mathematics

MDPI

Article

Hybrid Genetic Algorithm and CMA-ES Optimization for RNN-Based Chemical Compound Classification

Zhenkai Guo [1], Dianlong Hou [2] and Qiang He [3,*]

[1] School of Mathematics and Statistics Science, Ludong University, Yantai 264025, China; guozhenkai@ldu.edu.cn
[2] Dongying United Petroleum & Chemicals Co., Ltd., Dongying 257347, China
[3] College of Medicine and Biological Information Engineering, Northeastern University, Shenyang 110169, China
* Correspondence: heqiang@bmie.neu.edu.cn

Abstract: The compound classification strategies addressed in this study encounter challenges related to either low efficiency or accuracy. Precise classification of chemical compounds from SMILES symbols holds significant importance in domains such as drug discovery, materials science, and environmental toxicology. In this paper, we introduce a novel hybrid optimization framework named GA-CMA-ES which integrates Genetic Algorithms (GA) and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to train Recurrent Neural Networks (RNNs) for compound classification. Leveraging the global exploration capabilities og GAs and local exploration abilities of the CMA-ES, the proposed method achieves notable performance, attaining an 83% classification accuracy on a benchmark dataset, surpassing the baseline method. Furthermore, the hybrid approach exhibits enhanced convergence speed, computational efficiency, and robustness across diverse datasets and levels of complexity.

Keywords: compound classification; genetic algorithms; covariance matrix adaptation evolution strategy; recurrent neural networks

MSC: 92-08

check for updates

Citation: Guo, Z.; Hou, D.; He, Q. Hybrid Genetic Algorithm and CMA-ES Optimization for RNN-Based Chemical Compound Classification. *Mathematics* **2024**, *12*, 1684. https://doi.org/10.3390/math12111684

Academic Editor: Ioannis G. Tsoulos

Received: 26 April 2024
Revised: 16 May 2024
Accepted: 20 May 2024
Published: 29 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Productivity in drug development is declining, in part due to inefficiencies in the early lead discovery process, which typically screens large numbers of compounds to identify potential leads for subsequent preclinical development [1]. Related reporting shows that the price of pharmaceutical research and development (R&D) has been rising steadily [2]. Due to several factors, including escalating clinical trial complexity, expanding regulatory requirements, and rising drug manufacturing costs, this trend is anticipated to continue. The use of machine learning techniques for chemical classification has been a topic of significant interest in recent years, as it has numerous applications in drug discovery [3]. Machine learning has the potential to drastically cut the time and expense associated with drug discovery and development, revolutionizing the pharmaceutical industry. It can assist in locating promising drug candidates and improving clinical trial design thanks to its capacity to analyze enormous amounts of data quickly and accurately [4].

The realm of cheminformatics has witnessed remarkable advancements in recent years, with increasing demand for efficient and accurate methods to analyze, predict, and classify chemical compounds. This phenomenon has been propelled by continuously expanding repositories of chemical structures and related data [5,6]. Consequently, the possibility has arisen for the development of pioneering computational techniques to accelerate the drug discovery process. However, this is currently a daunting task due to the complexity and variety of chemical structures of different drugs [7,8]. One common method of representing

chemical compounds is the Simplified Molecular Input Line Entry System (SMILES), which encodes the structure of a molecule as a string of characters. SMILES strings use a series of characters to represent atoms and bonds between them, allowing for easy storage and manipulation of chemical structures [9]. Table 1 shows the common SMILES codes and their corresponding interpretations. However, the inherent complexity and variability of SMILES strings pose significant challenges for classification tasks, as traditional machine learning methods often struggle to capture the intricate relationships between chemical structure and properties [10].

Table 1. The meaning of SMILES codes.

Feature (Atoms)	Description (Br, B, Cl, O,P, S, F…)
=	Double bond
#	Triple bond
$	Quadruple bond
.	Non bond
@	Chirality
+	Cation
-	Anion
:	Aromatic bond
()	Branching in compounds
[]	Atom group
Cyclic Structure	The total number of rings in SMILES code

There are a number of notable existing research gaps. Current chemical compound classification research encounters several prevailing trends and challenges. First, while several classification strategies [11–24] have been proposed, many encounter issues with low efficiency or low accuracy. This is particularly evident when handling large-scale datasets or complex chemical structures. Second, the availability of diverse chemical datasets has expanded, facilitating more comprehensive model training and evaluation. However, ensuring the quality and standardization of these datasets remains a challenge, impacting the reproducibility and comparability of research findings.

Evolutionary algorithms are a class of optimization techniques inspired by the principles of biological evolution, employing processes such as selection, crossover, and mutation to iteratively search for optimal solutions [11]. Metaheuristics, on the other hand, encompasses a broader category of optimization algorithms that are not constrained by specific problem structures [12], instead relying on randomness, heuristics, and adaptive mechanisms to guide the search process towards approximate optimal solutions. Recent advancements in metaheuristic algorithms have introduced innovative approaches such as the Giant Trevally Optimizer [13] and Artificial Rabbits Optimization [14]. The Giant Trevally Optimizer draws inspiration from the unique foraging behavior of the Giant Trevally fish, utilizing aggregation and dispersal strategies to adjust individual positions for optimal solution exploration. Conversely, the Artificial Rabbits Optimization algorithm mimics the population dynamics and migration patterns of rabbits, iteratively updating individual positions to navigate towards optimal solutions.

