Transformer encoder again, and the entire process is repeated. This iterative cycle continues until a stopping criterion is met. In this research, the stopping criteria are a maximum number of iterations or the achievement of 100% accuracy on the validation folds. Furthermore, to avoid bias and computational waste, an early stopping mechanism is employed; the optimization process terminates if the validation accuracy does not improve for a defined number of consecutive generations.

3.4. Performance evaluation

Performance evaluation is a crucial step in assessing the effectiveness of ML/DL models. In this study, stratified k-fold cross-validation with $k = 5$ is employed to ensure a balanced assessment across different subsets of data and to mitigate the risks of overfitting and underfitting [62]. This stratification guarantees that each fold maintains a consistent distribution of classes and concentration levels, ensuring a fair evaluation with no data leakage between training and testing folds. To rigorously verify the final model's generalization capability on completely unseen data, a separate hold-out testing set is also utilized. Several evaluation metrics, including accuracy, precision, recall, and F1-score, are then used to provide a comprehensive analysis of model performance [63,64]. Beyond these scalar metrics, the analysis is enriched by visualizing the Confusion Matrix and Receiver Operating Characteristic-Area Under Curve (ROC-AUC) metrics. [64].

In the context of hyperparameter evaluation, the execution time is also recorded to assess the computational efficiency of each model in addition to these classification metrics. Execution time refers to the total duration (measured in seconds) required for a model to complete both the training and testing phases. To ensure reproducibility and a fair comparison of computational efficiency, all experiments were conducted on the Google Colab Pro platform. The environment utilized an Intel Xeon Processor, 12 GB of RAM, and an NVIDIA Tesla T4 GPU (16 GB VRAM). The implementation was carried out on an Ubuntu operating system using Python 3.10 with PyTorch and Scikit-learn frameworks.

3.5. Model ranking using TOPSIS

This study employs TOPSIS to rank several hyperparameter optimization methods based on their balanced performance across critical metrics, including predictive accuracy and execution time. TOPSIS was selected over other MCDM methods due to its computational simplicity and ability to provide a clear ranking based on the relative distance of each alternative from an ideal solution, without requiring the comparison of every method against each other.

TOPSIS operates by calculating the relative distance of each alternative from the ideal and negative-ideal solutions. The ideal solution comprises the best achievable value for each criterion, while the negative-ideal solution consists of the worst. The process begins with constructing a decision matrix $X = [x_{ij}]$, where each element x_{ij} denotes the rating of alternative A_i with respect to criterion C_j (with $i = 1, 2, \dots, m$ for m alternatives and $j = 1, 2, \dots, n$ for n criteria).

This matrix is then normalized using Min-Max normalization for each column (criterion) using Eq. (3).

$$r_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \quad (3)$$

where $\min(x_j)$ and $\max(x_j)$ represent the minimum and maximum values of criterion j across all alternatives, respectively. This normalization transforms all criteria values to a common scale $[0, 1]$, making them comparable despite differing original ranges.

The normalized matrix (r_{ij}) is subsequently weighted by multiplying each normalized value with its corresponding criterion weight, as shown in Eq. (4). In this step, v_{ij} represents the weighted normalized value of alternative A_i under criterion C_j , where the weight (w_j) is derived from the original weight (W_j) assigned to each criterion.

$$v_{ij} = w_j \times r_{ij} \text{ where } w_j = \frac{W_j}{\sum_{k=1}^n W_k} \text{ and } \sum_{j=1}^n w_j = 1 \quad (4)$$

Here, W_j denotes the original weight value of criterion C_j before normalization, while w_j represents the normalized weight ensuring that the total sum of all weights equals one. The term W_k represents the total sum of all original criterion weights used for normalization. Next, the Positive Ideal Solution (PIS), denoted as A^+ , and the Negative Ideal Solution (NIS), denoted as A^- , are determined by identifying the maximum and minimum values for each criterion across all alternatives, as defined in Eqs. (5) and (6).

$$A^+ = \{v_1^+, v_2^+, \dots, v_n^+\}, \quad v_j^+ = \max(v_{ij}) \quad (5)$$

$$A^- = \{v_1^-, v_2^-, \dots, v_n^-\}, \quad v_j^- = \min(v_{ij}) \quad (6)$$

The Euclidean distance from each alternative to both the Positive Ideal Solution (PIS) d_i^+ and the Negative Ideal Solution (NIS) d_i^- is calculated using Eqs. (7) and (8). In these equations, v_j^+ corresponds to the maximum weighted normalized value of criterion j across all alternatives, while v_j^- corresponds to the minimum value.

$$d_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \quad (7)$$

$$d_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad (8)$$

These distance measurements enable the computation of the TOPSIS score through Eq. (9), which represents the relative closeness of each alternative to the ideal solution. Finally, all alternatives are ranked based on their C_i values, where scores closer to 1 indicate more preferred options.

$$C_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad (9)$$

This study employs TOPSIS to evaluate and rank hyperparameter optimization methods by treating each method as an alternative assessed on two key criteria: accuracy, which reflects model performance, and execution time, which indicates computational efficiency. The TOPSIS framework enables a balanced evaluation that considers both criteria simultaneously, providing a ranking that is not based solely on accuracy.

4. Results and discussions

This section presents the classification results obtained using various classification methods. In addition, this section discusses the outcomes of hyperparameter optimization applied to the best-performing model from the default classification results. Additionally, this section provides a detailed explanation of the outcomes of the TOPSIS analysis.

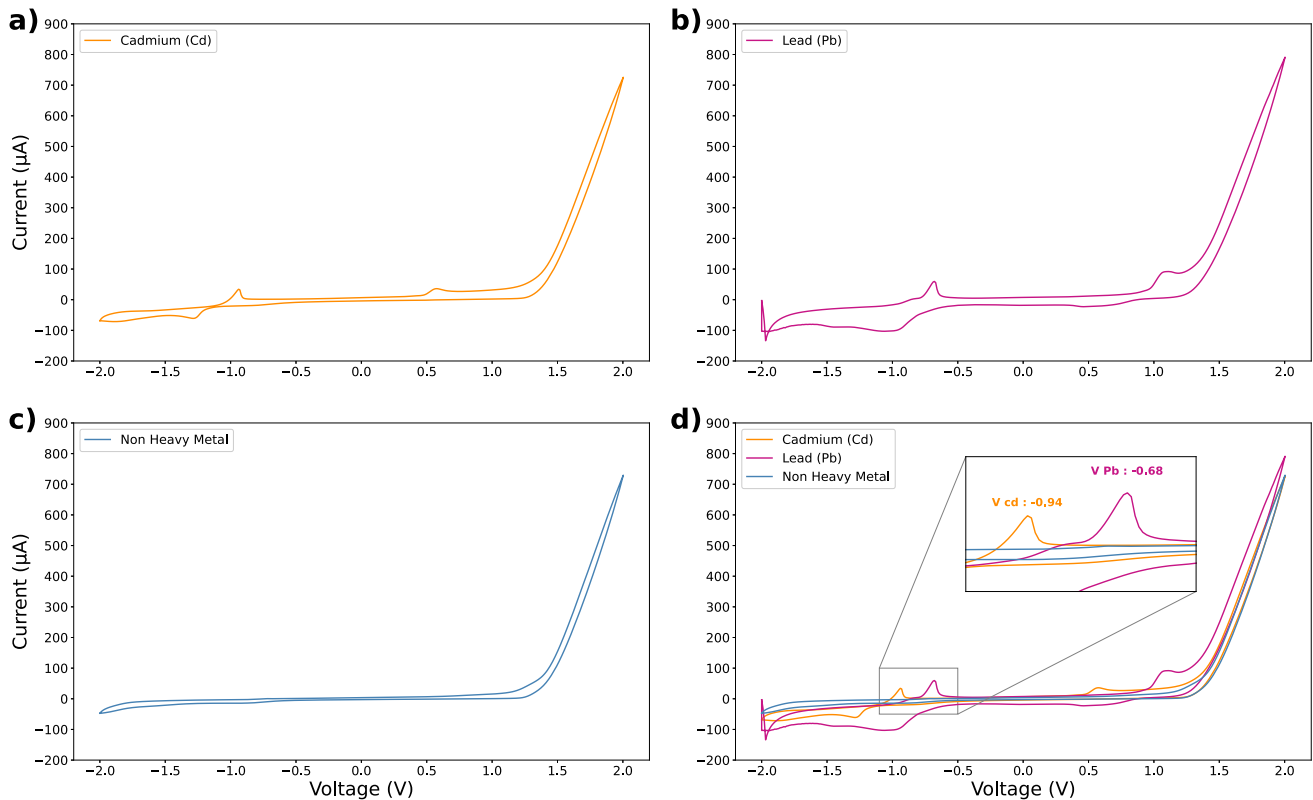


Fig. 4. Examples of voltammogram signals of (a) Cd, (b) Pb, (c) Non-Heavy Metal, and (d) Comparison of Cd, Pb, and Non-Heavy Metal.

