AIMS Environmental Science

AIMS Environmental Science, 13(3): 458–486.
DOI: 10.3934/environsci.2026019
Received: 11 December 2025
Revised: 24 May 2026
Accepted: 10 June 2026
Published: 29 June 2026

https://www.aimspress.com/journal/environmental

Solid waste classification: An approach based on reinforcement learning and differential evolution

Si Li*

Liaoning institute of science and engineering, Jinzhou, Liaoning, 121000, China

* **Correspondence:** Email: qingshanlan@sina.com.

Abstract: As cities grow and intelligent urban areas expand, effectively managing escalating waste volumes, from their generation and sorting to their final disposal, becomes increasingly essential. This investigation introduces a sophisticated deep-learning (DL) approach for categorizing solid waste into multiple waste categories, including glass, metal, paper, plastic, cardboard, and trash. Our model addresses issues overlooked in previous studies, including class imbalance and sensitivity to hyperparameters in classification tasks. It uses wide, dilated convolutional layers that adeptly identify and combine key features for precise classification. Addressing class imbalance, we implement a reinforcement learning (RL) approach, where the agent evaluates each sample individually and classifies it. For every accurate classification, the agent earns rewards, whereas inaccuracies lead to penalties, with greater penalties/rewards applied to the less prevalent class. This system enables the agent to develop an optimal strategy guided by specific reward functions and a well-defined learning environment. To enhance hyperparameter optimization, the proposed approach improves the differential evolution (DE) algorithm using a clustering-guided mutation strategy based on k-means clustering. This strategy clusters the candidate hyperparameter population using k-means, selects the cluster with the lowest average objective value, and uses the best candidate within that cluster to guide the mutation process. A unique method revitalizes the candidate solutions throughout the population, advancing the hyperparameter adjustment process. Exhaustive evaluations on the TrashNet and Trash datasets demonstrate the efficacy of our model, achieving a high classification accuracy of 89.908% on TrashNet and 87.438% on Trash. These findings highlight the system’s capacity to properly handle particular difficulties in solid waste classification, especially data imbalance and hyperparameter sensitivity.

Keywords: solid waste management; reinforcement learning; imbalanced classification; differential evolution; hyperparameter optimization

1. Introduction

Due to technological advances affecting urban lifestyles, numerous challenges emerge daily for those dwelling in cities, notably in solid waste management [1]. Globally, an individual living in an urban environment produces approximately 0.74 kg of solid waste daily, a figure expected to surge to 2.2 billion tons by 2025 [2]. This waste encompasses a variety of materials, including paper, plastics, textiles, glass, metals, rubber, organic matter, and electronic and chemical products, which can be categorized into several material-specific waste classes depending on the classification scheme. Recent data suggest that the most frequently discarded items are paper products, food waste, and glass, which together represent 99.5% of all waste collected. This highlights the disposal of both non-compostable and compostable materials. The categorization of waste by its biodegradability is increasingly vital for effective disposal and recycling practices. This sorting mechanism has recently become a global focal point for scholarly inquiry and research [3].

The adoption of machine learning and DL techniques for sorting solid waste is becoming more prevalent [1,4]. However, these technologies frequently face obstacles like class imbalance, which can undermine the performance of diagnostic applications. To mitigate these issues, measures are undertaken at both the data and algorithmic levels. Regarding the data, strategies like oversampling minority classes or undersampling dominant ones are employed to equalize data representation. On the algorithmic front, adjustments are designed to increase the algorithms' responsiveness to less frequent class instances [5,6]. However, interventions at the data level may result in overtraining, where the model becomes too adept at identifying repeated or artificially augmented instances rather than real-world data. Algorithmically, tweaking the focus of the model may impair its capability to recognize more prevalent conditions. Meanwhile, DRL is gaining recognition as an effective method to address these issues, introducing innovative ways to handle class imbalances [7]. DRL refines classification processes by reducing extraneous data and highlighting essential attributes, proving its effectiveness across various fields. A key benefit of DRL is that it is possible to adapt development strategies derived from received benefits, which is crucial for correcting data imbalances. By tailoring the reward system to favor precise identification of underrepresented classes, DRL models change their focus toward these critical but sometimes overlooked segments. This strategy not only fosters fairness among classes but also enhances the diagnostic accuracy of the system. Nevertheless, DRL techniques still face difficulty adjusting hyperparameters [8].

To address the challenge of hyperparameter sensitivity in Deep DRL, researchers have delved into a variety of optimization strategies, like grid search and evolutionary techniques [9]. The exhaustive search approach methodically assesses various hyperparameter blends in a preset matrix to identify the best-performing sets. Conversely, genetic algorithms utilize evolutionary principles to gradually enhance hyperparameter configurations toward peak efficiency. However, while grid search is thorough, it can be laborious and slow, examining every possible permutation. On the other hand, genetic algorithms are adaptive but may encounter difficulties in achieving convergence and sometimes accept less than optimal outcomes [10]. DE algorithms handle these drawbacks through the use of variation vectors to iteratively refine the pool of candidate solutions, boosting search effectiveness and accelerating convergence while skillfully evading local minima. This method progressively optimizes solutions across generations by capitalizing on differences between randomly selected examples, targeting the universal optimum lacking segmenting hyperparameters. The mentioned framework enhances system accuracy and functionality by effectively balancing exploration with exploitation [11]. The DE algorithm consists of three main stages: mutation, crossover, and selection. During mutation, new candidate solutions are generated by perturbing existing solutions using vector differences within the population. During crossover, the mutated vector is combined with the target vector to produce a trial vector. In the selection stage, the trial vector and the target vector are compared, and the better solution is retained for the next generation. This process maintains population diversity while progressively improving solution quality across iterations [12].

The conducted study introduces a sophisticated deep-learning framework for categorizing urban solid waste into multiple classes, such as glass, metal, paper, plastic, cardboard, and trash, tackling the prevalent issues of class imbalance and hyperparameter sensitivity. Our framework utilizes wide dilated convolutional layers that adeptly detect and assimilate essential characteristics for precise analysis. The classification challenge is framed as a strategic prediction game structured in the form of a sequence of steps involved in decision management. At every interval, the agent is presented with a scenario depicted by a training sample, upon which it makes a classification based on a predefined strategy. Successful classifications yield positive feedback, while errors result in penalties, with higher incentives given for accurately identifying less represented classes. The ultimate objective for the agent is to enhance its overall rewards through accurate classification decisions throughout this process. To improve hyperparameter optimization, the standard DE algorithm is enhanced using a clustering-guided mutation strategy. In this strategy, the population of candidate hyperparameter solutions is grouped using k-means clustering, the cluster with the lowest average objective value is selected, and the best candidate within that cluster is used to guide mutation. This clustering-guided mechanism directs the search toward promising regions of the hyperparameter space while preserving population diversity, thereby improving the efficiency and robustness of hyperparameter optimization.

The primary inputs of this article are detailed as follows:

- This model employs the concept of reinforcement learning to deal with class imbalance through the use of the decision-making approach to classification. In this case, the model will be able to generate step-by-step decisions using a system where there is a changing reward structure for such decisions. The reward structure aims at addressing multi-class imbalance by awarding more rewards for making accurate predictions about less common categories of waste.
- The study enhances the model's performance by integrating a clustering-guided DE algorithm for hyperparameter optimization. This integration reduces the model's sensitivity to hyperparameter settings, a common challenge in deploying DL models. By optimizing these parameters effectively, the proposed clustering-guided DE algorithm ensures that the model achieves optimal performance without the extensive manual tuning that is typically required. This automation of hyperparameter optimization contributes significantly to the model's efficiency and effectiveness in classifying solid waste.
- A notable improvement in the system is the enhancement of the DE algorithm through a clustering-guided mutation strategy. In this strategy, k-means clustering is used to organize the population of candidate hyperparameter solutions into groups. The cluster with the lowest average objective function value is then selected, and the best candidate within this cluster is used to guide the mutation process. This strategy refines the search process within the DE framework by directing mutation toward promising regions of the hyperparameter space while preserving population diversity. As a result, the enhanced DE algorithm provides more robust and effective hyperparameter

optimization, improving the model’s adaptability and classification performance on challenging datasets.

The organization of this paper is segmented into several parts to facilitate comprehension and ensure a coherent structure. Section 2 provides an exhaustive review of the existing literature, setting the stage for our investigation. Section 3 delves into the proposed methods, elaborating on the critical techniques employed. Section 4 discusses the findings of our experimental testing, highlighting their relevance and the lessons learned. Section 5 wraps up the discussion by recapping the main discoveries and suggesting avenues for future inquiry, identifying specific areas that merit additional investigation.

2. Related work

Over the past few years, numerous strategies involving machine learning and DL paradigms have emerged for sorting solid waste. The following parts present an in-depth analysis of each category.

2.1. Machine learning

Meza et al. [13] investigated the implementation of artificial intelligence (AI) to improve management practices for municipal solid waste (MSW), specifically to aid in decision-making. Their research evaluated two AI techniques: support vector machines (SVM) and long short-term memory (LSTM) networks. The analysis includes testing the SVM's effectiveness with certain data categories and applying the LSTM in various configurations, as well as temporal analyses to study patterns from yearly MSW collection cycles. Abu-Qdais et al. [14] pursued enhancements in sustainable waste management by automating the classification of solid waste. Their work assessed the performance of both conventional and advanced machine learning techniques in categorizing waste. They compared traditional approaches, such as random forest (RF) and SVM, with a DL-based convolutional neural network (CNN). Additionally, they introduced a novel DL model named JONET, which integrates a pretrained DenseNet 201 with a newly designed architecture that includes a 1024-neuron fully connected layer designed to classify six distinct waste types. They utilized both publicly accessible and specially created datasets for training and validating the models, aiming to heighten classification precision and enhance safety for sanitation workers. Tao et al. [15] tackled the issue of analyzing thermochemical properties and extracting inorganic constituents from MSW, which are traditionally managed using time-consuming, intricate, and material-consuming techniques. Their research introduced a method that mixes hyperspectral imaging with machine learning algorithms to boost the accuracy of identifying inorganic materials and to estimate the content of carbon, hydrogen, oxygen, nitrogen, and the low heating value (LHV) of organic substances. Principal component analysis was employed to distill critical data from spectral images, while artificial neural networks were used for effective classification and predictive analysis, offering a viable method for the quick assessment and categorization of MSW elements. Adeleke et al. [16] crafted an adaptive neuro-fuzzy inference system (ANFIS) model, refined using evolutionary strategies such as particle swarm optimization (PSO) and genetic algorithms (GA), to correct data collection errors in waste management that affect sustainable practices. This research examined the variability in the physical makeup of solid waste in Johannesburg across seasons by employing three clustering methods: grid partitioning (GP), subtractive clustering (SC), and fuzzy c-means (FCM), each assessed under varied hyperparameter configurations. This method underscores the role of artificial intelligence in depicting the physical attributes of waste, which is vital for improving waste management techniques. Fatovatikhah et al. [17] tackled the urgent issue of forecasting flood-related waste amid increasing extreme weather events driven by climate change. Their research utilized LSTM and SVM models to forecast flood-related waste effectively. They analyzed two comprehensive datasets from *Advancing Sustainable Materials Management: Facts and Figures 2015*, which spans nine years (1960–2015) and includes data on materials produced and processed in municipal waste. This study sorted waste into categories such as paper, glass, metals, plastics, rubber, textiles, wood, food, yard waste, and various inorganic materials to bolster flood management and mitigation strategies. S Shah et al. [18] responded to the mounting issues of solid waste control in urban areas of India by developing a classification method for seven types of solid waste into biodegradable and non-biodegradable classifications. Their approach leveraged image data from both training and testing sets, utilizing hybrid features (HF) to extract features via MeQryEP and principal component analysis (PCA) to identify textural and structural properties. These features were combined and processed using a deep neural network (DNN) to categorize the waste, significantly improving the accuracy of classification beyond conventional methods like PCA combined with other algorithms.