Nevertheless, the application of metaheuristic algorithms in chemical classification presents certain constraints. Typically, the use of metaheuristic algorithms may entail significant computational costs, especially when confronted with expansive solution spaces. This can result in prolonged training durations and heightened resource utilization [25,26]. In contrast, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) prioritizes local exploitation by dynamically adjusting the covariance matrix. This approach facilitates a more targeted exploration of promising candidate solutions [27,28]. While both methods have individually showcased their efficacy across diverse domains, their distinct advantages and synergistic potential imply that a hybridized approach could potentially deliver superior outcomes within the realm of RNN-based chemical compound classification. To overcome the above issues, in this paper we propose a novel chemical classification opti-

mization method for the drug discovery domain. The main contributions of this paper are summarized as follows:

(1) In the design of our algorithm, we take into account both global and local exploration capability to ensure a comprehensive ability to discover potential regions within a wide search space while mitigating the drawbacks of local optima and lengthy computation time.
(2) We design a unique hybrid optimization mechanism that combines the advantages of genetic algorithms and the CMA-ES for training recurrent neural networks (RNNs) to perform chemical compound classification from SMILES symbols task.

The rest of this paper is organized as follows. In Section 2, we review the current research into compound classification strategies. In Section 3, we devise a model formation problem. Section 3 proposes the primary materials and method used in the model. Section 4 shows the experimental design and experimental results. Finally, Section 6 concludes our research.

2. Related Work

Chemical similarity search plays a pivotal role in drug discovery [15]. Evaluating chemical similarity can involve considering the 3D structural features of compounds, such as the approach proposed by Rush, which utilizes Gaussian representation of molecular shape [16]. Rush's method, based on 3D shape similarity, facilitated the identification of two new inhibitor scaffolds for Wyeth's ZipA program. Additionally, Ferreira introduced a novel approach integrating semantic similarity with existing structural comparison methods for automatic classification of chemical compounds [17]. This integration has demonstrated feasibility and effectiveness in enhancing compound classification systems. Furthermore, Lo et al. developed a new approach utilizing cell cycle arresting patterns as molecular signatures to prioritize FDA-approved drugs with repurposing potential [18]. Subsequent 3D chemical structure similarity clustering has aided in identifying FDA-approved drugs that inadvertently induce DNA damage. The principle underlying chemical similarity assumes that compounds with similar structures may exhibit similar biological activities, forming the basis of similarity-based virtual screening [19]. However, it is essential to acknowledge that this assumption may not always hold.

Machine learning is currently one of the most important and rapidly evolving topics in computer-aided drug discovery. Poroikov designed a pass program which uses a variant of Naive Bayes to estimate drug activity based on query structures using the prior probability distribution [20]. In experiments, the authors found that the predictions made by machine learning remained reliable even when excluding as much as 60% of the information. Genetic Algorithms (GAs) use a metaheuristic technique that draws inspiration from the principles of genetics to search for valuable solutions to intricate problems. GAs have found wide applications in medical research [21]. Sarkate proposed KNN algorithms and genetic algorithms to improve the classification accuracy of drug and medical datasets [22]. Nevertheless, there is a lack of definitive approaches for determining the appropriate number of nearest neighbors, as excessively high or low values of k result in unfavorable false positive or false negative rates, respectively. Karakaplan [23] generalized molecular classification with data obtained from chemical databases and molecular docking by using a combination of deep learning and GA for use in drug design. In their paper, the authors explored the feasibility of using GAs to optimize DNNs, demonstrating that this combination can be used to classify data. RNNs are an important family of machine learning tools, and are is widely used in many fields. Feng et al. [24] utilized machine learning predictors to analyze inhibitor data for four major opioid receptors, combining gradient boosting decision tree algorithms with two NLP-based molecular fingerprints and one traditional 2D fingerprint. The resulting predictions enabled discrimination of DrugBank compounds based on binding affinity thresholds and selectivities for different receptors. These predictions were further analyzed for ADMET properties, guiding repurposing efforts for DrugBank compounds targeting specific opioid receptors.

3. Materials and Methods

In this part, we introduce the framework of our Hybrid Genetic Algorithm and CMA-ES Optimization for RNN-based Chemical Compound Classification (GA-CMA-ES-RNN) method, providing a detailed description of each part of the proposed algorithm. The main parameters and notations used in this paper are shown in Table 2.

Table 2. Notation.

Notations	Description
G	original molecular graph in 3D space
G'	projected molecular graph in 2D space
P_t	population at generation t
$\mathbf{p}_i$	i-th candidate solution in the population
$\|M_t\|$	size of the mating pool
m	mean vector
C	covariance matrix
s_s	step size
p_m	mutation probability
t	time step
h_t	hidden state at time t
W_{hh}	weight matrix connecting the hidden state to itself
W_{xh}	weight matrix connecting the input to the hidden state
b_h	bias term of the hidden state
σ	activation function
c_c	learning rate for updating the covariance matrix
W_{hy}	weight matrix connecting the hidden state to the output
b_y	bias output term

3.1. Data Collection and Preprocessing

In this article, our data were taken from three datasets: Protein Data Bank (PDB) [29], ChemPDB [30], and the Macromolecular Structure Database (MSD) [31]. Each of these databases contains a vast amount of chemical and biochemical information, including structural data represented in Simplified Molecular Input Line Entry System (SMILES) notation. The SMILES strings were processed to remove irrelevant atoms and bonds, normalize the molecular graphs, and construct adjacency matrices. These preprocessed data were then used as input to our neural network for training and classification. By filtering the data based on various criteria and using high-quality data from these databases, we obtained a dataset of chemical compounds from various datasets which contained 2500 compounds along with their respective labels. The dataset was preprocessed using a genetic algorithm-based approach to generate diverse and high-quality samples of the chemical compounds, which were used as input to our neural network for training and classification.