Table 2

Preprocessed data for classification input.

Sample_ID	Features												Label
	V_{mean}	μA_{mean}	V_{std}	μA_{std}	$V_{skewness}$	$\mu A_{skewness}$	$V_{kurtosis}$	$\mu A_{kurtosis}$	V_{max}	μA_{max}	V_{min}	μA_{min}	
Pb_1	0.00	55.39	1.16	166.45	0.00	2.87	-1.20	7.39	2.00	794.78	-2.00	-74.26	Pb
Pb_2	0.00	55.40	1.16	164.52	0.00	2.83	-1.20	7.17	2.00	780.09	-2.00	74.26	
...	...	...	...	...	...	...	...	...	...	...	...	...	
Pb_{224}	0.00	60.64	1.16	228.06	0.00	2.04	-1.20	3.60	2.00	895.08	-2.00	-198.36	
Pb_{225}	0.00	49.96	1.16	217.38	0.00	1.96	-1.20	3.48	2.00	852.32	-2.00	-217.25	Cd
Cd_1	0.00	55.39	1.16	166.45	0.00	2.87	-1.20	7.39	2.00	794.78	-2.00	-74.26	
Cd_2	0.00	55.40	1.16	164.52	0.00	2.83	-1.20	7.17	2.00	780.09	-2.00	-72.06	
...	...	...	...	...	...	...	...	...	...	...	...	...	
Cd_{224}	0.00	50.96	1.16	167.41	0.00	2.65	-1.20	6.36	2.00	766.82	-2.00	-96.68	Non-Heavy Metal
Cd_{225}	0.00	48.22	1.16	158.02	0.00	2.61	-1.20	6.22	2.00	721.31	-2.00	-80.82	
Non_1	0.00	68.19	1.16	229.55	0.00	2.08	-1.20	3.45	2.00	894.03	-2.00	-173.99	
Non_2	0.00	52.29	1.16	151.65	0.00	2.84	-1.20	7.22	2.00	723.21	-2.00	-47.91	
...	...	...	...	...	...	...	...	...	...	...	...	...	Non-Heavy Metal
Non_{224}	0.00	53.82	1.16	156.00	0.00	2.85	-1.20	7.27	2.00	744.82	-2.0	-46.39	
Non_{225}	0.00	53.61	1.16	155.31	0.00	2.85	-1.20	7.26	2.00	741.60	-2.00	-46.55	

4.1. Data description

The data conducted from this study consists of 675 excel files for three sample types: Pb, Cd, and non-heavy metal, with 225 files for each type. This comprises 45 replicates per concentration level, where 5 replicates from each concentration were initially set aside for independent hold-out testing. Each excel file contains two columns with the variables Voltage (V) and Current (μA). Each variable has 800 rows of data. Representative voltammograms for each sample type are presented in Fig. 4, with Cd in Fig. 4(a), Pb in Fig. 4(b), non-heavy metal in Fig. 4(c), and an overlay of all three in Fig. 4(d) to facilitate direct comparison. The voltammograms from non-heavy-metal samples displayed an almost linear response, whereas the Cd and Pb signals exhibited distinct peaks. The heavy-metal species can

be further distinguished by their characteristic peak potentials: in the zoomed region of Fig. 4(d), Cd exhibits a peak at approximately -0.94 V, while Pb peaks near -0.68 V. Although the peak positions may vary slightly across individual experiments, the Cd peaks typically fall within -0.9 to -1.0 V, while Pb peaks generally occur in the -0.6 to -0.7 V range.

After data acquisition, the preprocessing phase was performed to prepare the raw voltammogram signals for ML/DL applications. For the SVM and DNN models, feature extraction was applied to generate 12 feature columns from each data: V_{mean} , μA_{mean} , V_{std} , μA_{std} , $V_{skewness}$, $\mu A_{skewness}$, $V_{kurtosis}$, $\mu A_{kurtosis}$, V_{max} , μA_{max} , V_{min} , and μA_{min} . Table 2 presents the preprocessed data results. The preprocessed data were then used to train the models, with the class label serving as the target variable for sample type classification.

Table 3
Classification results.

Model	Fold	Accuracy (%)	Precision (%)	Recall (%)	F-1 Score (%)	Avg. Accuracy (%)	Avg. Precision (%)	Avg. Recall (%)	Avg. F1-Score (%)
SVM	1	94.17	94.74	94.17	94.14	92.50 ± 1.18	93.38 ± 1.02	92.50 ± 1.18	92.46 ± 1.14
	2	90.83	91.97	90.83	90.87				
	3	93.33	94.22	93.33	93.20				
	4	91.67	92.54	91.67	91.67				
	5	92.50	93.43	92.50	92.44				
DNN	1	99.17	99.19	99.17	99.17	95.17 ± 3.14	95.43 ± 2.94	95.17 ± 3.14	95.18 ± 3.12
	2	94.17	94.59	94.17	94.18				
	3	95.00	95.07	95.00	95.00				
	4	97.50	97.67	97.50	97.50				
	5	90.00	90.62	90.00	90.04				
Transformer Encoder	1	100.00	100.00	100.00	100.00	98.00 ± 2.08	98.20 ± 1.76	98.00 ± 2.08	98.01 ± 2.07
	2	97.50	97.60	97.50	97.50				
	3	99.17	99.19	99.17	99.17				
	4	99.17	99.19	99.17	99.17				
	5	94.17	95.04	94.17	94.19				

4.2. Classification results

The dataset prepared in the previous stage was then processed using classification techniques employing several ML/DL methods: SVM, Deep Neural Network (DNN), and Transformer Encoder. Table 3 presents the classification results for SVM, DNN, and Transformer Encoder models using stratified 5-fold cross-validation. These results were obtained without any hyperparameter optimization. The results demonstrate that the Transformer Encoder achieved the highest performance across all metrics, with an average accuracy of 98.00%. This superior performance can be attributed to its ability to capture complex relationships through its attention mechanism, effectively modeling intricate dependencies between features and resulting in robust generalization across folds.

Following the Transformer Encoder's top performance, the DNN also showed strong results with an average accuracy of 95.17%. However, the Transformer Encoder consistently performed better, achieving an accuracy 2.83 percentage points higher. This indicates that while the DNN is a powerful model for capturing nonlinear patterns, its predefined hierarchical architecture might be less adaptable than the Transformer's attention mechanism in identifying the specific complex relationships within this dataset.

The superior performance of the Transformer Encoder is clearly demonstrated when contrasted with the SVM model. The Transformer Encoder achieved 98.00% accuracy compared to SVM's 92.50%, resulting in a difference of 5.50 percentage points. The lower performance of SVM may be attributed to its reliance on kernel transformations, which are less flexible than the learning capacity of DNN and Transformer Encoder. Additionally, the fixed kernel parameters in the default settings could limit the ability of SVM to capture the complexities of the dataset, resulting in reduced accuracy and consistency across folds.

Given these results, the Transformer Encoder is clearly the best-performing model even without any hyperparameter tuning based on the accuracy metric. It is important to note that its superior accuracy was consistent with its precision, recall, and F1-score, indicating robust and balanced performance across all evaluation criteria. Based on this comprehensive evaluation, the Transformer Encoder was selected to proceed to the hyperparameter optimization stage to further enhance its performance.