2.2. Deep learning

The use of DL for sorting solid waste has been emphasized in several research initiatives, marking its rising significance in this domain. Li et al. [2] introduced the use of CNN and Graph-LSTM, two sophisticated deep-learning approaches, to tackle recycling and waste management issues in developing nations where the quantity of municipal solid waste is increasing. Their research employed a CNN to classify six types of trash materials (paperboard, metal, glass, plastic, paper, and organic) found on conveyor belts within waste handling facilities. The addition of Graph-LSTM contributed to the system's capability to process long-term relationships, enhancing overall performance and generalization, which facilitated more effective and autonomous waste management systems. Chen et al. [19] investigated how noise-enhanced data augmentation influences the accuracy of classifying municipal solid waste (MSW) using computer vision techniques. Their evaluation included two sophisticated classification frameworks, CNN and deep residual shrinkage networks (DRSN), to overcome obstacles like small sample sizes and inconsistencies in MSW image features. By adopting noise-enhanced data augmentation, they aimed to refine the training process and increase the system's tolerance to noisy environments, a key factor for improving MSW categorization in commercial settings. Jayaraman et al. [20] developed the MSW-Net model, a complex hierarchical stacking framework designed to streamline MSW classification. This model integrates a bespoke CNN with a Bayesian-Optimized MobileNet, employing gradient boosting as the primary meta-classifier. This novel method was geared toward optimizing solid waste management, minimizing manual sorting and associated health hazards, and supporting municipal entities in accurately classifying waste with reduced human input. Niu et al. [21] tackled the detection of solid waste sites in urban settings using an innovative DL model that merges a multi-scale dilated CNN with a swin-transformer. This model excels at discerning both intricate local details and broader global insights from high-resolution remote sensing imagery, thus enhancing solid waste mapping in various urban areas without the need for pixel-specific annotations. The approach demonstrated considerable progress in urban waste management by adeptly integrating diverse spatial details. Murugan et al. [22] introduced a cutting-edge method that incorporates CNNs within an easily navigable HTML and Django framework to advance solid waste management (SWM) techniques. This system improves waste prediction and sorting, enabling users to effortlessly classify and identify waste types, including the separation of biodegradable from non-biodegradable substances. It blends sophisticated DL algorithms with intuitive interfaces, fostering responsible disposal practices and positively impacting environmental sustainability. Prakash et al. [23] unveiled the municipal solid waste prediction model employing a tree hierarchical deep CNN optimized with a balancing composite motion optimization algorithm to boost prediction precision and reduce computational demands in waste management frameworks. Applied across various districts of Chennai, this methodology utilizes morphological filtering and extended empirical wavelet transformations for data preparation before applying the THDCNN-BCMOA algorithm. Developed in Python, this strategy forecasted various waste categories for 2025–2035, showing notable gains in precision and efficiency compared to previous techniques. Yudhana et al. [24] employed machine learning for global waste management challenges, focusing on waste sorting. They crafted a CNN model that delivers efficient waste categorization, employing sophisticated preprocessing methods and a streamlined design to improve performance. Evaluated using the Kaggle data, this model emphasizes the likelihood of machine learning to offer accurate and effective solutions for minimizing landfill waste, thus safeguarding the environment and conserving resources through precise waste sorting. Zhang et al. [1] crafted an advanced hybrid DL framework for sorting urban waste into organic and recyclable categories, addressing the essential demand for effective waste management in expanding smart cities. This framework integrates various phases: data gathering and preprocessing, feature extraction using AlexNet, waste prediction through a deep belief network (DBN), and hyperparameter fine-tuning with Optuna. This comprehensive method boosted the accuracy of waste prediction and classification, enhancing waste management strategies. Lilhore et al. [25] explored improvements in waste handling through an innovative hybrid CNN-LSTM model incorporating transfer learning to refine waste collection, sorting, composting, and disposal within the framework of smart city development. Utilizing transfer learning from ImageNet and advanced data augmentation tactics, this model proficiently addresses challenges like overfitting and limited sample sizes. Its efficacy was evaluated on the TrashNet dataset, which includes 27027 images, using various CNN architectures such as VGG-16, ResNet-34, ResNet-50, and AlexNet. These models were tested over numerous epochs with an adaptive moment estimator (AME) optimization algorithm, focusing on metrics like precision, recall, training and testing losses, and accuracy to enhance and evaluate model performance.

The current literature outlines various approaches to solid waste categorization, but these methods frequently encounter problems such as class imbalances and the intricacies of hyperparameter adjustment. In response, our research employs a reinforcement learning strategy that adeptly addresses class imbalances. We have crafted a reward mechanism that boosts the identification of less prevalent classes, providing greater incentives for correctly spotting rare occurrences, thus enhancing classification precision and deepening the model’s grasp of these pivotal groups. Additionally, we have incorporated a clustering-guided DE algorithm that minimizes the model's dependency on hyperparameters, making the optimization process more efficient.

3. Materials and methods

Figure 1 demonstrates the suggested framework based on RL and aimed at classifying solid wastes into multiple classes. First, it utilizes a DL-based classifier that includes three convolutional layers with dilations, allowing the extraction of visual features. In the new scheme, RL is incorporated into the process of classification: the image goes through the dilated CNN, after which the obtained feature representation constitutes the RL state; then, the Q-network calculates the Q-values for each class; finally, the chosen action represents the class assigned to the object. The reward depends on both prediction quality and the class frequency.

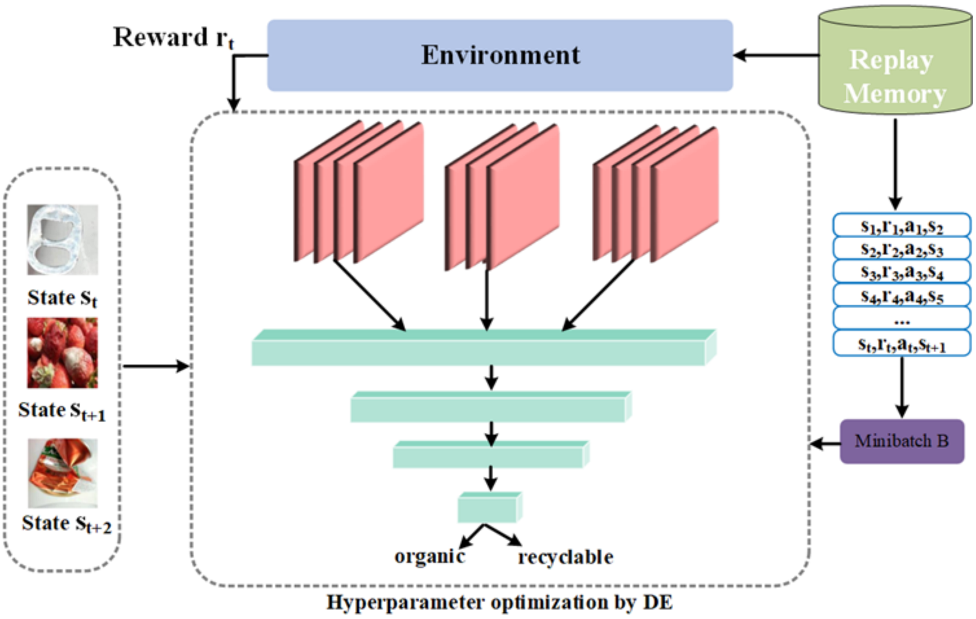


Figure 1. Structure of the reinforcement learning–based multi-class solid waste classification model, including the flow from dilated CNN feature extraction to Q-value generation.

Figure 2 shows the relationship between the wide-dilated CNN architecture and the Q-network within the suggested reinforcement learning–based approach to multi-class solid waste classification. As shown, the input image of the waste goes through several layers of dilated convolutions to generate feature maps (or feature vectors). The extracted CNN features are used as the representation of the reinforcement learning state and fed into the Q-network to predict Q-values for all possible actions of the classes. The class with the maximum Q-value is chosen as the classification output of the waste. This prediction is compared with the actual label to obtain a class frequency-dependent reward. In the final step, the temporal difference update applies this reward to the Q-network's weights.

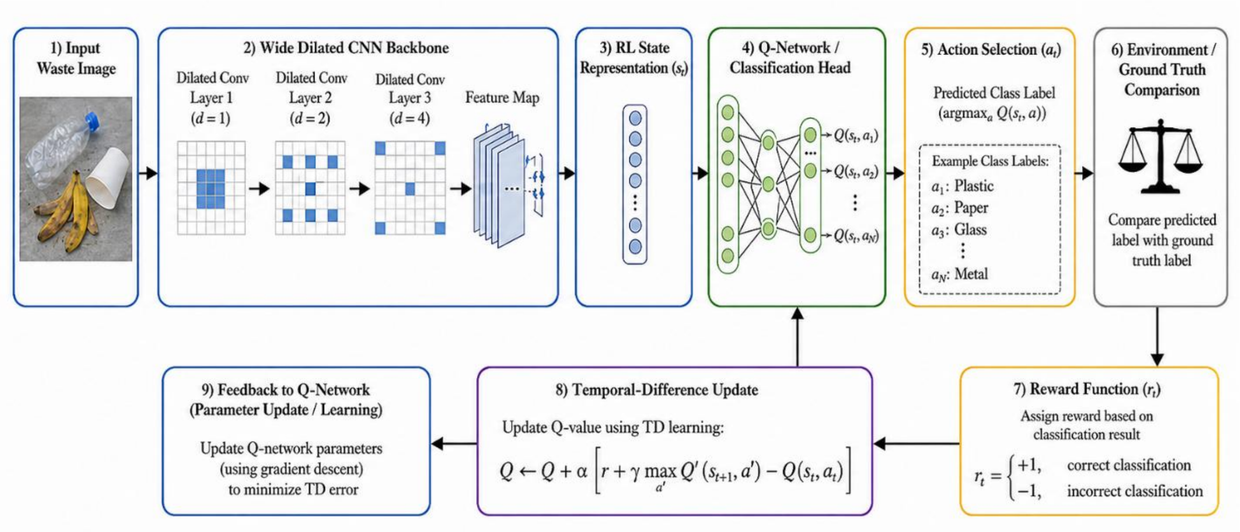


Figure 2. Interaction between the dilated CNN backbone and Q-network in the proposed reinforcement learning–based multi-class solid waste classification framework.