3.2. Sample SMILES String Representation

The conversion of the 3D molecular structure of Arglecin into its 2D representation and subsequent SMILES code can be described mathematically using the concept of graph theory [32]. The 3D molecular structure of Arglecin can be represented as a graph, wherein atoms are represented by nodes and chemical bonds between atoms are represented by edges. Each atom is assigned a unique atomic number, and each bond is assigned a bond order that represents the number of electron pairs shared between the bonded atoms. To convert the 3D molecular structure of Arglecin into a 2D representation, the process of projection is used; this involves projecting the 3D structure onto the 2D plane while preserving the topology of the molecular graph. The projection is achieved by selecting

a suitable plane of projection which intersects the molecule in a way that preserves the connectivity of the atoms and bonds. With this 2D projection, the SMILES code for the molecule can be generated using a set of rules based on the connectivity of the atoms and the type of bonds between them. The SMILES code is a linear representation of the 2D molecular structure in which atoms are represented by their elemental symbols and bonds are represented by a set of symbols indicating their type and orientation. The conversion from the 3D to the 2D representation of Arglecin can be mathematically expressed as a projection function P which maps the 3D molecular graph G to a 2D projection graph G'. The projection function is defined as follows:

$$P(G) = G'. \tag{1}$$

The projection function can be represented mathematically using linear algebra, with the 3D coordinates of the atoms in the molecular graph transformed into 2D coordinates in the projected graph. In summary, the conversion of the 3D molecular structure of Arglecin into its 2D representation and SMILES code involves the use of projection techniques and a set of rules based on the connectivity of the atoms and the type of bonds between them (Figure 1). The conversion can be mathematically expressed using graph theory, linear algebra, and set theory. This process of converting SMILES data into a molecular graph and then into an adjacency matrix is critical for the accurate and efficient training of the RNN, as it allows the neural network to learn meaningful patterns in the chemical space. The process is illustrated below in Figure 2. By representing chemical compounds as graphs and matrices, it is possible to effectively leverage the power of graph theory and deep learning to solve complex problems in cheminformatics.

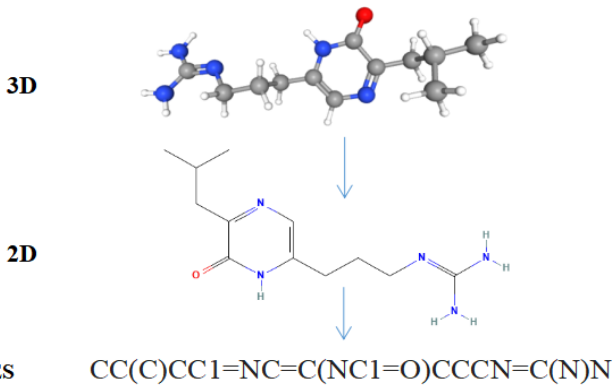


Figure 1. The conversion of compounds in three planes. The figure illustrates the conversion process of compounds represented in three planes: 3D, 2D, and SMILES notation.

3.3. Data Augmentation Techniques

Data augmentation plays a crucial role in enhancing the performance of machine learning models, particularly in the context of this thesis, which focuses on the design and implementation of a hybrid Genetic Algorithm and CMA-ES optimization for Recurrent Neural Network-based classification of chemical compounds from SMILES notation. Augmenting the dataset increases its diversity, helps to improve the generalization capabilities of the model, and prevents overfitting. The following sections provide a detailed account of the data augmentation process.

In the first step of data augmentation, rotatable bonds within the molecules are identified. By identifying rotatable bonds, the algorithm can generate alternative SMILES strings representing distinct conformations of the same molecule. The resulting SMILES strings correspond to unique conformations of the compound while retaining the same molecular identity. Generating these alternative SMILES strings increases the diversity of the dataset, which in turn aids the model in capturing the structural variability of chemical compounds.

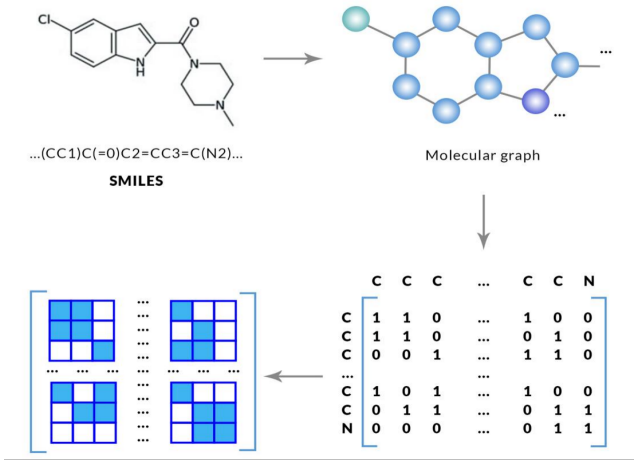


Figure 2. Molecular adjency matrix from SMILES notation. The figure presents the molecular adjacency matrix derived from SMILES (Simplified Molecular Input Line Entry System) notation. Each element in the matrix represents the connectivity between atoms in the molecular structure as encoded by the SMILES string.

After generating a pool of augmented SMILES strings for each molecule, the next step is to select a subset of these strings to include in the dataset. The selection process can be performed randomly or based on specific criteria, such as maximizing diversity or novelty among the generated strings. To ensure that the model is exposed to a wide range of molecular conformations and stereochemistries during training, the chosen augmented data should be a representative sample of the structural variability within the compound space. By incorporating this augmented dataset into the training process, the model is better equipped to handle the classification of chemical compounds from SMILES notation.