4.3. Hyperparameter optimization results

In this study, the hyperparameter optimization process focused on the Transformer Encoder (TE) model, as it demonstrated the best performance during the classification stage. This study employs six hyperparameter optimization methods: GA, Grid Search (GS), Random

Search (RS), Particle Swarm Optimization (PSO), Bayesian Optimization (BO), and Tree-structured Parzen Estimator (TPE). Each method is evaluated based on its classification accuracy, ROC-AUC score, and execution time. The optimization process evaluated several key hyperparameters of the Transformer Encoder model, specifically testing the following candidate values: num_heads [2, 4, 6], key_dims [16, 32], dense_units [64, 128], and batch_size [16, 32, 64].

The evaluation was conducted using two experimental configurations. In configuration 1, GS, RS, BO, and TPE were allocated 10 maximum trials, while GA and PSO were configured with an initial population size of 10 and a maximum of 10 iterations. Configuration 2 expanded these parameters, with GS, RS, BO, and TPE given 20 maximum trials, while GA and PSO used an initial population size of 20 and a maximum of 20 iterations. Table 4 shows the hyperparameter optimization results using configuration 1 and Table 5 shows the results using configuration 2.

The hyperparameter optimization results show that all methods achieved high accuracy, exceeding 98% in both configurations, with most methods performing even better in configuration 2. This pattern is consistent with the ROC-AUC scores, which also exceeded 0.99 across all methods, and similarly improved in configuration 2. However, this accuracy improvement required substantially more computation, with execution times increasing by 11%–73% in configuration 2 compared to configuration 1 due to the expanded parameter settings. GA consistently achieved the highest accuracy in both configurations, with 99.00% in configuration 1 and 99.17% in configuration 2—an improvement of 0.17%. Its ROC-AUC scores followed the same trend, exceeding 0.99 across all folds. Despite this gain, GA remained the most computationally efficient method, with the smallest increase in execution time (11.5%) compared to other methods that experienced increases of up to 73%. RS, BO, and TPE achieved comparable average accuracies of around 98.84% in configuration 1. However, among these three, only BO showed improvement in configuration 2, increasing by 0.16% to reach 99.00%, while RS and TPE remained relatively constant. GS and PSO also improved, with GS increasing by approximately 0.16% and PSO by around 0.33% in configuration 2.

The confusion matrix visualization in Fig. 5 further confirms GA's superiority, showing that GA misclassified the fewest samples compared to other methods. Beyond the 5-fold cross-validation results, holdout testing, as presented in Table 6, further validated GA's performance with only one misclassification in configuration 1 and no errors in configuration 2—making it the only method to achieve perfect classification on the holdout set.

In terms of execution time, GA was the most efficient method, as shown in Table 7. In configuration 1, GA completed the optimization in just 866.13 seconds—reducing execution time by approximately 14% compared to TPE (1008.34 s). In configuration 2, GA again led with

Table 4
Hyperparameter optimization results with configuration 1.

Hyperparameter Optimization	Fold	Best Parameter	Best Accuracy (%)	Best ROC-AUC	Worst Parameter	Worst Accuracy (%)	Worst ROC-AUC	Avg. Best Accuracy (%)	Avg. Best ROC-AUC
TE-GS	1	[2, 32, 64, 16]	99.17	1.0000	[2, 16, 64, 16]	90.00	0.9985	98.67 ± 1.13	0.9956 ± 0.0059
	2	[2, 32, 128, 16]	96.67	0.9843	[4, 16, 128, 16]	89.17	0.9692		
	3	[2, 32, 128, 16]	100.00	1.0000	[4, 16, 128, 16]	91.67	0.9884		
	4	[4, 32, 64, 16]	98.33	0.9955	[2, 16, 128, 16]	90.83	0.9855		
	5	[2, 32, 128, 16]	99.17	0.9983	[2, 32, 64, 16]	84.17	0.9639		
TE-RS	1	[2, 32, 128, 16]	99.17	0.9992	[6, 32, 64, 64]	66.67	0.9169	98.84 ± 1.13	0.9961 ± 0.0059
	2	[2, 32, 128, 16]	96.67	0.9843	[6, 16, 64, 64]	63.33	0.8738		
	3	[2, 32, 128, 16]	100.00	1.0000	[6, 16, 64, 64]	67.50	0.9254		
	4	[6, 32, 64, 16]	99.17	0.9988	[6, 32, 64, 64]	64.17	0.9240		
	5	[2, 32, 128, 16]	99.17	0.9983	[6, 16, 64, 64]	64.17	0.9058		
TE-PSO	1	[4, 32, 128, 16]	99.17	0.9956	[2, 16, 64, 64]	60.83	0.9230	98.50 ± 1.22	0.9936 ± 0.0058
	2	[2, 32, 128, 16]	96.67	0.9843	[6, 32, 64, 32]	61.67	0.9216		
	3	[6, 32, 64, 16]	100.00	1.0000	[2, 16, 64, 64]	51.67	0.9011		
	4	[2, 32, 128, 16]	97.50	0.9898	[6, 32, 64, 32]	64.17	0.9318		
	5	[2, 32, 128, 16]	99.17	0.9983	[2, 32, 128, 32]	67.50	0.9133		
TE-BO	1	[2, 32, 128, 16]	99.17	0.9992	[2, 16, 64, 64]	60.83	0.9230	98.84 ± 1.13	0.9961 ± 0.0059
	2	[2, 32, 128, 16]	96.67	0.9843	[2, 16, 64, 64]	52.50	0.8795		
	3	[2, 32, 128, 16]	100.00	1.0000	[2, 16, 64, 64]	51.67	0.9011		
	4	[6, 32, 64, 16]	99.17	0.9988	[2, 16, 64, 64]	53.33	0.8931		
	5	[2, 32, 128, 16]	99.17	0.9983	[2, 16, 64, 64]	54.17	0.9065		
TE-TPE	1	[2, 32, 128, 16]	99.17	0.9992	[6, 32, 64, 64]	66.67	0.9169	98.84 ± 1.13	0.9961 ± 0.0059
	2	[2, 32, 128, 16]	96.67	0.9843	[2, 32, 64, 64]	61.67	0.9131		
	3	[2, 32, 128, 16]	100.00	1.0000	[2, 32, 64, 64]	69.17	0.9410		
	4	[6, 32, 64, 16]	99.17	0.9988	[6, 32, 64, 64]	64.17	0.9240		
	5	[2, 32, 128, 16]	99.17	0.9983	[2, 32, 64, 64]	65.00	0.9289		
TE-GA	1	[6, 32, 128, 16]	100.00	1.0000	[6, 32, 64, 64]	66.67	0.9169	99.00 ± 1.22	0.9966 ± 0.0062
	2	[2, 32, 128, 16]	96.67	0.9843	[6, 32, 64, 32]	61.67	0.9216		
	3	[6, 32, 64, 16]	100.00	1.0000	[6, 32, 128, 64]	75.00	0.9377		
	4	[6, 32, 64, 16]	99.17	0.9988	[6, 32, 64, 64]	64.17	0.9240		
	5	[6, 32, 64, 16]	99.17	0.9998	[2, 32, 64, 32]	64.17	0.9058		

Table 5
Hyperparameter optimization results with configuration 2.