The RL-based classification methodology follows these steps: Input of a waste image → feature extraction using wide dilated CNN layers to form a state s_t → passing of the information to the Q-network (classification head) to get Q-values for all possible classes → choice of action as the class label prediction a_t → calculation of reward proportional to class frequency → Q-update based on temporal difference.

3.1. Training

An agent learns decision-making skills through repeated interaction in the RL process. One goal is to learn an optimal policy, denoted as π, that maps each state into actions such that the expected sum of future rewards is maximized. During the entire process, the agent keeps interacting with the environment in terms of observation, action selection, reward, and policy adjustment based on the feedback from the environment.

Within the proposed framework, the Markov decision process (MDP) is defined as the tuple $(S, A, P, \rho_0, \gamma)$, where S is the state space, A is the action space, P is the transition function, r is the reward function, ρ_0 is the initial-state distribution, and γ is the discount factor. In this supervised multi-class image-classification setting, one training episode is formed by sequentially presenting waste samples to the agent. At each time step t, the input waste image x_t is first processed by the wide dilated CNN backbone, and the extracted feature representation is used as the RL state s_t. Therefore, the state is not the raw image but the CNN-derived feature vector or feature map. The action a_t corresponds to selecting one class label from the dataset-specific waste categories. The transition corresponds to moving from the current training sample to the next sample after the classification decision. The reward $r(s_t, a_t)$ evaluates whether the selected class matches the ground-truth label while assigning class-frequency-sensitive rewards to reduce bias toward more frequent classes. The discounted cumulative reward at time step t is expressed as $R_t = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k)$, where γ controls the contribution of future rewards.

The expected benefits of adhering to strategy π and selecting action a in state s, referred to as Q-values, assess the merit of that action within that state. These values are calculated as follows:

$$Q^{\pi}(s,a) = E[R_t|s_t = s, a_t = a, \pi] \tag{1}$$

The best action-value function, denoted as $Q^{\pi}(s,a)$, quantifies the maximum predicted reward obtainable under any strategy post existing in conditions and executing action a. This function is articulated as follows:

$$Q^{*}(s,a) = max_{\pi}\ E[R_t|s_t = s, a_t = a, \pi] \tag{2}$$

This method is defined by the Bellman equation, which posits that the top-expected value of an operation yields the sum of the instant award and the best-expected value of subsequent actions. The Bellman equation is articulated as follows:

$$Q^{*}(s,a)=E[r + \gamma\ max_{a'}Q^{*}(s',a')|s_t = s, a_t = a] \tag{3}$$

The derivation of the optimal action-value function utilizes a recursive method according to the Bellman equation, outlined as follows:

$$Q_{i+1}(s,a)=E[r + \gamma\ max_{a'}Q_i(s',a')|s_t = s, a_t = a] \tag{4}$$

Q learning endeavors to incrementally refine the prime value roles. Starting with the first value estimations for all state-action combinations and systematically adjusting these estimates at each step as follows:

$$Q(s_t,a_t)= Q(s_t,a_t) + \alpha(r_{t+1} + \gamma\ max_{a'}Q(s_{t+1},a') - Q(s_t,a_t)) \tag{5}$$

where α is the learning rate parameter. The above helps the agent to improve its classification strategy, taking into account both the immediate reward and the expected future reward.

The reinforcement learning approach described above uses the temporal difference loss for training the Q-network. The loss is a measure of how far the prediction is from the actual Q-value, which is calculated from the immediate reward and the future reward estimate. The temporal difference (TD) loss for a particular transition can be written as follows:

$$TD\ Loss\ (s_t,a_t) = (r_{t+1} + \gamma\ max_{a'}Q(s_{t+1},a') - Q(s_t,a_t))^2 \tag{6}$$

This equation measures the squared difference between the predicted Q-value for a particular condition–action pair and the combination of the immediate reward plus the maximum expected future reward, denoted by γ. Through minimizing this discrepancy, the neural network methodically tunes its parameters to more accurately match the estimated Q-values to the designated target Q-values, thus enhancing the agent's approach to maximize overall rewards. This improvement, driven by the decrease in the TD loss, plays a critical role in the agent's continual progress and strategic enhancement for better decision-making in the RL framework.

Within our scheme, the definitions of condition, operation, and award are established as follows:

- State s_t: At time step t, the state s_t is defined as the feature representation obtained from the wide dilated CNN after processing the input waste image x_t. Therefore, the raw image is not directly used as the RL state; instead, the CNN feature vector or feature map summarizes the visual patterns required for class discrimination.
- Action a_t: Refers to the selection of one particular class label from the set of actions, which is defined in relation to the given dataset. For the TrashNet dataset, the action space consists of

cardboard, glass, metal, paper, plastic, and trash; in the case of the Trash dataset, it is glass, metal, paper, and plastic. The UCB approach is used to strike a balance between exploring possible class labels and exploiting existing values of Q. In this improved algorithm, the action selection process can be summarized as follows:

$$a_t = arg \max_a [Q_t(a) + \varphi \sqrt{\frac{\ln n_t}{n_t(a)}}] \tag{7}$$

where $Q_t(a)$ is the estimated Q-value of the action a at time step t, n_t denotes the total number of action-selection decisions up to time step t, $n_t(a)$ is the number of times the action a has been selected, and $\varphi > 0$ is the exploration coefficient that controls the contribution of the UCB exploration term. A larger φ encourages more exploration of less-selected class actions, whereas a smaller φ makes the policy more exploitative.

- Reward r_t: The reward function is designed to guide the agent toward accurate and balanced multi-class classification. Because TrashNet and Trash are multi-class datasets, the reward function is formulated using class-frequency-based weights. Each class is assigned a weight according to the number of samples in that class. Let K denote the number of classes, N the total number of training samples, and N_c the number of samples belonging to class c. The initial class weight is defined as follows:

$$w_c = \frac{N}{K \times N_c} \tag{8}$$

To keep the reward scale stable, the class weights are normalized as follows:

$$\overline{w}_c = \frac{w_c}{max_{j \in K} \times w_j} \tag{9}$$

Where $\overline{w}_c$ is the normalized class-frequency weight. Classes with fewer samples are assigned larger normalized weights, whereas more frequent classes are assigned smaller weights. Therefore, rare classes receive stronger learning signals during training. The multi-class reward function is then defined as follows:

$$r_t(s_t, a_t, y_t) = \begin{cases} +\lambda \overline{w}_{yt}, if\ a_t = y_t \\ -\lambda \overline{w}_{yt}, if\ a_t \neq y_t \end{cases} \tag{10}$$

Where y_t is the ground-truth class label of the input sample, a_t is the class selected by the agent, $\overline{w}_{yt}$ is the normalized weight of the true class, and λ is the reward-scaling coefficient. This formulation extends the reward system to the multi-class imbalanced classification setting by assigning stronger reward signals to correctly classified underrepresented classes and stronger penalties to misclassified underrepresented classes. Consequently, the learner is encouraged to develop a balanced classification strategy across all waste categories instead of being biased toward more frequent classes.

Table 1 presents the hyperparameters modified in the course of our study, outlining a range of values informed by previous research in solid waste classification.

Table 1. Model parameter setting for the suggested system.

Hyperparameter	Range	Best value
Learning rate	0–1	0.15
Batch size	8–512	61
Number of epochs	32–1024	260
Activation function	Options: ReLU, permeable ReLU, Linear, Tanh, S-shaped function	ReLU
Dropout rate	0–1	0.18
Number of MLP layers	1–6	5
Number of CNN layers	1–6	3
Discount factor (γ)	0–1	0.22

3.2. Hyperparameter tuning

Optimizing hyperparameters is important for RL models because it markedly influences their learning efficiency, firmness, and overall performance. Hyperparameters in reinforcement learning, such as learning rates, discount factors, and the size of the replay buffer, dictate how quickly and effectively an RL agent can learn from its environment. Proper tuning ensures that the model does not converge too slowly, avoiding high computational costs, or too quickly, which can lead to premature convergence on suboptimal policies. Additionally, optimal hyperparameters help balance the exploration–exploitation trade-off, which is crucial for the RL agent to discover and leverage the best long-term strategies rather than settling for immediate rewards. Thus, hyperparameter optimization is about enhancing performance and ensuring the robustness and adaptability of RL models to varied environments and tasks [26].

3.2.1. Random key (RK)

The article adopts the RK technique [27] over model parameter enhancement due to the goals and constraints of our study. First, the RK technique is prized for its simplicity and versatility. It allows for straightforward implementation even in complex optimization tasks, using arrays of numbers to represent solutions that are easy to manipulate and evaluate. This simplicity makes it highly adaptable to a wide range of hyperparameter types without the need for specialized encoding schemes. Another significant advantage of using the RK technique is its compatibility with both continuous and discrete variables. This feature is especially beneficial in machine learning, where model parameters such as learning rate or batch size can vary in nature. The technique adapts seamlessly to these variations, providing a unified approach to optimizing a mixed set of hyperparameters. Additionally, the RK method is robust against the stochastic nature of training DL models. By generating and iteratively refining a population of RKs based on performance, it avoids getting trapped in local optima and promotes exploration across the solution space. Moreover, the RK technique scales effectively with the problem's scope. As the number of hyperparameters increases, its population-based framework efficiently manages the expanded search space. This scalability is crucial for optimizing modern deep-learning models that often include numerous hyperparameters. Furthermore, the ease of integrating the RK technique with other evolutionary algorithms enhances its capabilities. It can be combined with crossover and mutation operations from genetic algorithms or differential evolution strategies, thereby improving exploratory capabilities and accelerating convergence. By selecting the RK technique, the paper strategically utilizes a robust, adaptable, and scalable optimization method that meets the demands of complex algorithmic tuning, aiming to enhance the overall performance and efficiency of the proposed machine learning model.