3.4. Genetic Algorithm

In this study, we utilize the baseline GA for optimizing the hyperparameters of the recurrent neural network (RNN) model used for chemical classification, which processes preprocessed SMILES data in the form of molecular graphs and adjacency matrices. By employing the baseline GA in conjunction with the RNN model, we aim to address the challenges associated with optimizing complex machine learning models and improving their accuracy and generalization performance. The ability of the GA to efficiently search large solution spaces and find near-optimal solutions is expected to enhance the performance of the RNN model in chemical classification tasks. Furthermore, the data augmentation techniques described in the previous chapter, such as rotatable bond identification, alternative SMILES string generation, stereoisomer enumeration, randomization of atom order, and selection of representative augmented data, all contribute to increasing the diversity and comprehensiveness of the dataset. This in turn helps to improve the performance and generalization capabilities of the RNN-based classification model.

- The selection process in the genetic algorithm is a critical component that directs the search towards high-performing candidate solutions. In this study, we employ a sophisticated selection mechanism called tournament selection with elitism. Here,

$$P_t = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}, \tag{2}$$

  where $\mathbf{p}_i$ is a vector of the RNN hyperparameters and weight initialization. The tournament selection process can be described as follows:

  1. For each selection step, choose a random subset $T \subseteq P_t$ of size k (the tournament size):

$$T = \{\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \ldots, \mathbf{p}_{i_k}\}. \tag{3}$$

2. Determine the candidate with the highest fitness value in the tournament:

$$\mathbf{p}_{winner} = \arg\max_{\mathbf{p}_i \in T} fitness(\mathbf{p}_i). \tag{4}$$

3. Add the winner to the mating pool M_t for the next generation:

$$M_t = M_t \cup \{\mathbf{p}_{winner}\}. \tag{5}$$

This process is repeated until $|M_t| = N$. In summary, the selection process in the genetic algorithm (specifically, tournament selection) serves as a mechanism to bias the search towards better-performing candidate solutions.

- Crossover is responsible for generating new candidate solutions by combining the features of selected parent solutions [33]. The crossover mechanism generates offspring by recombining the features of two parent solutions. This process encourages the exploration of new regions in the search space while retaining the promising characteristics of the selected parents. The overall process is shown in Figure 3. In summary, the crossover process in the genetic algorithm (specifically, uniform crossover) is crucial for generating new candidate solutions by recombining the features of selected parent solutions.

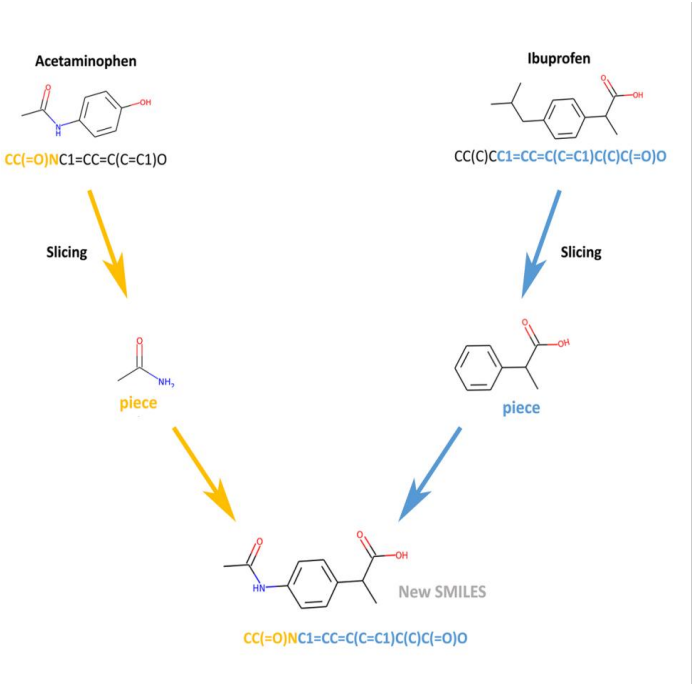


Figure 3. Crossover operator process. This figure illustrates the process of the crossover operator in the genetic algorithm. Each curve represents a different crossover strategy, with details provided in the legend. Crossover can generate new candidate solutions by combining the features of selected parent solutions (i.e., Acetaminophen and Ibuprofen).

- Mutation is another essential step in the genetic algorithm, responsible for introducing small random changes to the candidate solutions. The mutation mechanism introduces small random changes to the candidate solutions, allowing the search to explore new areas in the solution space and helping to maintain diversity in the population. In this paper, we use a common mutation mechanism called Gaussian mutation. The Gaussian mutation process can be described as follows:

1. For each offspring solution $\mathbf{p}_i \in P_{t+1}$, iterate through each element: $j = 1, 2, \ldots, d$.

2. We apply the following Gaussian perturbation to the element.

$$\mathbf{p}_i(j) = \begin{cases} \mathbf{p}_i(j) + N(0, s_s^2), & \textit{with probability } p_m \\ \mathbf{p}_i(j), & \textit{with probability } 1 - p_m \end{cases} \quad (6)$$

- Lastly, by implementing a parallel GA with an island model approach, we can leverage the benefits of parallelization to enhance the performance and efficiency of the optimization process. This enables the GA to effectively explore the vast and complex search space of RNN hyperparameter configurations, ultimately improving the classification accuracy and generalization capabilities of the resulting models for chemical compound categorization using SMILES notation.

3.5. RNN Architecture

The Recurrent Neural Network (RNN) architecture employed in this study is designed to classify chemical compounds from their preprocessed SMILES notation, as shown in Figure 4. An RNN was chosen over other algorithms, such as feedforward neural networks and convolutional neural networks (CNNs), due to the ability of RNNs to handle variable-length sequential data, which makes them particularly suitable for processing SMILES strings [34,35].