Hyperparameter Optimization	Fold	Best Parameter	Best Accuracy (%)	Best ROC-AUC	Worst Parameter	Worst Accuracy (%)	Worst ROC-AUC	Avg. Best Accuracy (%)	Avg. Best ROC-AUC
TE-GS	1	[6, 32, 128, 16]	100.00	1.0000	[2, 32, 64, 32]	65.00	0.9509	98.83 ± 1.25	0.9956 ± 0.0059
	2	[2, 32, 128, 16]	96.67	0.9843	[2, 32, 64, 32]	60.83	0.9071		
	3	[2, 32, 128, 16]	100.00	1.0000	[6, 32, 64, 32]	65.00	0.9625		
	4	[4, 32, 64, 16]	98.33	0.9955	[2, 32, 64, 32]	64.17	0.9241		
	5	[2, 32, 128, 16]	99.17	0.9983	[2, 32, 64, 32]	65.00	0.9336		
TE-RS	1	[2, 32, 128, 16]	99.17	0.9992	[6, 32, 64, 64]	66.67	0.9169	98.84 ± 1.13	0.9961 ± 0.0059
	2	[2, 32, 128, 16]	96.67	0.9843	[6, 32, 64, 32]	61.67	0.9216		
	3	[2, 32, 128, 16]	100.00	1.0000	[6, 32, 64, 32]	65.00	0.9625		
	4	[6, 32, 64, 16]	99.17	0.9988	[6, 32, 64, 64]	64.17	0.9240		
	5	[2, 32, 128, 16]	99.17	0.9983	[6, 16, 64, 64]	64.17	0.9058		
TE-PSO	1	[6, 32, 128, 16]	100.00	1.0000	[2, 16, 64, 64]	60.83	0.9230	98.83 ± 1.55	0.9969 ± 0.0047
	2	[4, 32, 128, 32]	95.83	0.9876	[6, 32, 128, 32]	50.00	0.8439		
	3	[4, 32, 64, 16]	100.00	1.0000	[6, 32, 64, 32]	65.00	0.9625		
	4	[6, 32, 64, 16]	99.17	0.9988	[6, 32, 128, 32]	49.17	0.8579		
	5	[2, 32, 128, 16]	99.17	0.9983	[2, 16, 64, 64]	54.17	0.9065		
TE-BO	1	[6, 32, 128, 16]	100.00	1.0000	[2, 16, 64, 64]	60.83	0.9230	99.00 ± 1.22	0.9963 ± 0.0060
	2	[2, 32, 128, 16]	96.67	0.9843	[2, 16, 64, 64]	52.50	0.8795		
	3	[6, 32, 128, 16]	100.00	1.0000	[2, 16, 64, 64]	51.67	0.9011		
	4	[6, 32, 64, 16]	99.17	0.9988	[2, 16, 64, 64]	53.33	0.8931		
	5	[2, 32, 128, 16]	99.17	0.9983	[2, 16, 64, 64]	54.17	0.9065		
TE-TPE	1	[2, 32, 128, 16]	99.17	0.9992	[2, 16, 64, 64]	60.83	0.9230	98.84 ± 1.13	0.9961 ± 0.0059
	2	[2, 32, 128, 16]	96.67	0.9843	[2, 16, 64, 64]	52.50	0.8795		
	3	[2, 32, 128, 16]	100.00	1.0000	[2, 16, 64, 64]	51.67	0.9011		
	4	[6, 32, 64, 16]	99.17	0.9988	[2, 16, 64, 64]	53.33	0.8931		
	5	[2, 32, 128, 16]	99.17	0.9983	[2, 16, 64, 64]	54.17	0.9065		
TE-GA	1	[6, 32, 64, 16]	100.00	1.0000	[6, 32, 64, 64]	66.67	0.9169	99.17 ± 1.29	0.9992 ± 0.0015
	2	[6, 32, 128, 16]	96.67	0.9962	[6, 32, 64, 32]	61.67	0.9216		
	3	[6, 32, 64, 16]	100.00	1.0000	[2, 16, 64, 64]	51.67	0.9011		
	4	[6, 32, 64, 16]	100.00	1.0000	[6, 32, 64, 64]	64.17	0.9240		
	5	[6, 32, 64, 16]	99.17	0.9998	[6, 16, 64, 64]	64.17	0.9058		

Table 6

Performance comparison of hyperparameter optimization methods in Configuration 1 and Configuration 2 during holdout testing.

Hyperparameter Optimization	Configuration 1			Configuration 2		
	Correct Predict	False Predict	Accuracy (%)	Correct Predict	False Predict	Accuracy (%)
TE-GS	71	4	94.67	74	1	98.67
TE-RS	71	4	94.67	71	4	94.67
TE-PSO	70	5	93.33	74	1	98.67
TE-BO	71	4	94.67	74	1	98.67
TE-TPE	71	4	94.67	71	4	94.67
TE-GA	74	1	98.67	75	0	100.00

Table 7

Execution time comparison of hyperparameter optimization methods between Configuration 1 and Configuration 2.

HPO	Fold	Configuration 1		Configuration 2	
		Time (s)	Total Time (s)	Time (s)	Total Time (s)
TE-GS	1	245.71	1370.98	435.51	2175.33
	2	240.73		457.91	
	3	234.90		419.60	
	4	367.54		430.19	
	5	282.10		432.12	
TE-RS	1	245.18	1208.52	398.90	2088.47
	2	269.00		368.64	
	3	238.69		369.11	
	4	220.40		559.59	
	5	235.25		392.23	
TE-PSO	1	272.76	1469.23	635.78	2244.11
	2	256.51		388.09	
	3	402.79		448.55	
	4	244.71		425.08	
	5	292.46		346.61	
TE-BO	1	216.64	1130.22	331.52	1615.59
	2	219.98		340.81	
	3	225.46		290.58	
	4	265.65		286.46	
	5	202.49		366.22	
TE-TPE	1	231.93	1008.34	332.03	1676.76
	2	191.42		395.85	
	3	203.23		329.58	
	4	181.41		296.90	
	5	200.35		322.40	
TE-GA	1	102.28	866.13	147.91	965.78
	2	233.44		381.36	
	3	71.28		71.55	
	4	188.59		63.76	
	5	270.54		301.20	

965.78 s, requiring about 40% less time than BO (1615.59 s). This efficiency was even more pronounced against the most computationally expensive models: GA reduced the computation time by approximately 57% compared to PSO (2244.11 s), 56% compared to GS (2175.33 s), and 54% compared to RS (2088.47 s). This advantage can be attributed to its ability to efficiently explore the parameter space by evolving solution populations through crossover and mutation, which helps the algorithm find high-quality configurations much faster.

Expanding on the overall performance evaluation, a minor variation in accuracy was observed specifically in Fold 2. Rather than indicating a systemic limitation, this slight difference reflects a natural distribution shift inherent to the randomized data partitioning of k-fold cross-validation. Because the training subset in this particular fold inevitably presents a slightly different distribution compared to the broader dataset, a marginal variance in performance is an expected outcome. Ultimately, the model's ability to maintain an exceptionally high overall accuracy despite this normal distribution shift highlights its robustness and reliable generalization across diverse data subsets.

Another important finding is that there is no single best hyperparameter setting that works for all situations. The optimal combination changed depending on which data fold was used and which optimization method was applied. This happens because: (1) different data

distributions in each fold require different parameter settings, and (2) hyperparameters interact with each other such that the performance of one parameter is influenced by the specific values chosen for other parameters.

4.4. TOPSIS ranking results for balanced performance

After obtaining the performance evaluation results for each model and hyperparameter optimization method, this study uses TOPSIS to evaluate the balance between accuracy and execution time. For the baseline TOPSIS analysis, equal weights of 50:50 were assigned to both criteria. This equal weighting serves as the primary evaluation metric, underscoring the study's core objective of identifying methods that deliver an optimal balance between predictive performance and computational efficiency. However, acknowledging that stakeholder priorities may vary depending on specific application requirements, a sensitivity analysis was also conducted. This analysis evaluates the robustness of the rankings across four additional weight distribution scenarios: Scenario A (70:30) and Scenario B (80:20), which prioritize accuracy; and Scenario C (30:70) and Scenario D (20:80), which prioritize execution time.

The input for the TOPSIS consists of the average accuracy and total execution time for each method obtained from the hyperparameter optimization process. The methods are labeled A1 to A12, serving as alternatives in the TOPSIS process. Table 8 presents the TOPSIS ranking results.

As shown in Table 8, the TE-GA consistently ranked as the top two alternatives across almost all weight scenarios, which shows the efficiency of this method in balancing accuracy and execution time. Specifically, A12 achieved the highest ranking in the baseline 50:50 scenario, as well as in the 70:30, 80:20, and 30:70 configurations, because it successfully combined the highest overall accuracy (99.17%) with a highly efficient execution time (965.78 s). The sensitivity analysis reveals an interesting shift only in the extreme 20:80 scenario, where execution time is given extreme priority. Under this specific condition, A11 surpassed A12 to claim the first rank, as A11 required a shorter computation time of 866.13 s while still maintaining an excellent accuracy of 99.00%. The small gain in accuracy in A12 required slightly more time, giving A11 the edge only when speed is strictly prioritized.