The RK technique utilizes a coding mechanism comprising T quantitative vectors, having D dimensions denoted as $p_1, p_2, \dots, p_T$. The given entities form a collective set, each representing a possible solution linked to model parameters through an association mechanism termed the RK. In optimizing C model parameters (as detailed in Table 1), every model parameter, indexed out of 1 to C, is expressed through D_c positions within the entity. For ongoing model parameters, D_c is set to an individual, resulting in the overall dimensionality D computed as $D = \sum_{c=1}^{C} D_c$. Each vector p_i is segmented into C parts, with each part holding D_c positions that propose possible values for a hyperparameter. For classification-type tuning parameters, the RK method transforms the quantitative portion D_c of entity p_i into a specified class-specific list, MAP_c, that provides choices for the c^{th} model parameters. This alteration is produced by classifying items in the D_c portion and using the rank of the top entry as a reference measure into the MAP_c list to pick the appropriate class label. This approach guarantees that evolutionary mechanisms like mutation, crossover, and selection, when performed on the quantitative entities p_i, yield configurations that seamlessly integrate encompassing categorically and discrete model parameter values. Figure 3 displays this using a D_c of 5, where the RK, once ordered as a sequence of R values, crucially aligns with a predetermined set of selections. This structure promotes the emergence of essential attributes inside the core, boosting the model's capability to evaluate and rank the significance of predictors efficiently.

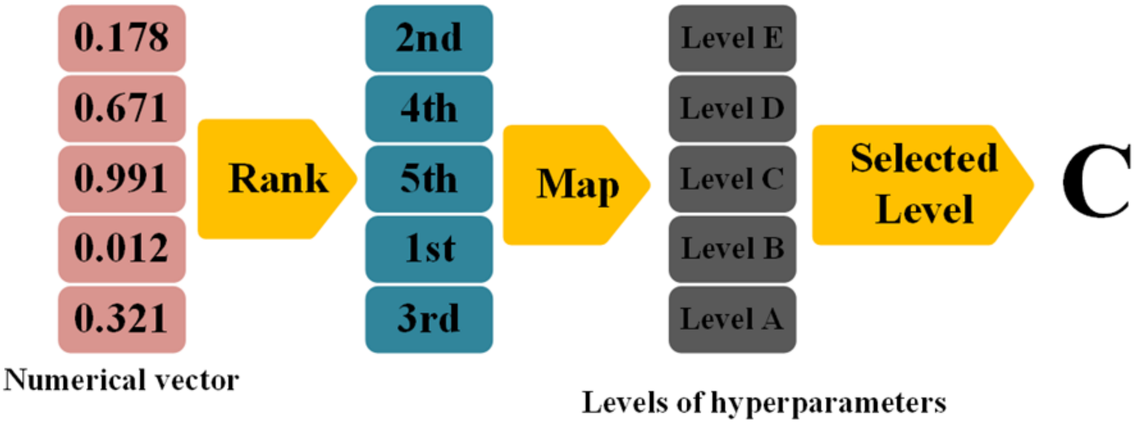


Figure 3. Depiction of the RK encoding technique for hyperparameter tuning.

3.2.2. Differential evolution

To enrich the efficacy of the RK procedure, we operate the DE algorithm. The integration of DE enhances the optimization capabilities of the RK technique in several crucial ways, making this combination particularly effective for hyperparameter tuning in complex machine learning models. First, DE is renowned for its robustness and ability to navigate large, complex search spaces efficiently. This feature is particularly beneficial when paired with the RK technique, as it significantly improves the convergence rate toward optimal solutions. DE achieves this through its strategic mechanism of mutation and crossover, which generates diverse candidate solutions and secures a thorough exploration of the search space. This diversity prevents premature convergence on suboptimal solutions, a common challenge in optimization tasks. Moreover, DE's simplicity and flexibility in handling various objective functions make it an excellent choice for enhancing the RK technique. Whether the optimization task entails unbroken and separate values, DE's mechanisms adapt seamlessly, providing a robust and unified approach to hyperparameter optimization. This flexibility is crucial for machine learning applications where hyperparameter types and their interactions vary significantly. Another key advantage of incorporating DE is its proven effectiveness in various real-world optimization problems across domains. This empirical validation provides a strong foundation for its application in enhancing the RK technique, ensuring that the combined approach is theoretically sound and practical. Lastly, using DE with the RK technique allows for an adaptive optimization process. DE's parameters, such as the crossover rate and mutation factor, can be fine-tuned dynamically during the optimization process to respond to the specific needs and challenges of the task. This adaptability leads to more tailored and more effective optimization outcomes. By integrating DE with the RK technique, we aim to leverage the strengths of both approaches: the broad search capabilities and simple representation of RK, enhanced by the strategic exploration and adaptive mechanisms of DE. This combination promises to strongly boost the functionality and utility of the model parameter adjustment process, ultimately leading to better-performing machine learning models [12].

DE operates through three core phases: mutation, crossover, and selection. The mutation phase has an important impact on the algorithm's development by injecting fresh diversity of genetic traits into the solution pool. This injection is achieved by altering the current solution with components derived from others within the population, using predefined mathematical strategies. Typically, this includes updating a baseline solution by incorporating the normalized difference between two additional solutions, thus generating a novel variation. These mutations are critical as they sustain diversity within the solution set and unearth possibly more effective solutions. The capacity of this step to create a separate array of superior remedies is fundamental to DE's success, propelling the algorithm aimed at the most optimal outcomes.

Inside the DE methodology, the variation operation starts as follows [12]:

$$\vec{v}_{i,g} = \vec{x}_{r_1,g} + F\,(\vec{x}_{r_2,g} - \vec{x}_{r_3,g}) \tag{11}$$

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & \text{if } \mathrm{rand}(0,1) \leq \mathrm{CR} \text{ or } \mathrm{j} = j_{rand} \\ x_{i,j,g} & \text{otherwise} \end{cases} \tag{12}$$

Within the context of the framework, CR represents the crossover rate, and j_{rand} is an arbitrarily selected index from the set {1, 2, ..., D}, with D representing the dimensions of the solution vector. After this, the selection stage takes place, during which both the initial target entity and the newly created trail vector, resulting from the crossover process, are examined in opposition to one another. The algorithm then opts for the better solution, furthering the adaptive cycle. This stage is vital as it promotes ongoing enhancement of the solutions in the pool, guaranteeing that only optimal solutions are preserved.

The refined DE model provides a novel mutation approach based on the latest field innovations [28]. Initially, this method implements k-means clustering on the existing generation to delineate uniquely identified clusters within the search space. This categorization splits the community into many sets, including the quantity of clusters, k, drawn from the range using a random process $[2, \sqrt{N}]$. The emphasis is subsequently directed toward the set with the lowest average performance metric, marking it as the principal focus of thorough examination. Within the DE system, an innovative mutation operator is employed, structured around clustering techniques, as follows:

$$\overrightarrow{v^{clu}}_{i} = \overrightarrow{win}_g + F\,(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}) \tag{13}$$

In the formula, $\vec{x}_{r_1,g}$ and $\vec{x}_{r_2,g}$ are two randomly selected candidate solutions within the community, while $\overrightarrow{win}_g$ denotes the top-performing solution in a particularly advantageous cluster. One should recognize that $\overrightarrow{win}_g$ might not always show the solution with maximal efficiency across the entire population. The mutation strategy, focused on clusters, is carried out across M iterations. Subsequently, the population evolves through many steps, as stated by the procedures of the generic population-based algorithm [29]:

- Selection: k solutions are chosen arbitrarily to function as the first centroids for k-means clustering.
- Generation: M new candidate solutions are generated through mutation, forming the group v^{clu}.
- Exchange: M applicants are randomly chosen from the current population to establish a new set named B.
- Revision: The most effective M applicants from the combined groups of v^{clu} and B are chosen to create the new set B′. The population is reorganized by merging (P-B) with B′.

4. Empirical evaluation

4.1. Datasets

This research employed two well-known datasets to ascertain the efficacy of the suggested system:

- TrashNet [30]: Available on platforms like Kaggle and GitHub, it features an extensive collection of waste images sorted into six groups: glass, metal, plastic, cardboard, paper, and trash, cumulatively amounting to 82 MB in JPG format. The data includes diverse items such as glass bottles, metal cans, plastic containers, pieces of cardboard, and paper products. Comprehensive details are provided in Table 2.

Table 2. Detailed specifications of the TrashNet dataset.

Class	Detail	Full data
Stiff paper	Torn cardboard, packing cartons	393
Glass	Broken containers, jars, glass	491
Metal	Can, crushed can, jar cover	400
Paper	Envelope, paper, page of magazines, newspaper	584
Plastic	Bottles, box, milk bottle	482
Trash	Various kinds of frames, such as chips packets, chocolate packets, sauce packets	127

- Trash [31]: Hosted on Kaggle, this dataset comprises 3907 images, including 136 in PNG format and the remainder in JPG, totaling 92 MB. It is divided into four categories: glass, paper, metal, and plastics, with each category containing items pertinent to its type, such as shattered glass, plastic bags, metal lids, and waste paper products. Further specifics of this dataset are listed in Table 3.

Table 3. Detailed specifications of the Trash dataset.

Class	Elements	Full data
Glass	Vase, containers, jugs, glass	914
Metal	Cans, bottle caps, crushed cans	801
Paper	Stiff paper, cups, juice packs	1187
Plastic	Containers, bottle caps, packets, disposable bags	1005

4.2. Metrics

To assess our model, we make use of a detailed array of criteria: accuracy, F-measure, geometric mean (G-mean), and area under the curve (AUC). Each metric is chosen on account of its unique proficiency for evaluating different facets of model performance in multiple classes, such as glass, metal, paper, plastic, cardboard, and trash.

- Accuracy: This metric is fundamental as it provides a straightforward measure of the total precision of the system. It calculates the proportion of total accurate forecasts (each of two true positives and true negatives) for every forecast created. This gives us a general idea of the model's effectiveness over every category.
- F-measure: We include the F-measure because it balances the system's accuracy and recall, offering a more nuanced view of its performance, especially when dealing with imbalanced data. The F-measure is beneficial as it combines precision (the accuracy of positive predictions) and recall (the ability to find all relevant instances) into a single metric, which helps evaluate the model's accuracy in predicting the smaller group.
- Geometric mean (G-mean): G-mean acts as a measurement tool to evaluate the level of equality among the performances in all classes. In a multiple-class environment, it is obtainable through the recognition accuracy of each class. Therefore, this tool helps determine whether the model is performing consistently in all classes, including those that appear frequently and rarely. This metric is useful when dealing with imbalanced data, especially since it is not heavily influenced by the most frequently appearing classes.

- Area under the curve (AUC): The AUC measures the ability of the model to avoid false classifications. It offers an aggregate performance measure across all possible classification thresholds. AUC is particularly valuable as it is independent of any threshold. It measures the quality of the model's output probabilities and indicates how well the model can separate the two classes.