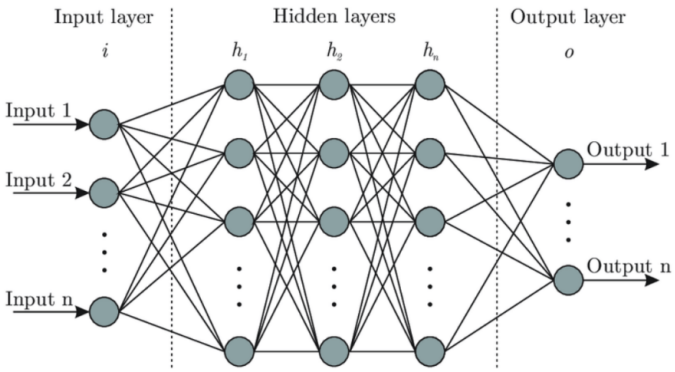


Figure 4. Architecture of the RNN network. The different components and layers of the RNN, including the input, hidden, and output layers, are labeled and described in the legend for clarity.

Each input node corresponds to a specific feature of the molecular representation, enabling the RNN to process the sequential data and learn the underlying patterns in the molecular structures. This design choice allows the RNN to effectively capture the spatial and temporal relationships between atoms within the chemical compounds [36]. We denote the input at time step t as x_t.

The hidden layers in the RNN consist of a set of recurrent units, which maintain internal states to capture temporal dependencies in the input data. These internal states allow the RNN to learn and recognize patterns in the sequential data, ultimately enabling accurate chemical compound classification. The GA optimizes the number of hidden layers and the number of recurrent units in each layer to improve the model's ability to capture complex relationships in the input data. The hidden state at time step t, denoted by h_t, is computed as follows:

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h). \quad (7)$$

The output layer of the RNN produces the final classification probabilities for each chemical compound [37]. Here, the GA is responsible for optimizing the output layer's activation function and other relevant hyperparameters to achieve the best classification performance. The predicted output at time step t, denoted by $\hat{y}_t$, is computed using the softmax function, as follows:

$$\hat{y}_t = softmax(W_{hy}h_t + b_y). \quad (8)$$

3.6. CMA-ES Optimization

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is a powerful optimization technique that has been applied to various complex optimization problems, including those involving nonlinear, non-convex, or noisy objective functions [38]. Thus, we integrate CMA-ES with our baseline Genetic Algorithm (GA) and Recurrent Neural Network (RNN) system to further improve the performance of the compound classification model. The integration of CMA-ES into our hybrid GA-RNN system aims to fine-tune the RNN model's hyperparameters and weight initializations, complementing the role of the GA. This combined approach leverages the global search capabilities of GA and the local search capabilities of CMA-ES, ensuring a thorough exploration of the solution space and better convergence towards optimal solutions. The key steps of the CMA-ES algorithm are as follows:

1. Initialize the mean vector m, covariance matrix C, and step size s_s.
2. For each generation, we have the following process:
   (a). Sample λ candidate solutions x_i from a multivariate normal distribution with mean m and covariance matrix:

$$\mathrm{x}_i \sim N\left(\mathrm{m}, s_s^2 \mathrm{C}\right). \tag{9}$$

   (b). Evaluate the fitness of each candidate solution using the predefined fitness function.
   (c). Select the μ best candidate solutions based on their fitness values.
   (d). Update the mean vector m using the weighted average of the selected candidate solutions:

$$\mathrm{m}_{t+1} = \mathrm{m}_t + s_s \sum_{i=1}^{\mu} w_i (\mathrm{x}_i - \mathrm{m}_t) \tag{10}$$

   where w_i are the weights associated with the selected candidate solutions.
   (e). Update the covariance matrix C using the weighted outer product of the selected candidate solutions:

$$\mathrm{C}_{t+1} = (1 - c_c)\mathrm{C}_t + c_c \sum_{i=1}^{\mu} w_i (\mathrm{x}_i - \mathrm{m}_t)(\mathrm{x}_i - \mathrm{m}_t)^T. \tag{11}$$

   (f). Update the step size s_s using a suitable adaptation rule.
3. Repeat steps 2(a) to 2(f) until a termination criterion is met.

3.7. Implementation in the Hybrid Algorithm

The application of CMA-ES optimization to our RNN model involves refining the hyperparameter search by adaptively updating the search distribution based on the fitness landscape. Unlike the GA, which relies on mutation and crossover operators to explore the solution space, the CMA-ES updates the covariance matrix to control the direction and magnitude of the search steps. First, we apply the GA to explore the solution space and identify promising candidate solutions. These solutions serve as initial points for the subsequent CMA-ES optimization. Next, we apply the CMA-ES to fine-tune the hyperparameters (e.g., the mean vector m, covariance matrix C, and step size s_s) and weight initializations of the RNN model, guiding the search towards optimal solutions based on the updated covariance matrix. This process iteratively refines the search distribution, ultimately leading to improved model performance [39]. The flowchart of hybrid algorithm is illustrated in Figure 5.