In contrast, TE-GS and TE-PSO, along with TE-Random Search Configuration 2, consistently occupied the lowest rankings (8th to 12th) across most weight distributions. For instance, A6 and A2 suffered from extremely long execution times exceeding 2100 s, making them unsuitable for applications requiring computational efficiency. The sensitivity analysis shows these methods fluctuating at the bottom depending on the weights; for example, A1 and A2 improved slightly to the 9th and 11th ranks when accuracy was prioritized (70:30 and 80:20) but plummeted when time efficiency was factored in. This consistently low ranking occurs because GS evaluates every possible parameter combination in the search space, which guarantees completeness but incurs a severe computational cost. Similarly, PSO struggled to converge efficiently in this context, leading to prolonged execution without proportional accuracy gains, ultimately placing both methods at the bottom of the evaluation.

Other alternatives, specifically TE-BO, TE-TPE, and TE-RS Configuration 1, achieved middle-to-upper rankings (ranging from the 3rd

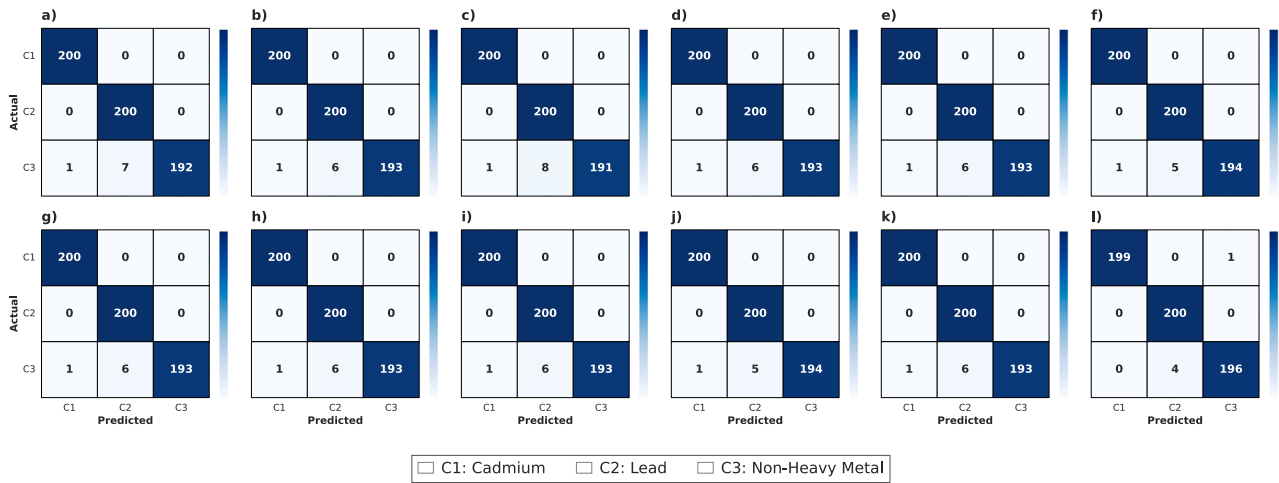


Fig. 5. Confusion matrices comparing the performance of six hyperparameter optimization methods across both configurations. Subfigures (a–f) show results for Configuration 1 using TE-GS, TE-RS, TE-PSO, TE-BO, TE-TPE, and TE-GA, while (g–l) present the corresponding results for Configuration 2.

Table 8

TOPSIS results under various weight scenarios.

ID Alternative	HPO	Configuration	Avg. Accuracy (%)	Total Time (s)	Rank (50:50)	Rank (70:30)	Rank (80:20)	Rank (30:70)	Rank (20:80)
A1	TE-GS	1	98.67	1370.98	8	11	11	6	6
A2	TE-GS	2	98.83	2175.33	11	9	9	11	11
A3	TE-RS	1	98.84	1208.52	5	6	6	5	5
A4	TE-RS	2	98.84	2088.47	10	8	8	10	10
A5	TE-PSO	1	98.50	1469.23	9	12	12	8	7
A6	TE-PSO	2	98.83	2244.11	12	10	10	12	12
A7	TE-BO	1	98.84	1130.22	4	5	5	4	4
A8	TE-BO	2	99.00	1615.59	6	3	3	7	8
A9	TE-TPE	1	98.84	1008.34	3	4	4	3	3
A10	TE-TPE	2	98.84	1676.76	7	7	7	9	9
A11	TE-GA	1	99.00	866.13	2	2	2	2	1
A12	TE-GA	2	99.17	965.78	1	1	1	1	2

Table 9

Comparison with previous studies.

Author	ML/DL Method	Hyperparameter Optimization	Best Accuracy	MCDM
[13]	SVM	Yes, not specified	98.31%	✗
[65]	SVM, RF, and KNN	Grid Search	95.6%	✗
[10]	LDA and SVM	No	100%	✗
[66]	SVM	Yes, not specified	93.5%	✗
This research	Transformer Encoder	GA	99.17%	✓

to 7th positions), demonstrating a moderate to good balance between accuracy and execution time. The sensitivity analysis highlights their dynamic nature; for instance, A8 jumped to the 3rd rank when accuracy was heavily weighted due to its strong 99.00% accuracy, but fell to the 8th rank when time was prioritized (20:80) due to its longer (1615.59 s) execution time. Conversely, A9 remained highly stable in the 3rd and 4th ranks across all scenarios, a stability largely attributed to its fast execution time of 1008.34 s. These middle-to-upper-ranked methods share a common characteristic: they employ intelligent optimization approaches, such as probabilistic models, to explore the parameter space. This exploration strategy makes them more efficient than exhaustive search methods like GS, but less systematic and optimally balanced than evolutionary algorithms like GA.

Overall, these results confirm that TE-GA is the most robust and superior method, dominating the top rankings even under rigorous sensitivity analysis. While A11 is preferable when execution speed is the primary concern, A12 remains the optimal choice for a genuinely balanced objective. By combining highly efficient execution time with the highest predictive accuracy, TE-GA delivers the best overall performance.

4.5. Comparison with previous research

The proposed method is compared with previous studies on the application of ML/DL to voltammetry data. The comparison focuses on the accuracy evaluation metric, the use of hyperparameter optimization, and the integration of MCDM to assess the balance between accuracy and execution time. The results are presented in Table 9. As shown in Table 9, the proposed research achieved an accuracy of 99.17%, which is higher than all other studies except the work of [10]. Although Ciursa [10] reported 100% accuracy, this result is unreliable because no cross-validation was performed, which creates a high risk of overfitting. In contrast, the proposed method used cross-validation to ensure reliable performance while still achieving very high accuracy.

The strong performance is attributed to the TE-GA architecture, which was developed by integrating a Transformer Encoder with a GA for hyperparameter optimization. Earlier studies generally relied on conventional models such as SVM, without providing detailed explanations of hyperparameter optimization. In addition, previous works focused solely on accuracy without considering execution time, whereas the proposed research incorporates MCDM to balance the trade-off between accuracy and execution time.

4.6. Limitations of the study

Despite the promising performance of the TE-GA model, this study has certain limitations that should be acknowledged. First, the experiments were conducted using controlled standard solutions containing individual contaminants (Pb and Cd) at a fixed concentration range (50–250 ppm). Consequently, the model's capability to generalize across concentration ranges beyond this tested scope, including limit-of-detection (LOD) analysis, remains unevaluated. Furthermore, the tests were performed without the interference of co-existing ions, mixed contaminants, or organic compounds typically present in real wastewater matrices. Therefore, this work explicitly serves as a proof-of-concept laboratory study, and the current results should not be extrapolated to field deployment without further extensive validation.

Additionally, the electrochemical parameters, such as electrolyte type, scan rate, and temperature, were kept constant to isolate the algorithmic performance. The impact of varying these conditions on voltammogram morphology – such as peak shifting or broadening – and the subsequent stability of the model remains a subject for future sensitivity analysis.

Finally, the feature extraction approach employed in this study may limit the model's broader generalizability. The current analysis prioritized the use of raw voltammetric signals and basic statistical parameters rather than higher-order time-frequency features (e.g., FFT, wavelets) or specific peak-based characteristics.