Accuracy, F-measure, and G-means metrics are calculated as follows:

$$Accuracy = \frac{True\ Positives\ (TP) + True\ Negatives\ (TN)}{Total\ observations} \tag{14}$$

$$F - measure\ (F1\ Score) = 2 \times \frac{Precision + Recall}{Precision + Recall} \tag{15}$$

$$Precision = \frac{TP}{TP + False\ Positives\ (FP)} \tag{16}$$

$$G - means = \sqrt{Recall \times Specificity} \tag{17}$$

Where

$$Recall = \frac{TP}{TP + False\ Negatives\ (FN)} \tag{18}$$

$$Specificity = \frac{TN}{TN + FP} \tag{19}$$

4.3. Overall performance

The analysis was conducted on Python 3.8 on Windows OS 64-bit, facilitated by the TensorFlow 2.3 framework for implementing DL models. The hardware setup included an NVIDIA GeForce RTX 2080 Ti GPU, which supplied the processing capability to handle the extensive training cycles required by DL architectures. We utilized the Keras library, a high-level neural network API running on top of TensorFlow, which allowed for easy and fast prototyping and ran seamlessly on both CPUs and GPUs.

A 5-fold stratified cross-validation strategy was used to evaluate the model's performance and reliability, ensuring that each fold preserved the class distribution of the original dataset. This method is crucial for maintaining the integrity of our data's statistical properties, mainly when dealing with imbalanced classes, which is often the situation in real-world data. Layered cross-validation enhances the generalizability and validity of our model by ensuring that every training and validation set is a representative subset of the entire data population. It prevents the model from being biased toward more frequent classes and allows for more accurate and reliable performance metrics. This technique is particularly effective in providing a robust evaluation framework that mimics varied scenarios that the model might encounter when deployed in practical applications.

For each fold, hyperparameter optimization was performed only on the training and validation subsets, while the corresponding test fold remained completely unseen until final evaluation. Therefore, the proposed clustering-guided DE framework had no access to the test fold during hyperparameter optimization. This prevents information leakage and ensures that the reported accuracy reflects generalization performance rather than overfitting the test data.

In the evaluation phase, our model underwent rigorous evaluation, being compared to a variety of other models: three from traditional machine learning (SVMLSTM [13], SWC-ML [14], and HIML [15]), and six from advanced DL frameworks (Graph-LSTM [2], DRSN-CNN [19], MSW-Net [20], CNN-Transformer [21], SWM-CNN [22], and Hierarchical-CNN [23]). We also explored the impact of omitting key components such as RL and hyperparameter optimization (HO) to gauge their influence on the model's total efficacy. The outcomes of these assessments, based on tests conducted using the TrashNet and Trash datasets, are given in Tables 4 and 5.

For the TrashNet dataset, the comparison between traditional and advanced machine learning models reveals significant advancements by models like Hierarchical-CNN and SWM-CNN over simpler ones like SVMLSTM and SWC-ML, with enhancements in accuracy and F-measure reaching as much as 8.7% and 7.8%, respectively. Our proposed model, particularly the versions without RL and HO, displayed even more substantial improvements. The complete version of our proposed model demonstrated an impressive increase in productivity metrics, with accuracy and F-measure improving by at least 4.24% and 4.09%, respectively, compared to the best-performing advanced model, Hierarchical-CNN. This underscores the benefits of incorporating both RL and HO into the model. In comparing the derivatives of our model, the version without RL already performed well, yet incorporating RL led to further enhancements, with accuracy and F-measure increasing by 3.34% and 2.43%, respectively. Similarly, removing HO decreased performance, emphasizing the merit of optimized model parameter settings for achieving peak model efficiency.

For the Trash dataset, similar patterns were observed. Advanced DL models like Hierarchical-CNN and CNN-Transformer substantially outperformed more traditional approaches, achieving accuracy improvements of up to 7.9% over models like SVMLSTM. The fully integrated version of our proposed model outdid all others, showing improvements in accuracy by 6.4% and in F-measure by 3.25% compared to the highest-performing existing model, Hierarchical-CNN. When analyzing the variants of our proposed model on the Trash dataset, the version without RL had less pronounced performance enhancements, with a noticeable drop in F-measure of 2.07%. The absence of hyperparameter optimization also resulted in a clear decline in presentation metrics, reaffirming the significance of these components in managing complex classification tasks such as solid waste classification. These analyses not only highlight our model's superiority but also confirm its integrated features' crucial role in boosting classification accuracy and reliability.

The advantage of the developed model is mainly a result of the use of RL to effectively manage class imbalances and the strategic implementation of DE for hyperparameter optimization. These features address critical challenges that have often limited the success of previous models. Specifically, RL is adept at dynamically adjusting to the varying distributions of classes within training data, enabling the model to focus on underrepresented classes and thus improve overall accuracy in a balanced manner. This is particularly crucial in solid waste classification, where some waste types are inherently less common but equally important to identify accurately. Furthermore, using DE for hyperparameter optimization enhances the model’s ability to fine-tune parameters efficiently, ensuring optimal learning conditions that adapt to the unique challenges presented by the data. These enhancements not only contribute to the superior performance of our model but also highlight the deficiencies in previous models, which may not have adequately addressed the issues of class imbalance and were likely less effective at optimizing their learning processes for maximum performance. This strategic incorporation of advanced methodologies distinctly boosts the effectiveness and reliability of our model compared to its predecessors.

Table 4. Benchmarking of our system versus other systems on the TrashNet dataset.

Model	Accuracy	F-measure	G-means	AUC
SVMLSTM [13]	76.968 ± 0.055	80.823 ± 0.069	81.183 ± 0.086	0.733 ± 0.044
SWC-ML [14]	78.093 ± 0.085	81.901 ± 0.054	82.245 ± 0.069	0.743 ± 0.099
HIML [15]	79.250 ± 0.031	82.585 ± 0.078	82.945 ± 0.054	0.759 ± 0.064
Graph-LSTM [2]	80.209 ± 0.071	83.249 ± 0.088	83.593 ± 0.077	0.768 ± 0.036
DRSN-CNN [19]	81.321 ± 0.038	84.398 ± 0.027	84.701 ± 0.035	0.786 ± 0.073
MSW-Net [20]	82.504 ± 0.063	85.537 ± 0.012	85.812 ± 0.075	0.794 ± 0.025
CNN-Transformer [21]	83.099 ± 0.000	86.774 ± 0.061	87.062 ± 0.017	0.799 ± 0.077
SWM-CNN [22]	84.682 ± 0.069	88.405 ± 0.002	88.689 ± 0.027	0.806 ± 0.036
Hierarchical-CNN [23]	85.668 ± 0.053	89.588 ± 0.047	89.871 ± 0.077	0.814 ± 0.061
Proposed w/o RL	86.569 ± 0.084	91.252 ± 0.054	91.516 ± 0.071	0.826 ± 0.084
Suggested w/o HO	88.275 ± 0.059	92.726 ± 0.096	92.967 ± 0.082	0.834 ± 0.048
Proposed model	89.908 ± 0.021	93.679 ± 0.016	93.956 ± 0.083	0.851 ± 0.027

Table 5. Comparative analysis of our model versus other models on the Trash dataset.

Model	Accuracy	F-measure	G-means	AUC
SVMLSTM [13]	73.105 ± 0.091	78.343 ± 0.068	79.069 ± 0.088	0.714 ± 0.098
SWC-ML [14]	74.899 ± 0.004	79.392 ± 0.021	80.189 ± 0.046	0.724 ± 0.087
HIML [15]	75.842 ± 0.021	80.669 ± 0.058	81.410 ± 0.001	0.734 ± 0.076
Graph-LSTM [2]	76.480 ± 0.058	81.667 ± 0.018	82.480 ± 0.054	0.741 ± 0.020
DRSN-CNN [19]	77.122 ± 0.028	82.518 ± 0.089	83.300 ± 0.072	0.756 ± 0.090
MSW-Net [20]	77.875 ± 0.063	83.867 ± 0.017	84.684 ± 0.097	0.771 ± 0.048
CNN-Transformer [21]	79.271 ± 0.024	85.309 ± 0.015	86.127 ± 0.096	0.778 ± 0.053
SWM-CNN [22]	80.459 ± 0.095	86.567 ± 0.057	87.381 ± 0.052	0.787 ± 0.039
Hierarchical-CNN [23]	81.042 ± 0.028	87.223 ± 0.065	88.020 ± 0.043	0.794 ± 0.050
Proposed w/o RL	82.493 ± 0.020	88.403 ± 0.085	89.204 ± 0.041	0.800 ± 0.050
Proposed w/o HO	83.353 ± 0.065	89.247 ± 0.034	90.096 ± 0.033	0.811 ± 0.040
Proposed model	87.438 ± 0.076	90.476 ± 0.083	91.246 ± 0.076	0.825 ± 0.060

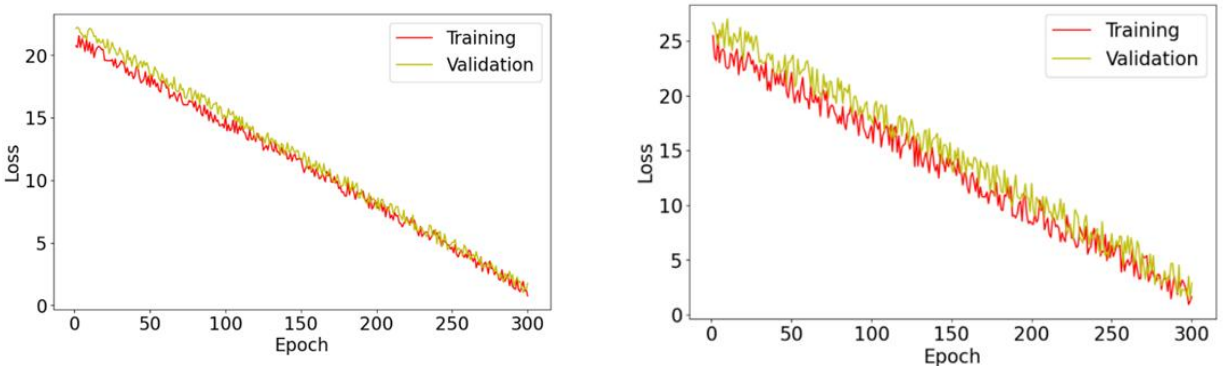
To find out the numerical importance of our proposed model's dominance compared to current systems, we conducted paired t-tests using results from the TrashNet and Trash datasets. P-values were computed for each analysis between our model and existing systems over four key metrics: accuracy, F-measure, G-means, and AUC. The results show enhancements that are numerically meaningful and not caused by random variation. For example, in the TrashNet dataset, p-values for comparisons with advanced models such as Hierarchical-CNN were below 0.05 for all metrics, confirming statistically significant enhancements in performance. Notably, improvements in accuracy and F-measure yielded p-values of 0.01 and 0.02, respectively, demonstrating notable advancements. A similar pattern was evident in the Trash dataset, with p-values of 0.03 for G-means and 0.04 for AUC when compared against the same model, verifying our model's consistent and reliable superiority across various datasets. These findings support the effectiveness of integrating RL for imbalanced classification and the proposed DE algorithm for hyperparameter optimization. The computational efficiency of the examined models, including runtime and GPU memory consumption, is reported in Table 6 for the TrashNet and Trash datasets.