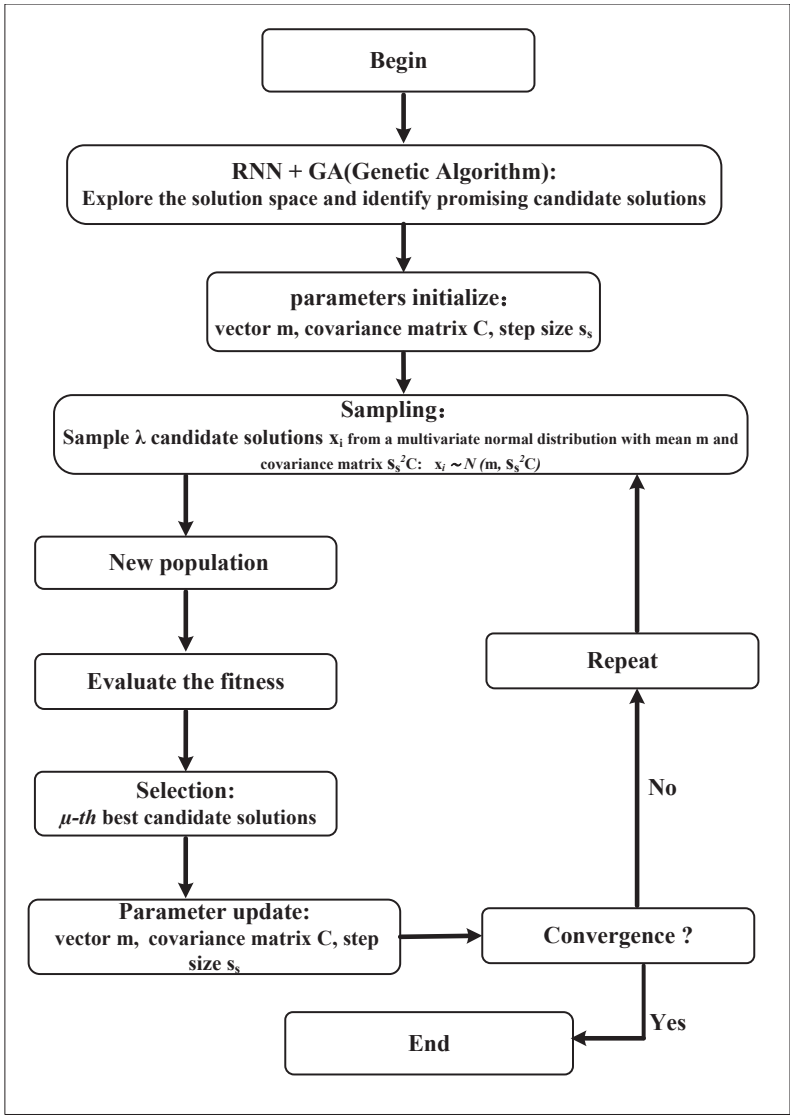


Figure 5. Flowchart of te hybrid algorithm. Here, the GA is employed to explore and identify promising solutions, which then serve as initial points for subsequent CMA-ES optimization to refine the hyperparameter and RNN model weight initializations, iteratively enhancing model performance by updating the search distribution.

4. Experimental Results

This section provides the experimental settings, then compares the results of different parameters and different algorithms from two performance indexes, namely, accuracy and running time.

4.1. Experimental Settings

We used a dataset of chemical compounds from various datasets containing 2500 compounds along with their respective labels. The dataset was preprocessed using a genetic algorithm-based approach to generate diverse and high-quality samples of the chemical compounds. The generated compounds were filtered based on the criteria, including their molecular weight, number of atoms, and solubility, to ensure that only high-quality compounds were used for training and evaluation. The list of databases used included the Protein Data Bank (PDB) [29], ChemPDB [30], and Macromolecular Structure Database (MSD) [31]. Each of these databases contains a vast amount of chemical and biochemical information, including structural data represented in Simplified Molecular Input Line Entry System (SMILES) notation.

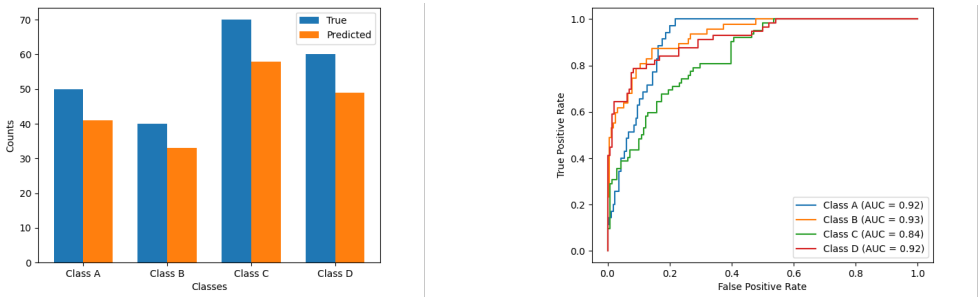
Experiments were carried out in the PyCharm Community environment on a PC with an Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz, 3.41 GHz processor, and 8.00 GB RAM. The hyperparameters of the algorithm were tuned using a grid search. We set the population size to 100–1000, the number of generations to 50, the mutation rate to 0.1, and the learning rate to 0.001.

4.2. *Experimental Results in Different Aspects*

In this section, we present the results of our research. The proposed Hybrid Genetic Algorithm and CMA-ES Optimization for Recurrent Neural Network (GA-CMA-ES-RNN) was applied to classify chemical compounds into four distinct classes: Class A, Class B, Class C, and Class D.

4.2.1. The Case of Experimental Classification

In Figure 6a, the algorithm demonstrates an overall accuracy of approximately 83% on average across all four classes. This level of accuracy is a strong indication of the successful integration of the genetic algorithm and CMA-ES optimization with the RNN for the classification task. The hybrid algorithm's performance benefits from the combination of the genetic algorithm's ability to explore the solution space and the CMA-ES optimization's ability to fine-tune the hyperparameters, leading to improved classification results. In Figure 6b, the ROC curves demonstrate the algorithm's ability to effectively discriminate between the classes. The high AUC scores indicate that the algorithm can correctly classify a large proportion of the compounds while minimizing the number of false positive classifications. It should be noted that the AUC scores may be higher than average due to the smaller dataset size used in this study. The algorithm's performance could potentially improve with larger datasets and higher computational resources.