5. Conclusions

This study demonstrates the effectiveness of ML/DL models in classifying voltammetric signals for water contaminant detection. Among the evaluated models, the Transformer Encoder achieved the highest classification accuracy due to its ability to capture complex feature interactions, followed by DNN, while SVM showed the lowest performance. The hyperparameter optimization focused on the Transformer Encoder model, where the proposed TE-GA proved to be the most efficient method, achieving a high accuracy of 99.17% in only 965.78 s. In terms of execution time, TE-GA significantly outperformed other techniques, reducing the computation time by up to 57% compared to methods such as TE-GS, TE-RS, TE-PSO, TE-BO, and TE-TPE. A TOPSIS analysis confirmed TE-GA as the optimal alternative, providing the best balance between accuracy and execution time for practical monitoring applications.

Future work will focus on exploring advanced feature engineering, such as FFT or wavelet transforms, to further enhance the model's robustness against complex water matrices. Additionally, we aim to extend the current TE-GA architecture into real-time monitoring systems capable of detecting mixed contaminants in field-representative environments.

CRediT authorship contribution statement

Rizqy Ahsana Putri: Writing – original draft, Validation, Methodology, Investigation, Conceptualization. **Riyanarto Sarno:** Writing – review & editing, Supervision, Conceptualization. **Wahyu Prasetyo Utomo:** Validation, Supervision, Formal analysis. **Fadlilatul Taufany:** Writing – review & editing, Validation, Methodology. **Dwi Sunaryono:** Validation, Methodology, Funding acquisition. **Shoffi Izza Sabilla:** Writing – review & editing, Methodology. **Sang-Seok Lee:** Writing – review & editing, Supervision. **Kelly Rossa Sungkono:** Resources, Project administration, Funding acquisition. **Taufiq Choirul Amri:** Visualization, Investigation, Data curation. **Ruri Agung Wahyuono:** Software, Formal analysis. **Rini Handayani:** Investigation, Data curation. **Mohamad Almas Prakasa:** Writing – review & editing, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by the Directorate General of Higher Education, Research and Technology of the Republic of Indonesia under Penelitian Pendidikan Magister menuju Doktor untuk Sarjana Unggul (PMDSU) (Contract No. 038/E5/PG.02.00.PL/2024 and No. 1801/PKS/ITS/2024); the Indonesian Endowment Fund for Education (LPDP) on behalf of the Indonesian Ministry of Higher Education, Science and Technology and managed under the EQUITY Program (Contract No. 4299/B3/DT.03.08/2025 & No 3029/PKS/ITS/2025); the Institut Teknologi Sepuluh Nopember (ITS) managed under the Strategic Research Grant (SRG) Type D Scheme (Contract No. 1670/PKS/ITS/2026); and HETI under Penelitian Post-Doctoral.

Data availability

Data will be made available on request.

References

- [1] Y. Zeng, H. Wang, J. Hu, J. Zhang, F. Wang, T. Wang, Q. Zhou, R.A. Dahlgren, M. Gao, H. Gao, Z. Chen, Illuminated fulvic acid stimulates denitrification and as(III) immobilization in flooded paddy soils via an enhanced biophotoelectrochemical pathway, *Sci. Total Environ.* 912 (2024) 169670, <http://dx.doi.org/10.1016/j.scitotenv.2023.169670>.
- [2] N.S.A. Kassim, S.A.I.S.M. Ghazali, F.L. Bohari, N.A.Z. Abidin, Assessment of heavy metals in wastewater plant effluent and lake water by using atomic absorption spectrophotometry, *Mater Today Proc* 66 (2022) 3961–3964, <http://dx.doi.org/10.1016/j.matpr.2022.04.671>.
- [3] U. Airlangga, Standar biaya dan tarif layanan satuan usaha akademik dan satuan usaha komersial di lingkungan universitas airlangga tahun 2024 (cost standards and service rates for academic business units and commercial business units within airlangga university in 2024), 2025, URL https://sim.ditkeu.unair.ac.id/app/sbm/SBTL_UA2024.pdf. (Accessed: 08 September).
- [4] S. He, Y. Niu, L. Xing, Z. Liang, X. Song, M. Ding, W. Huang, Research progress of the detection and analysis methods of heavy metals in plants, *Front. Plant Sci.* Volume 15 - 2024 (2024) <http://dx.doi.org/10.3389/fpls.2024.1310328>.
- [5] M. Roulier, P.A. Baya, S. Roberge, D. Larivière, Comparison of radium-226 separation methods based on chromatographic and extraction resins for its determination by ICP-MS in drinking waters, *J. Mass Spectrom.* 59 (2) (2024) e5005, <http://dx.doi.org/10.1002/jms.5005>.
- [6] B.H. Abdelmonem, L.T. Kamal, R.M. Elbaz, M.R. Khalifa, A. Abdelnaser, From contamination to detection: The growing threat of heavy metals, *Heliyon* 11 (1) (2025) <http://dx.doi.org/10.1016/j.heliyon.2025.e41713>.
- [7] B. Du, X. Tian, Z. Chen, Y. Ge, C. Chen, H. Gao, Z. Liu, J. Tung, D. Fixler, S. Wei, S. Chen, H. Zhang, Ultrasensitive optoelectronic biosensor arrays based on twisted bilayer graphene superlattice, *Natl. Sci. Rev.* 12 (10) (2025) nwaf357, <http://dx.doi.org/10.1093/nsr/nwaf357>.
- [8] Z. Chen, J. Li, T. Li, T. Fan, C. Meng, C. Li, J. Kang, L. Chai, Y. Hao, Y. Tang, O.A. Al-Hartomy, S. Wageh, A.G. Al-Sehemi, Z. Luo, J. Yu, Y. Shao, D. Li, S. Feng, W.J. Liu, Y. He, X. Ma, Z. Xie, H. Zhang, A CRISPR/Cas12a-empowered surface plasmon resonance platform for rapid and specific diagnosis of the omicron variant of SARS-CoV-2, *Natl. Sci. Rev.* 9 (8) (2022) nwac104, <http://dx.doi.org/10.1093/nsr/nwac104>.
- [9] C.F. Ross, Considerations of the use of the electronic tongue in sensory science, *Curr Opin Food Sci* 40 (2021) 87–93, <http://dx.doi.org/10.1016/j.cofs.2021.01.011>.
- [10] P. Ciursa, M. Oroian, Voltammetric E-tongue for honey adulteration detection, *Sensors* 21 (15) (2021) 5059, <http://dx.doi.org/10.3390/s21155059>.
- [11] R.A. Putri, R. Sarno, W.P. Utomo, F. Taufany, K.R. Sungkono, T.C. Amri, A. Kamilah, R. Handayani, S.-S. Lee, A.M. Tjoa, A.A. Sagan, Cross-attention feature fusion network for robust estimation of Cd²⁺ and Pb²⁺ in water samples using cyclic voltammetry, *Chem. Eng. J. Adv.* 26 (2026) 101065, <http://dx.doi.org/10.1016/j.cej.2026.101065>.
- [12] Z. Zhang, B. Deng, Z. Cao, S. Zhang, J. Zhang, J. Zhang, D. Wei, P. Xiang, X. Liu, S. Xia, Deep learning-based transmembrane pulsed electrochemisorption for improved energy efficacy of recovering ammonia from wastewater, *Energy & Environ. Sustain.* 1 (3) (2025) 100029, <http://dx.doi.org/10.1016/j.eesus.2025.100029>.