Table 6. Comparison of runtime (S) performance and GPU utilization of various models using the TrashNet and Trash datasets.

Model	TrashNet		Trash	
	(S)	(GB)	(S)	(GB)
SVMLSTM [13]	2056	16.8	2125	18.3
SWC-ML [14]	1963	18.9	2230	20.5
HIML [15]	1752	17.4	2036	19.5
Graph-LSTM [2]	2365	23.5	2412	24.3
DRSN-CNN [19]	2574	26.2	2463	28.9
MSW-Net [20]	2652	25.8	2532	24.1
CNN-Transformer [21]	2752	24.8	2274	26.6
SWM-CNN [22]	3254	21.5	3522	23.7
Hierarchical-CNN [23]	2632	20.5	2621	21.4
Proposed model	2741	19.1	2412	19.2

Table 6 shows that the suggested model shows a decent runtime result compared to other deep-learning models, with significantly smaller GPU memory consumption. In particular, it uses 2741 seconds and 19.1 GB on the TrashNet dataset and 2412 seconds and 19.2 GB on the Trash dataset. Meanwhile, even though some of the examined models (e.g., DRSN-CNN and MSW-Net) demonstrate comparable runtime results, they require significantly more GPU memory, reaching 26.2 GB and 25.8 GB, respectively, on the TrashNet dataset. Hence, it can be stated that the developed classifier provides a good trade-off between computation and classification efficiency, which is crucial for real-world waste classification applications.

In addition, it should be mentioned that there is also an additional computational cost involved in performing the k-means clustering in the differential evolution (DE) algorithm. However, this cost does not become excessive since the k-means operation is done only on the population of hyperparameter vectors. Thus, the main factors influencing the computational complexity of k-means in DE are limited to the population size, the number of parameters of the vector, the number of clusters, and the number of iterations, which remain relatively small compared to CNN–RL training. Moreover, most of the time reported in Table 6 refers to the execution time of the trained models with a certain combination of hyperparameters; therefore, DE-based search is performed offline. From a theoretical perspective, the use of clustering in DE can improve the exploration–exploitation balance by grouping similar candidate solutions and guiding mutation toward promising regions of the search space. Figure 4 presents the training and validation loss curves of the proposed model on the TrashNet and Trash datasets over 300 epochs.



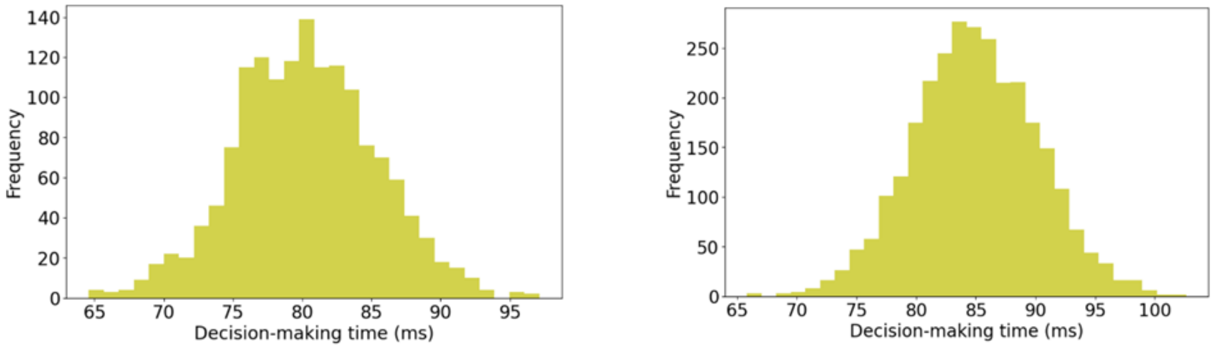
a

b

Figure 4. Loss progression curves for training and validation. a) TrashNet and b) Trash datasets.

Based on Figure 4, the model demonstrates strong robustness and generalization ability without evident overfitting. For both datasets, the loss curves for training and validation closely align throughout the training process, indicating that the model learns effectively from the training data without merely memorizing it. The consistent decrease in loss for both the training and validation phases suggests that the model effectively captures the underlying patterns rather than the noise within the data. Furthermore, the absence of divergence between these curves toward the later epochs strongly supports the model's stability and generalization capabilities. This parallel decline and convergence of loss values affirm that the proposed enhancements in the model, particularly its architecture and training regimen, successfully mitigate the risk of overfitting, making it well-suited for practical applications where generalization to unseen data is crucial.

Figure 5 presents the distribution of decision-making time for the proposed system on the TrashNet and Trash datasets, showing how consistently the model performs under operational conditions.



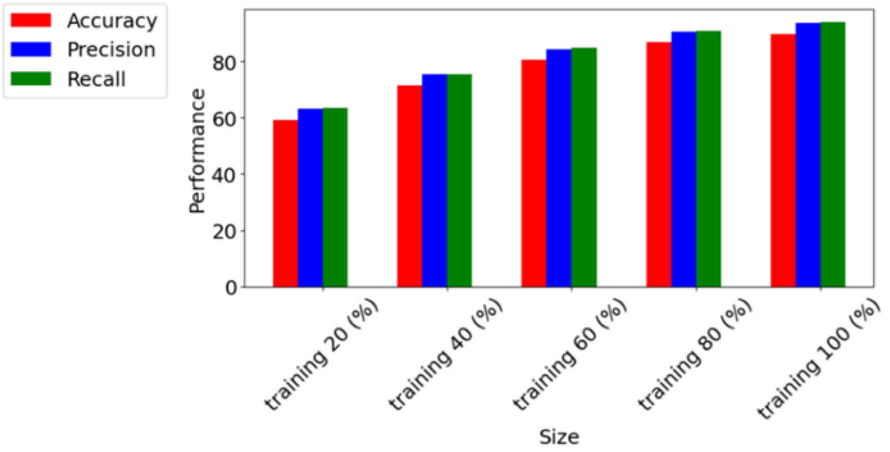
a

b

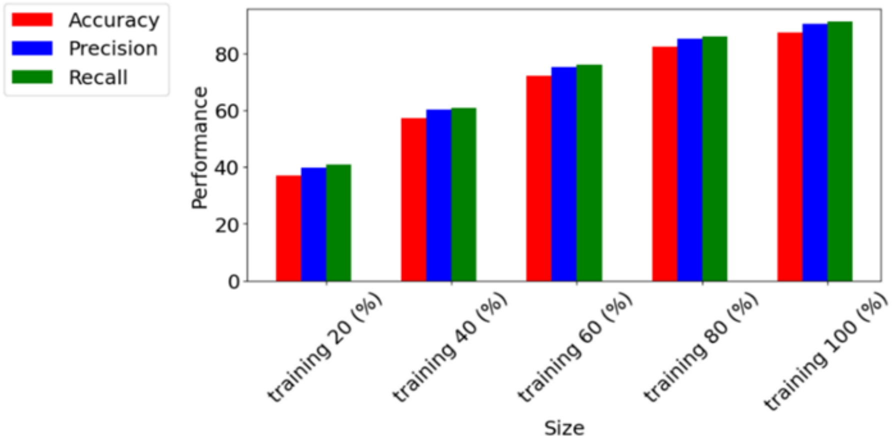
Figure 5. Distribution of decision-making time for the suggested model on (a) TrashNet and (b) Trash datasets.

The results indicate that the proposed model maintains stable and rapid decision-making behavior across both datasets, confirming its suitability for efficient solid waste classification. The histograms show a tight distribution around a mean decision-making time, with most decisions for the TrashNet dataset clustering between 75 and 85 ms, and a slightly broader spread for the Trash dataset centered around 80 to 90 ms. These distributions indicate that the proposed model not only maintains a quick response rate but also does so consistently across different datasets. The peak near the mode for both data suggests that the model efficiently processes data with minimal variation in response times, an essential characteristic for real-time applications where rapid and reliable decision-making is critical. This consistency and predictability in decision-making times demonstrate the model's suitability for deployment in environments requiring fast and stable performance in solid waste classification.

Figure 6 presents the scalability of the suggested system by showing the performance metrics, accuracy, precision, and recall, across varying proportions of building information between 20% and 100% for both the TrashNet and Trash datasets. For the TrashNet dataset, a clear trend shows enhancement in all metrics as the training data size rises. Starting with a training size of 20%, the model shows modest accuracy, precision, and recall, significantly improving as more data is utilized. Notably, accuracy and recall exhibit a steep increase between 20% and 60% of training data, stabilizing near their peak as the data size approaches 100%. This pattern underscores the model's ability to effectively learn and generalize from increasing volumes of data, indicating robustness and scalability. Similarly, the Trash dataset reflects consistent growth in performance metrics with increased training sizes. The initial leap in performance metrics from 20% to 40% training data is particularly noteworthy, suggesting the model's capacity to adapt and improve with additional data. Beyond 60%, the increments become more gradual, yet they continue to rise, reaching their zenith at full data utilization. This steady enhancement across all performance indicators with larger training sizes confirms the model's scalability and reliability for extensive real-world applications, making it particularly suitable for diverse and voluminous waste classification tasks. Both data exhibit the proposed model’s proficiency in scaling with data size while maintaining high performance, affirming its possible application for broader deployment in solid waste management systems.



a)



b)

Figure 6. Performance scalability of the suggested system on a) TrashNet and b) Trash datasets across different training sizes.

4.3.1. Analysis of the proposed DE

This section analyzes the proposed clustering-guided DE algorithm against a group of proven metaheuristic optimization strategies for hyperparameter tuning. This assessment includes six algorithms—HMS [32], Salp swarm algorithm (SSA) [33], cuckoo optimization algorithm (COA) [34], Firefly algorithm (FA) [35], Bat algorithm (BA) [36], and artificial bee colony (ABC) [37]—and the original DE algorithm. The findings, obtained from the TrashNet and Trash datasets, are reported in Tables 7 and 8.

The k-means clustering mechanism used in the proposed DE algorithm improves the search process by dividing the population into structured subregions of the hyperparameter space. While the standard DE algorithm maintains diversity through random vector differences, this search mechanism may generate mutation directions that are not strongly related to promising regions of the solution space. In contrast, the clustering-guided mutation strategy identifies promising subregions and selects strong candidate solutions from these regions to guide mutation. In this way, the proposed method strengthens exploitation around promising solutions while preserving global exploration across the search space. Therefore, the proposed DE algorithm achieves a better exploration–exploitation balance, which explains its superior performance in Tables 7 and 8.