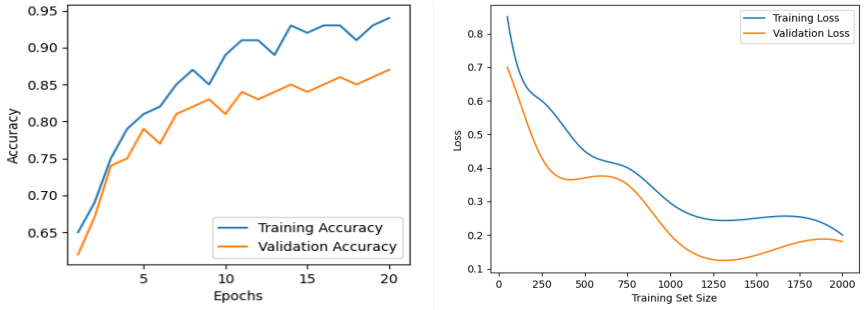


(**a**) Molecular Classification Distribution (**b**) Multiclass ROC curve

Figure 6. Multi-classification experiment results: (**a**) depicts the true and predicted values for the classification of chemical compounds, while (**b**) illustrates the ROC curves for each of the four classes (Class A, Class B, Class C, and Class D).

4.2.2. Experimental Results under Different Parameters

In Figure 7a, the curves show that the validation accuracy closely follows the training accuracy, indicating that the hybrid algorithm generalizes well to unseen data. This is a crucial aspect of the algorithm's performance, as it suggests that the model is not overfitting and has effectively learned the underlying patterns within the data. The RNN's capacity to model complex sequential data in combination with the genetic algorithm and CMA-ES optimization contributes to the high validation accuracy (Lipton et al., 2015). In Figure 7b, the curve demonstrates a smooth and steady decrease in loss as the size of the training set increases. This result confirms the effectiveness of the hybrid algorithm in learning the classification task and adjusting the model's parameters accordingly. In addition, the learning curve suggests that the model benefits from additional training data, which may further improve the algorithm's performance.



(**a**) Training/Validation Accuracy (**b**) Training/Validation Learning curve

Figure 7. Experimental results under different parameters: (**a**) presents the training and validation accuracy curves, while (**b**) presents the learning curve, showing the algorithm's progress in learning from the dataset.

In Figure 8, the presented 3D relationship graph serves as a comprehensive visualization tool, elucidating the intricate interdependencies among three fundamental parameters within the domain of the GA-CMA-ES-RNN algorithm: GA population size, CMA-ES step size, and accuracy. The parameter on the x-axis signifies the number of candidate solutions explored within the genetic algorithm framework. Larger population sizes inherently promote more extensive exploration of the solution space, potentially leading to the discovery of superior solutions. Meanwhile, the parameter on the y-axis critically influences the algorithm's adaptability and convergence rate towards optimal solutions. Optimal step sizes strike a delicate balance between exploration and exploitation, exerting a profound impact on the algorithm's efficacy in navigating the solution landscape. The accuracy values span from 80.5% to 84.6%, with warmer colors indicating higher accuracy levels. This color-coded representation facilitates intuitive interpretation of the data, enabling the identification of optimal parameter configurations conducive to maximizing accuracy. Upon analysis of the graph, a discernible pattern emerges, revealing an optimal combination of GA population size and CMA-ES step size conducive to achieving peak accuracy. Notably, configurations featuring a moderate CMA-ES step size (approximately 0.3) coupled with an increased GA population size (200) manifest superior performance, as evidenced by the prevalence of warmer-colored data points. This configuration yields a notable accuracy of 83.1%. Furthermore, the graph illustrates that extremes in CMA-ES step sizes tend to result in diminished accuracy. For instance, a GA population size of 100 paired with a CMA-ES step size of 0.4 yields an accuracy of 81.2%. This observation underscores the importance of maintaining a balance between exploration and exploitation to optimize algorithmic performance. An optimal CMA-ES step size facilitates adequate exploration of the search space while mitigating the risk of convergence towards local optima.

4.2.3. Comprehensive Comparison under Various Parameters

The GA population size on the left axis refers to the number of candidate solutions in the genetic algorithm. As the population size increases, the exploration of the search space is expected to improve. The CMA-ES step size is an essential parameter in the CMA-ES optimization process, controlling the rate of adaptation and convergence towards an optimal solution. The accuracy, precision, recall, and F1-score are standard performance metrics used to evaluate the classification performance of algorithms. The color of the lines is determined by the accuracy, with the Tealrose color scale indicating higher accuracy with a more vibrant shade. From the graph, it can be observed that the highest accuracy (84.6%) is achieved with a GA population size of 150 and a moderate CMA-ES step size of 0.3. This specific combination of parameters also results in high precision (84.1%), recall (85.2%), and F1-score (84.5%), indicating well-rounded performance of the algorithm. In addition, the graph shows that the performance decreases when the CMA-ES step size is either too small or too large, suggesting that there exists an optimal step size that balances the exploration and exploitation trade-off in the optimization process (Figure 9).