- [13] J.X. Leon-Medina, D.A. Tibađuiza, J.C. Burgos, M. Cuenca, D. Vasquez, Classification of as, pb and cd heavy metal ions using square wave voltammetry, dimensionality reduction and machine learning, *IEEE Access* 10 (2022) 7684–7694.
- [14] S.A. Lahari, N. Kumawat, K. Amreen, R.N. Ponnalagu, S. Goel, IoT integrated and deep learning assisted electrochemical sensor for multiplexed heavy metal sensing in water samples, *NPJ Clean Water* 8 (1) (2025) 10, <http://dx.doi.org/10.1038/s41545-025-00441-x>.
- [15] R.A. Putri, R. Sarno, F. Taufany, S.-S. Lee, D. Sunaryono, S.I. Sabilla, K.R. Sungkono, T.C. Amri, R. Agung Wahyuono, Voltammetry data for aquatic pollutants detection: A comparative study of dimensionality reduction techniques, in: 2025 International Conference on Computer Sciences, Engineering, and Technology Innovation (ICoCSETI), 2025, pp. 329–333, <http://dx.doi.org/10.1109/ICoCSETI63724.2025.11019150>.
- [16] T.C. Amri, R. Sarno, D. Sunaryono, R.A. Putri, ADASYN-GRU approach for enhanced detection of heavy metals in water using electrochemical sensors, in: 2025 International Conference on Computer Sciences, Engineering, and Technology Innovation (ICoCSETI), IEEE, 2025, pp. 768–773, <http://dx.doi.org/10.1109/ICoCSETI63724.2025.11018884>.
- [17] Y. Tian, J. Liu, J. Qiao, F. Ge, Y. Yang, Q. Zhang, Advancements in electrochemical sensing technology for heavy metal ions detection, *Food Chem X* 25 (2025) 102204, <http://dx.doi.org/10.1016/j.fochx.2025.102204>.
- [18] R.A. Babu, V. V. Priya, M.K. Mishra, I.R. Raja, S.K. Chebrolu, B. Swarna, Transformer-based tabular foundation models: Outperforming traditional methods with tabpfn, *Int. J. Eng. Sci. Inf. Technol.* 5 (3) (2025) 448–455, <http://dx.doi.org/10.52088/ijesty.v5i3.1146>.
- [19] A. Azma, Y. Liu, M. Azma, M. Saadat, D. Zhang, J. Cho, S. Rezaia, Hybrid machine learning models for prediction of daily dissolved oxygen, *J. Water Process. Eng.* 54 (2023) 103957, <http://dx.doi.org/10.1016/j.jwpe.2023.103957>.
- [20] S.B. Singh, A. Laishram, K. Thongam, K.M. Singh, Automatic detection and classification of dental anomalies and tooth types using transformer-based yolo with GA optimization, *IEEE Access* 13 (2025) 59326–59338, <http://dx.doi.org/10.1109/ACCESS.2025.3556523>.
- [21] R. Yi, L. Tang, Y. Tian, J. Liu, Z. Wu, Identification and classification of pneumonia disease using a deep learning-based intelligent computational framework, *Neural Comput. Appl.* 35 (20) (2023) 14473–14486, <http://dx.doi.org/10.1007/s00521-021-06102-7>.
- [22] N. Kesav, M.G. Jibukumar, Efficient and low complex architecture for detection and classification of brain tumor using RCNN with two channel CNN, *J. King Saud Univ. - Comput. Inf. Sci.* 34 (8) (2022) 6229–6242, <http://dx.doi.org/10.1016/j.jksuci.2021.05.008>.
- [23] R.A. Putri, Voltammetric water contaminant dataset, 2026, <http://dx.doi.org/10.5281/zenodo.18825892>.
- [24] Y. Tang, Y. Li, P. Chen, S. Zhong, Y. Yang, Nucleic acid aptamer-based sensors for bacteria detection: a review, *BioEssays* 47 (3) (2025) e202400111, <http://dx.doi.org/10.1002/bies.202400111>.
- [25] N. Kumar, H. Tran, V.N. Shanov, N.T. Alvarez, Electrochemical detection of lead ions using carbon nanotube post electrodes, *Chem. Eng. J.* 499 (2024) 156550.
- [26] A. Arif, A.J. Acevedo-González, C.R. Cabrera, R.C. Roberts, Bismuth functionalized inkjet-printed electrochemical sensor for aqueous lead (II) detection, *IEEE Sens J* 25 (11) (2025) 18574–18583, <http://dx.doi.org/10.1109/JSEN.2025.3562804>.
- [27] F. Mashkoor, M. Shueb, S. Zhu, J. Ahmed, C. Jeong, Mechanistic insights into high-performance and selective cd (II) detection using ag2wo4 nanoparticle-based electrochemical sensors for real-world applications, *Mater Chem Phys* 335 (2025) 130519.
- [28] T. Ran, J. Zhou, Y. Xu, Z. Li, C. Chen, T. Wu, Electrochemical sensors based on screen-printed electrodes of cocnfs for the detection of Cd2+ in water sources, *J. Alloys Compd.* 1020 (2025) 179487, <http://dx.doi.org/10.1016/j.jallcom.2025.179487>.
- [29] S. Perveen, J.A. Baig, S.T. Hussain Sherazi, H.I. Afridi, E. Fricovsky, S. Sanam, K. Akhtar, Graphene oxide/magnesium ferrite nanocomposite modified electrochemical sensor for simultaneous detection of cadmium and lead, *Microchem. J.* 214 (2025) 114148, <http://dx.doi.org/10.1016/j.microc.2025.114148>.
- [30] J. You, J. Li, Z. Wang, M. Baghayeri, H. Zhang, Application of co3o4 nanocrystal/rGO for simultaneous electrochemical detection of cadmium and lead in environmental waters, *Chemosphere* 335 (2023) 139133, <http://dx.doi.org/10.1016/j.chemosphere.2023.139133>.
- [31] A. Scandurra, S. Mirabella, Square wave anodic stripping voltammetry applied to a nano-electrode for trace analysis of pb(II) and cd(II) ions in solution, *IEEE Sens J* 21 (20) (2021) 22134–22142, <http://dx.doi.org/10.1109/JSEN.2021.3051762>.
- [32] F. Gao, F. Zhan, S. Li, P. Antwi-Mensah, L. Niu, Q. Wang, Dual signal-based electrochemical aptasensor for simultaneous detection of lead(II) and mercury(II) in environmental water samples, *Biosens. Bioelectron.* 209 (2022) 114280, <http://dx.doi.org/10.1016/j.bios.2022.114280>.
- [33] A. Azma, Y. Liu, S. Rezaia, A hybrid WPD-SSA machine learning transformer for modeling nonlinear dynamics of flow systems, *J. Water Process. Eng.* 82 (2026) 109448, <http://dx.doi.org/10.1016/j.jwpe.2026.109448>.
- [34] A.T. Haryono, R. Sarno, K.R. Sungkono, Transformer-gated recurrent unit method for predicting stock price based on news sentiments and technical indicators, *IEEE Access* 11 (2023) 77132–77146, <http://dx.doi.org/10.1109/ACCESS.2023.3298445>.
- [35] M. Zhao, G. Cao, X. Huang, L. Yang, Hybrid transformer-CNN for real image denoising, *IEEE Signal Process. Lett.* 29 (2022) 1252–1256, <http://dx.doi.org/10.1109/LSP.2022.3176486>.
- [36] Y. Shin, Multi-encoder transformer for Korean abstractive text summarization, *IEEE Access* 11 (2023) 48768–48782, <http://dx.doi.org/10.1109/ACCESS.2023.3277754>.
- [37] S.S. Kusumawardani, S.A.I. Alfarozi, Transformer encoder model for sequential prediction of student performance based on their log activities, *IEEE Access* 11 (2023) 18960–18971, <http://dx.doi.org/10.1109/ACCESS.2023.3246122>.
- [38] M. Chen, Y. Li, L. Zhang, L. Liu, B. Han, W. Shi, S. Wei, Elimination of random mixed noise in ECG using convolutional denoising autoencoder with transformer encoder, *IEEE J. Biomed. Health Informatics* 28 (4) (2024) 1993–2004, <http://dx.doi.org/10.1109/JBHI.2024.3355960>.
- [39] A. Miltiadous, E. Gionanidis, K.D. Tzimourta, N. Giannakeas, A.T. Tzallas, DICE-net: A novel convolution-transformer architecture for alzheimer detection in EEG signals, *IEEE Access* 11 (2023) 71840–71858, <http://dx.doi.org/10.1109/ACCESS.2023.3294618>.
- [40] B. Ma, L. Chen, X. Sun, X. Yang, Q. Kong, MUST: Multi-channel ultrasonic spectrogram transformer for microdamage detection in metals, *Mech. Syst. Signal Process.* 231 (2025) 112680, <http://dx.doi.org/10.1016/j.msssp.2025.112680>.
- [41] F. Wu, R. Ma, Y. Li, F. Li, S. Duan, X. Peng, A novel electronic nose classification prediction method based on TETCN, *Sens Actuators B Chem* 405 (2024) 135272, <http://dx.doi.org/10.1016/j.snb.2024.135272>.
- [42] J. Liu, J. Li, Y. Xu, S.S. Low, H. Ning, S. Yu, D. Xu, Q. Liu, A fused convolutional transformer for voltammetric electronic tongue analysis tasks, *J. Environ. Chem. Eng.* 12 (5) (2024) 113462, <http://dx.doi.org/10.1016/j.jece.2024.113462>.
- [43] A. Ghosh, K. Srinivasan, EffiDenseGenOp: Ensemble transfer learning with hyperparameter tuning using genetic algorithm optimization for PCOS detection from ultrasound sonography images, *IEEE Access* 13 (2025) 54285–54312, <http://dx.doi.org/10.1109/ACCESS.2025.3553895>.
- [44] U.H. Ainan, L.Y. Por, Y.-L. Chen, J. Yang, C.S. Ku, Advancing bankruptcy forecasting with hybrid machine learning techniques: Insights from an unbalanced polish dataset, *IEEE Access* 12 (2024) 9369–9381, <http://dx.doi.org/10.1109/ACCESS.2024.3354173>.
- [45] A. Azma, A.G.L. Borthwick, R. Ahmadian, Y. Liu, D. Zhang, Modeling the discharge coefficient of labyrinth sluice gates using hybrid support vector regression and metaheuristic algorithms, *Phys. Fluids* 37 (4) (2025) 045117, <http://dx.doi.org/10.1063/5.0260738>.
- [46] A. Azma, Y. Liu, M. Eftekhari, D. Zhang, Comparison of hybrid deep learning models for estimation of the time-dependent scour depth downstream of river training structures, *Phys. Fluids* 36 (10) (2024) 101911, <http://dx.doi.org/10.1063/5.0231597>.
- [47] A. Sanga, V. Zsófia, Evaluation of machine learning algorithm's selection criteria for satellite image classification, leveraging accuracy and execution times, *Egypt. J. Remote. Sens. Space Sci.* 28 (3) (2025) 534–541, <http://dx.doi.org/10.1016/j.ejrs.2025.08.001>.
- [48] S. Panja, S. Mondal, A. Nag, J.P. Singh, M.J. Saikia, A.K. Barman, An efficient malware detection approach based on machine learning feature influence techniques for resource-constrained devices, *IEEE Access* 13 (2025) 12647–12665, <http://dx.doi.org/10.1109/ACCESS.2025.3526878>.
- [49] R. Aalikhani, M. Fathian, M.R. Rasouli, Comparative analysis of classification-based and regression-based predictive process monitoring models for accurate and time-efficient remaining time prediction, *IEEE Access* 12 (2024) 67063–67093, <http://dx.doi.org/10.1109/ACCESS.2024.3397185>.
- [50] R. Khelifi, B. Nini, M. Berkane, Enhanced classification system for real-time embedded vision applications, *IEEE Access* 12 (2024) 162311–162326, <http://dx.doi.org/10.1109/ACCESS.2024.3489476>.
- [51] A. Najafi, A. Nemati, M. Ashrafzadeh, S.H. Zolfani, Multiple-criteria decision making, feature selection, and deep learning: A golden triangle for heart disease identification, *Eng. Appl. Artif. Intell.* 125 (2023) 106662, <http://dx.doi.org/10.1016/j.engappai.2023.106662>.
- [52] M.E. Alqaysi, A.S. Albahri, R.A. Hamid, Evaluation and benchmarking of hybrid machine learning models for autism spectrum disorder diagnosis using a 2-tuple linguistic neutrosophic fuzzy sets-based decision-making model, *Neural Comput. Appl.* 36 (29) (2024) 18161–18200, <http://dx.doi.org/10.1007/s00521-024-09905-6>.
- [53] M.E. Alqaysi, A.S. Albahri, R.A. Hamid, Hybrid diagnosis models for autism patients based on medical and sociodemographic features using machine learning and multicriteria decision-making (MCDM) techniques: An evaluation and benchmarking framework, *Comput. Math. Methods Med.* 2022 (2022) 1–19, <http://dx.doi.org/10.1155/2022/9410222>.
- [54] S. Ghanbari, F. Ahour, S. Keshipour, An optical and electrochemical sensor based on l-arginine functionalized reduced graphene oxide, *Sci Rep* 12 (1) (2022) 19398, <http://dx.doi.org/10.1038/s41598-022-23949-5>.