For the TrashNet dataset, the proposed DE algorithm exhibits a substantial performance improvement. Compared to the next-best-performing algorithm, ABC, the proposed DE achieves an accuracy improvement of approximately 8.54%, an F-measure improvement of around 9.49%, and a G-means increase of 8.56%. These enhancements are also reflected in the AUC metric, where the proposed DE shows an improvement of approximately 6.38% over ABC. This improvement indicates the ability of the proposed DE algorithm to navigate the complex hyperparameter space more effectively, mainly due to its clustering-guided mutation strategy. On the Trash dataset, a similar trend is observed, where the proposed DE algorithm again achieves the best performance. It shows an accuracy improvement of approximately 5.87% over the standard DE, along with F-measure and G-means improvements of about 5.88% and 6.93%, respectively. These results indicate that the clustering-guided enhancement of the standard DE algorithm contributes significantly to more effective hyperparameter tuning under different dataset conditions. Overall, the results confirm that the proposed DE algorithm optimizes hyperparameter settings more effectively than the compared metaheuristic methods and provides a robust optimization strategy for solid waste classification.

Table 7. Performance comparison of the proposed DE algorithm with other metaheuristic algorithms on the TrashNet dataset.

Model	Accuracy	F-measure	G-means	AUC
HMS	77.092 ± 0.076	79.845 ± 0.090	80.832 ± 0.004	0.739 ± 0.018
SSA	78.859 ± 0.026	80.962 ± 0.077	82.020 ± 0.098	0.745 ± 0.027
COA	80.309 ± 0.012	82.202 ± 0.017	83.237 ± 0.056	0.754 ± 0.047
FA	81.322 ± 0.072	83.044 ± 0.029	84.105 ± 0.074	0.771 ± 0.011
BA	82.077 ± 0.049	84.481 ± 0.058	85.456 ± 0.022	0.787 ± 0.015
ABC	82.769 ± 0.026	85.598 ± 0.096	86.538 ± 0.046	0.800 ± 0.076
DE	83.554 ± 0.001	86.517 ± 0.066	87.518 ± 0.079	0.814 ± 0.001
Proposed DE	89.908 ± 0.021	93.679 ± 0.016	93.956 ± 0.083	0.851 ± 0.027

Table 8. Performance comparison of the proposed DE algorithm with other metaheuristic algorithms on the Trash dataset.

Model	Accuracy	F-measure	G-means	AUC
HMS	76.236 ± 0.058	78.860 ± 0.009	79.617 ± 0.058	0.706 ± 0.039
SSA	77.952 ± 0.047	80.073 ± 0.027	80.781 ± 0.002	0.715 ± 0.067
COA	78.745 ± 0.013	80.908 ± 0.040	81.606 ± 0.045	0.731 ± 0.061
FA	79.385 ± 0.082	82.047 ± 0.003	82.756 ± 0.084	0.740 ± 0.057
BA	79.957 ± 0.099	82.673 ± 0.083	83.381 ± 0.060	0.749 ± 0.073
ABC	81.320 ± 0.089	83.418 ± 0.083	84.148 ± 0.072	0.766 ± 0.025
DE	82.577 ± 0.068	84.600 ± 0.095	85.316 ± 0.036	0.783 ± 0.068
Proposed DE	87.438 ± 0.076	90.476 ± 0.083	91.246 ± 0.076	0.825 ± 0.060

Figure 7 illustrates the optimization behavior of the proposed DE algorithm over 300 iterations on the TrashNet and Trash datasets.

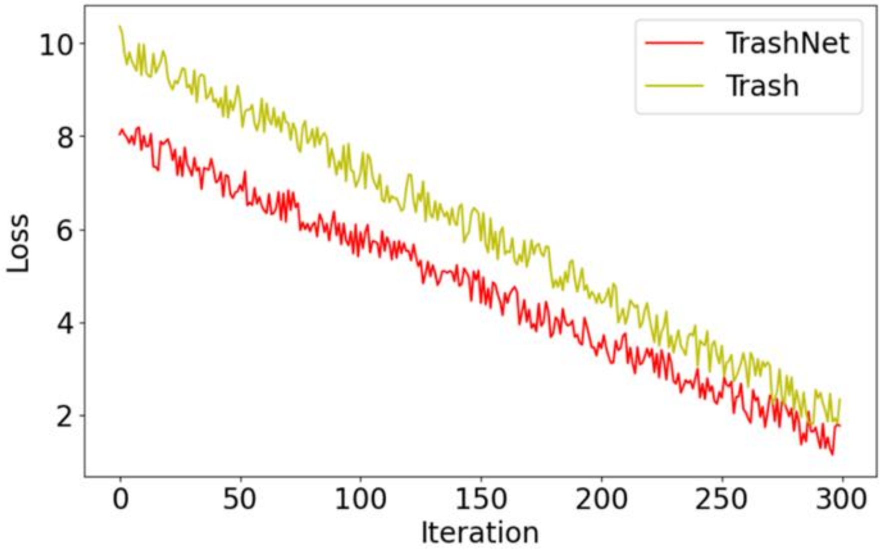


Figure 7. Optimization behavior of the proposed DE algorithm over 300 iterations on the TrashNet and Trash datasets.

Based on Figure 7, for the TrashNet dataset, the loss reduction trajectory shows a consistent and steep decline, indicating that the proposed DE algorithm effectively refines the model hyperparameters to improve classification performance. The initial rapid decrease followed by a steadier, gradual reduction suggests that the proposed DE algorithm can quickly escape suboptimal hyperparameter regions, improve convergence, and stabilize near an optimal solution. Similarly, the loss curve for the Trash dataset demonstrates robust optimization behavior, with a comparable pattern of sharp initial decline followed by gradual stabilization. This pattern reflects the proposed DE algorithm's ability to adaptively fine-tune hyperparameters according to dataset-specific characteristics while maintaining stable classification performance. The convergence patterns observed in both datasets confirm the proposed DE algorithm's effectiveness in handling different data conditions, making it a suitable optimization tool for complex machine learning tasks such as solid waste classification. Its consistent behavior across both datasets further supports its reliability and practical applicability in improving the robustness and effectiveness of environmental classification models.

4.3.2. Discussion

This article presents an innovative deep-learning model to categorize solid waste into multiple classes, such as glass, metal, paper, plastic, cardboard, and trash. The model integrates several advanced components, including wide dilated convolutional layers for feature extraction, RL for managing class imbalance, and a clustering-guided DE algorithm for robust hyperparameter optimization.

Reinforcement learning (RL) is highly effective when dealing with cases of data imbalance, which is common in solid waste classification, where some waste classes have fewer data points compared to others. RL facilitates learning using a reward-penalty scheme whose intensity is proportional to the number of instances in each class. Through this dynamic process, there will be reduced bias toward the dominant classes since the policy network will be motivated to give more consideration to the underrepresented and difficult classes. Through this process, the identification of infrequent classes will be improved without compromising the entire classification accuracy.

The proposed clustering-guided DE algorithm plays a crucial role in tuning the model hyperparameters, which strongly influence classification performance and computational efficiency. The original DE algorithm is robust for nonlinear, non-differentiable, and multi-modal optimization problems; however, its mutation process may rely heavily on random vector differences. To improve this process, the proposed method uses a clustering-guided mutation strategy, where k-means clustering organizes the population into structurally similar groups, and mutation is guided by the best candidate from the most promising cluster. This strategy provides a more directed search within the hyperparameter space while still preserving population diversity. As a result, the enhanced DE algorithm improves the probability of escaping local optima, identifies more effective hyperparameter settings, and contributes to higher classification accuracy and better generalization under real-world data variability.

The methodologies embedded within this solid waste classification model have broad applicability across similar domains that require efficient sorting and categorization based on complex and often imbalanced data. For example, in recycling management, where materials must be accurately sorted into numerous categories based on their composition and recyclability, the system’s functionality to detect subtle differences and learn from imbalanced data can significantly enhance sorting efficiency and accuracy. Additionally, the model's approach can be adapted in healthcare for patient diagnosis and treatment classification, where data imbalance is prevalent and accurate classification is crucial. In agricultural technology, this model could assist in categorizing plant diseases or pests from visual data, where early and accurate detection is crucial for effective intervention. Each of these domains shares the common challenge of requiring precise and reliable classification in the face of varied and often insufficient examples of critical categories, making the model’s architecture and methodologies ideally suited for these tasks.

Accurate classification of municipal solid waste makes operations more efficient and helps attain carbon neutrality. The Fixed Time & Place technique, as well as the TPDF method that sorts waste into four classes, has an impact on the percentage of waste sent to landfills rather than recycling plants. As a result of employing multi-class reinforcement learning for early sorting, waste, recyclable, and reusable materials can be sorted out. This means less waste is sent to the landfill, where it would produce unnecessary methane emissions when decomposing. Thus, the early sorting of waste improves the process of attaining carbon neutrality.

The restrictions of the suggested system are as follows:

- One significant limitation of the proposed model lies in its computational intensity. The integration of wide dilated convolutional layers, while effective for feature extraction, requires substantial computational resources, particularly with respect to processing power and memory usage. This can limit the model's scalability, especially when deployed in environments with restricted computational capacity, such as mobile devices or on-site processing units in waste management facilities. To mitigate this issue, model pruning or the implementation of more efficient convolution operations could be considered. These strategies would aim to reduce the complexity of the model without a substantial loss in performance, thereby making it more adaptable to various operational settings.
- The efficiency of the system is contingent upon the standard diversity of the training information. In scenarios where data is noisy, incomplete, or lacks representativeness, the model's performance may degrade, reflecting poor generalization to real-world conditions. This is particularly

pertinent in waste management, where the appearance and condition of materials can vary dramatically. Enhancing data preprocessing techniques, such as implementing more robust noise reduction and augmentation strategies, could improve model robustness. Additionally, continually updating the training dataset to reflect new waste and disposal conditions can help to gradually improve the system's connection and accuracy.

- While the proposed clustering-guided DE algorithm aims to optimize hyperparameters effectively, the model's performance remains sensitive to these settings. Inappropriate hyperparameter values can lead to suboptimal learning behaviors, such as overfitting or underfitting, particularly in a complex model architecture. To address this, further research could be directed toward developing more adaptive hyperparameter optimization techniques that could dynamically adjust parameters in response to variations in data or model performance indicators during training.
- The rapid evolution of products and materials can lead to significant changes in waste composition over short periods, possibly rendering the model's classifications obsolete if it cannot adapt quickly enough. To overcome this, implementing online learning or incremental learning strategies in which the model continuously adjusts its parameters as new waste types are introduced, without needing retraining from scratch, could be beneficial. This would ensure that the model remains efficient as new forms of waste are introduced into the system.