- [55] S. Hu, S. Zhang, J. Qin, K. Cai, C. Peng, L. Luo, Y. Gu, Y. Mei, Simultaneous determination of lead and cadmium in water by metal oxide framework complex-modified glassy carbon electrodes, *Microchem. J.* 205 (2024) 111154, <http://dx.doi.org/10.1016/j.microc.2024.111154>.
- [56] O.T. Matica, A.G. Brotea, E.-M. Ungureanu, L.R. Mandoc, L. Birzan, Electrochemical and spectral studies of rhodanine in view of heavy metals determination, *Electrochem. Sci. Adv.* 3 (6) (2023) e2100218, <http://dx.doi.org/10.1002/elsa.202100218>.
- [57] A. Teguh, J. Slamet, H. Saputro, K. Sungkono, R. Sarno, Optimizing IT governance and project management in software development through AI integration and COBIT 2019 framework, in: 2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA), IEEE, 2024, pp. 1–6, <http://dx.doi.org/10.1109/ICTIIA61827.2024.10761914>.
- [58] M. Malikhah, R. Sarno, S. Sabilla, Ensemble learning for optimizing classification of pork adulteration in beef based on electronic nose dataset, *Int. J. Intell. Eng. Syst.* 14 (4) (2021) 44–55, <http://dx.doi.org/10.22266/ijies2021.0831.05>.
- [59] D. Aulia, R. Sarno, S.C. Hidayati, M. Rivai, Optimization of the electronic nose sensor array for asthma detection based on genetic algorithm, *IEEE Access*, 2023, <http://dx.doi.org/10.1109/ACCESS.2023.3291451>.
- [60] S. Razavi, Optimizing transformer-based learning with metaheuristic algorithms for accurate and interpretable crude oil price forecasting, *J. Energy Syst. Eng.* 01 (02) (2025) 51–71, <http://dx.doi.org/10.22034/jese.2025.565440.1018>.
- [61] H. Wang, H. Wang, S. Qie, C. Wang, N. Li, H. Wang, Attention-based transformer-LSTM architecture for early diagnosis and staging of early-stage Parkinson's disease using fNIRS data, *Front. Aging Neurosci.* Volume 17 - 2025 (2025) <http://dx.doi.org/10.3389/fnagi.2025.1677722>.
- [62] M. Ardani, R. Sarno, M.M. Khosasih, M. Malikhah, D.P. Purbawa, C. Fatichah, D. Sunaryono, R.I. Susilo, S.I. Sabilla, K.R. Sungkono, Electronic nose signals for analysing similarity of male and female axillary odour to food material aroma., *Int. J. Intell. Eng. Syst.* 15 (5) (2022) <http://dx.doi.org/10.22266/ijies2022.1031.52>.
- [63] M. Hasanah, R.A. Putri, M. Aidiel, R. Putra, T. Ahmad, Analysis of weight-based voting classifier for intrusion detection system, *Int. J. Intell. Eng. Syst.* 17 (2) (2024) 190–200, <http://dx.doi.org/10.22266/ijies2024.0430.17>.
- [64] R. Malikhah, S. Inoue, M.S.H. Ardani, D.P. Purbawa, S.I. Sabilla, K.R. Sungkono, C. Fatichah, D. Sunaryono, A. Bakhtiar, Libriansyah, C.R.S. Prakoeswa, D. Tinduh, Y. Hernaningsih, Detection of infectious respiratory disease through sweat from axillary using an E-nose with stacked deep neural network, *IEEE Access* 10 (2022) 51285–51298, <http://dx.doi.org/10.1109/ACCESS.2022.3173736>.
- [65] X. Wu, H. Li, N. Yu, Development of electrochemical fingerprints for quick sensing identification of epimedium Koreanum nakai, *J. Food Meas. Charact.* (2024) <http://dx.doi.org/10.1007/s11694-024-02875-2>.
- [66] Y. Liu, Intelligent evaluation of antioxidant activity in tea products based on machine learning and cyclic voltammetry, *J. Food Meas. Charact.* 18 (8) (2024) 6451–6458, <http://dx.doi.org/10.1007/s11694-024-02661-0>.