5. Conclusion

This investigation developed and validated a deep-learning model for multi-class solid waste classification, covering categories such as glass, metal, paper, plastic, cardboard, and trash. The proposed model addresses class imbalance and hyperparameter sensitivity by combining wide dilated convolutional layers, reinforcement learning, and an enhanced clustering-guided DE algorithm. The wide dilated convolutional layers extract discriminative visual features, while the RL mechanism improves balanced classification by assigning class-frequency-sensitive rewards and penalties. In addition, the clustering-guided DE algorithm improves hyperparameter optimization by directing the search toward promising regions of the hyperparameter space while preserving population diversity. Exhaustive evaluations on the TrashNet and Trash datasets demonstrate the efficacy of our model, achieving a high classification accuracy of 89.908% on TrashNet and 87.438% on Trash. These results demonstrate the technical effectiveness of the proposed framework and its potential contribution to scalable and intelligent solid waste management systems. In upcoming research, the model could be enhanced with many impactful methods. First, additional DL architectures, like Generative Adversarial Networks (GANs), might be explored to address scalability and improve classification accuracy. GANs could generate synthetic training data, which is beneficial in overcoming issues related to the limited availability of labeled data in certain classes, thus better addressing class imbalance problems. Second, exploring the incorporation of real-time data streams into the model could extend its applicability to dynamic urban waste management systems. By enabling the model to adapt to real-time changes in waste composition and volume, the classification system could provide more timely and accurate predictions, aiding in more responsive and efficient waste management strategies. This could involve developing an online learning version of the model that can continuously update its parameters without retraining from scratch, which would be essential for adapting to changes in waste types and quantities over time.

Use of AI tools declaration

During the preparation of this work, the author used ChatGPT, version 4, for language editing and readability improvement. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the published content.

Acknowledgments

We would like to thank you for following the instructions above very closely in advance. It will definitely save us lot of time and expedite the process of your paper's publication.

Conflict of interest

The authors declare no conflicts of interest.

References

1. Zhang H, Cao H, Zhou Y, et al. (2023) Hybrid deep learning model for accurate classification of solid waste in the society. *Urban Clim* 49: 101485. https://doi.org/10.1016/j.uclim.2023.101485
2. Li N, Chen Y (2023) Municipal solid waste classification and real-time detection using deep learning methods. *Urban Clim* 49: 101462. https://doi.org/10.1016/j.uclim.2023.101462
3. Chu X, Chu Z, Wang X, et al. (2023) Comparative analysis on the performances of implementing compulsory and advocative policies in municipal solid waste classification. *Environ Impact Assess Rev* 99: 106982. https://doi.org/10.1016/j.eiar.2022.106982
4. Wu TW, Zhang H, Peng W, et al. (2023) Applications of convolutional neural networks for intelligent waste identification and recycling: A review. *Resour Conserv Recycl* 190: 106813. https://doi.org/10.1016/j.resconrec.2022.106813
5. Kosolwattana T, Liu C, Hu R, et al. (2023) A self-inspected adaptive SMOTE algorithm (SASMOTE) for highly imbalanced data classification in healthcare. *BioData Min* 16: 15. https://doi.org/10.1186/s13040-023-00332-2
6. Wang Z, Liu Q (2023) Imbalanced Data Classification Method Based on LSSASMOTE. *IEEE Access* 11: 32252–32260. https://doi.org/10.1109/ACCESS.2023.3262657
7. Yang J, El-Bouri R, O'Donoghue O, et al. (2023) Deep reinforcement learning for multi-class imbalanced training: applications in healthcare. *Mach Learn* https://doi.org/10.1007/s10994-023-06403-1
8. Danaei S, et al. (2022) Myocarditis diagnosis: a method using mutual learning-based abc and reinforcement learning, In: 2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo), New York: IEEE, 000265-000270. https://doi.org/10.1109/CINTI-MACRo57952.2022.10029424
9. Ali YA, Awwad EM, Al-Razgan M, et al. (2023) Hyperparameter search for machine learning algorithms for optimizing the computational complexity. *Processes* 11: 349. https://doi.org/10.3390/pr11020349
10. Franchini G (2024) GreenNAS: A Green Approach to the Hyperparameters Tuning in Deep Learning. *Mathematics* 12: 850. https://doi.org/10.3390/math12060850

11. Jiang J, Wen Z, Mansoor A, et al. (2024) Efficient Hyperparameter Optimization with Adaptive Fidelity Identification, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 26181–26190. https://doi.org/10.1109/CVPR52733.2024.02476
12. Moravvej SV, Mousavirad SJ, Oliva D, et al. (2022) An improved de algorithm to optimise the learning process of a bert-based plagiarism detection model, In: 2022 IEEE Congress on Evolutionary Computation (CEC), New York: IEEE, 1–7. https://doi.org/10.1109/CEC55065.2022.9870233
13. Solano Meza JK, Orjuela Yepes D, Rodrigo-Ilarri J, et al. (2023) Comparative analysis of the implementation of support vector machines and long short-term memory artificial neural networks in municipal solid waste management models in megacities. *Int J Environ Res Public Health* 20: 4256. https://doi.org/10.3390/ijerph20054256
14. Abu-Qdais H, Shatnawi N, AL-Alamie EA (2024) Intelligent system for solid waste classification using combination of image processing and machine learning models. *J Exp Theor Artif Intell In press*. https://doi.org/10.1080/0952813X.2024.2319241
15. Tao J, Gu Y, Hao X, et al. (2023) Combination of hyperspectral imaging and machine learning models for fast characterization and classification of municipal solid waste. *Resour Conserv Recycl* 188: 106731. https://doi.org/10.1016/j.resconrec.2022.106731
16. Adeleke O, Akinlabi S, Jen TC, et al. (2023) A machine learning approach for investigating the impact of seasonal variation on physical composition of municipal solid waste. *J Reliable Intell Environ* 9: 99–118. https://doi.org/10.1007/s40860-021-00164-8
17. Fatovatikhah F, Ahmedy I, Noor RM (2024) Waste Prediction Approach Using Hybrid Long Short-Term Memory with Support Vector Machine. *Int J Comput Intell Syst* 17: 103. https://doi.org/10.1007/s44196-024-00481-9
18. Shah KB, Visalakshi S, Panigrahi R (2023) Seven class solid waste management-hybrid features based deep neural network. Int J Syst Des Comput 1: 1–10. https://doi.org/10.55529/ijsdc.11.1.10
19. Chen C, Liang R, Song M, et al. (2024) Noise-assisted data enhancement promoting image classification of municipal solid waste. *Resour Conserv Recycl* 209: 107790. https://doi.org/10.1016/j.resconrec.2024.107790
20. Jayaraman V, Lakshminarayanan AR (2024) MSW-Net: A hierarchical stacking Model for automated municipal solid waste classification. *J Air Waste Manage Assoc*. https://doi.org/10.1080/10962247.2024.2367123
21. Niu B, Feng Q, Yang J, et al. (2023) Solid waste mapping based on very high-resolution remote sensing imagery and a novel deep learning approach. *Geocarto Int* 38: 2164361. https://doi.org/10.1080/10106049.2022.2164361
22. Murugan PR, Muneeswaran V, Vardhan NVS, et al. (2024) Deep Learning-based Hybrid Image Classification Model for Solid Waste Management, In: 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS), New York: IEEE, 168–172. https://doi.org/10.1109/ICC-ROBINS61647.2024.10557452
23. Prakash TS, Patnayakuni SP, Shibu S (2023) Municipal solid waste prediction using tree hierarchical deep convolutional neural network optimized with balancing composite motion optimization algorithm. *J Exp Theor Artif Intell*. https://doi.org/10.1080/0952813X.2023.2254924

24. Yudhana A, Fahmi M (2023) Improving Waste Classification Using Convolutional Neural Networks: An Application of Machine Learning for Effective Environmental Management. *Rev D'Intelligence Artif* 37: In press. https://doi.org/10.18280/ria.370415
25. Lilhore UK, Simaiya S, Dalal S, et al. (2024) A smart waste classification model using hybrid CNN-LSTM with transfer learning for sustainable environment. *Multimedia Tools Appl* 83: 29505–29529. https://doi.org/10.1007/s11042-023-16518-2
26. Karl F, Pielok T, Moosbauer J, et al. (2023) Multi-objective hyperparameter optimization in machine learning—An overview. *ACM Trans Evol Learn Optim* 3: 1–50. https://doi.org/10.1145/3611387
27. Bean JC (1994) Genetic algorithms and random keys for sequencing and optimization. *ORSA J Comput* 6: 154–160. https://doi.org/10.1287/ijoc.6.2.154
28. Mousavirad SJ, Schaefer G, Korovin I, et al. (2021) An enhanced differential evolution algorithm using a novel clustering-based mutation operator, In: 2021 IEEE international conference on systems, man, and cybernetics (SMC), New York: IEEE, 176–181. https://doi.org/10.1109/SMC52423.2021.9658739
29. Deb K (2005) A population-based algorithm-generator for real-parameter optimization. *Soft Comput* 9: 236–253. https://doi.org/10.1007/s00500-004-0371-x
30. Yang M, Thung G (2016) Classification of trash for recyclability status. CS229 Project Report 2016: 3. https://cs229.stanford.edu/proj2016/report/ThungYang-ClassificationOfTrashForRecyclabilityStatus-report.pdf
31. Ochecolo DI, Trash dataset. Kaggle, 2024. Available from: https://www.kaggle.com/dimonisochecolo/trash-dataset.
32. Mousavirad SJ, Schaefer G, Ebrahimpour-Komleh H (2021) The human mental search algorithm for solving optimisation problems, In: Enabling AI Applications in Data Science, Cham: Springer, 27–47. https://doi.org/10.1007/978-3-030-51552-2_3
33. Tubishat M, Ja'afar S, Alswaitti M, et al. (2021) Dynamic salp swarm algorithm for feature selection. *Expert Syst Appl* 164: 113873. https://doi.org/10.1016/j.eswa.2020.113873
34. Yang XS, Deb S (2009) Cuckoo search via Lévy flights, In: 2009 World congress on nature & biologically inspired computing (NaBIC), New York: IEEE, 210–214. https://doi.org/10.1109/NABIC.2009.5393690
35. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. *Int J Bio-Inspired Comput* 2: 78–84. https://doi.org/10.1504/IJBIC.2010.032124
36. Yildizdan G, Baykan ÖK (2020) A novel modified bat algorithm hybridizing by differential evolution algorithm. *Expert Syst Appl* 141: 112949. https://doi.org/10.1016/j.eswa.2019.112949
37. Vakilian S, Moravvej SV, Fanian A (2021) Using the artificial bee colony (ABC) algorithm in collaboration with the fog nodes in the Internet of Things three-layer architecture, In: 2021 29th Iranian Conference on Electrical Engineering (ICEE), New York: IEEE, 509–513. https://doi.org/10.1109/ICEE52713.2021.9528250

AIMS Press