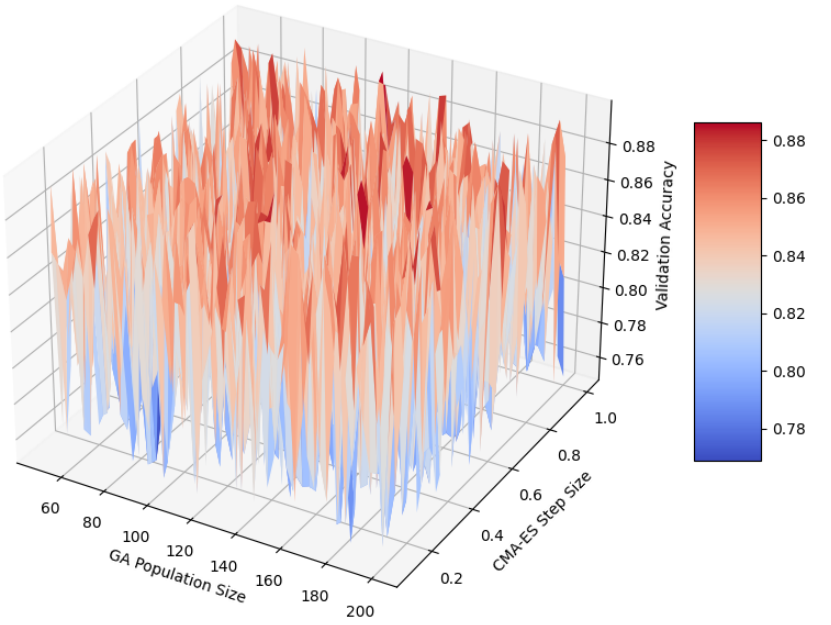


Figure 8. The 3D relationship between GA-CMA-ES parameters: the x-axis of the graph delineates the GA population size, discretized into values of 100, 150, and 200; the y-axis corresponds to the CMA-ES step size, featuring values of 0.2, 0.3, 0.4, and 0.5; and the z-axis represents accuracy, quantifying the overall classification performance of the algorithm.

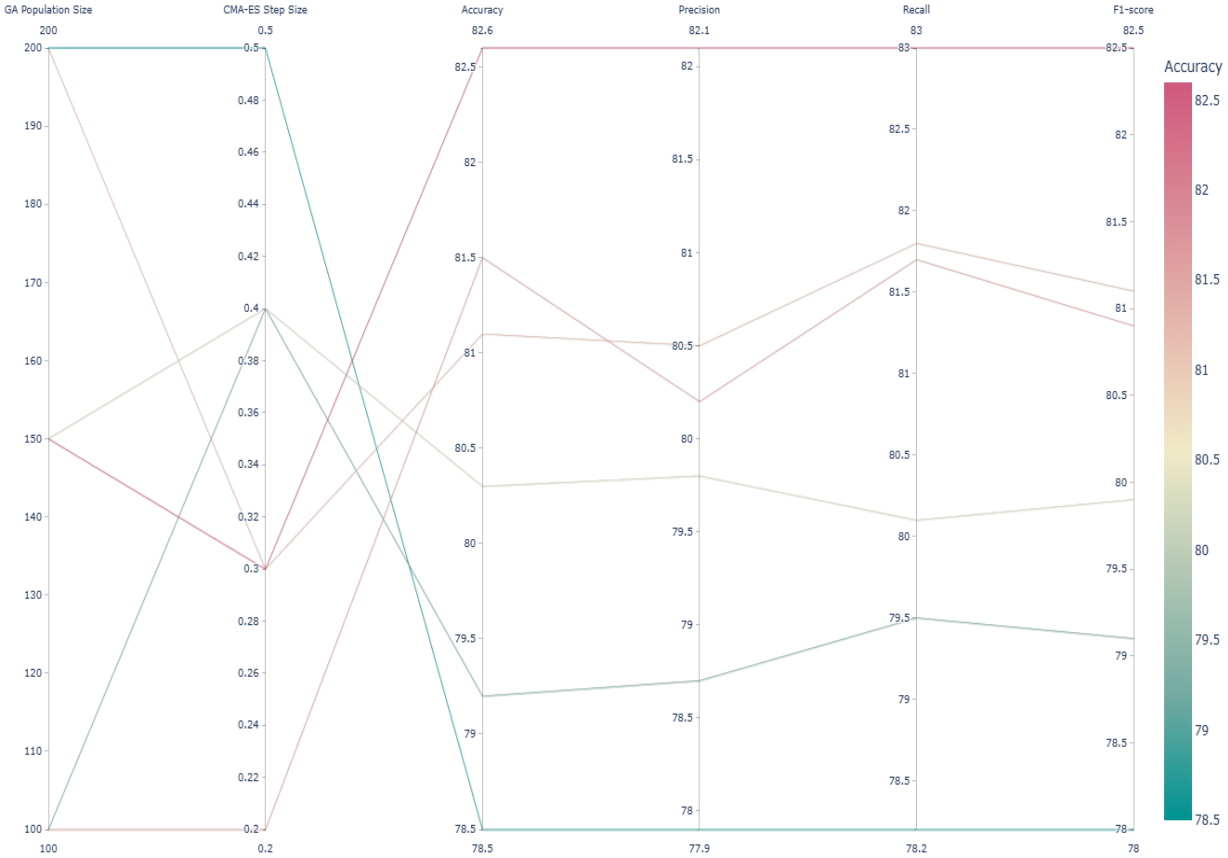


Figure 9. Comprehensive view of the various parameters, showing the relationship between various parameters and the performance of the GA-CMA-ES-RNN algorithm. The investigated parameters are the GA population size, CMA-ES step size, accuracy, precision, recall, and F1-score. Each line in the parallel coordinates plot represents a specific combination of these parameters.

4.3. Comparison with Other Methods

In this section, we present different comparison results for the classification of complex organic compounds using various algorithms, including the proposed hybrid CMA-ES algorithm.

4.3.1. Accuracy Comparison

The results are summarized below. From Figure 10, it is evident that the hybrid CMA-ES algorithm achieves the highest accuracy rate of 83.84%, outperforming all other algorithms in this comparison. This demonstrates the effectiveness of the hybrid genetic algorithm and CMA-ES optimization in improving the classification performance of Recurrent Neural Networks for chemical compounds from SMILES notation. The test results were obtained using a stratified k-fold cross-validation method, which involves splitting the dataset into ten equally sized subsets (or "folds"). The algorithm was then trained on k-1 folds and tested on the remaining fold. This process was repeated nine times, with each fold being used as a test set exactly once. The average accuracy across all nine iterations is reported as the overall accuracy of the algorithm. The stratified k-fold cross-validation ensures that the distribution of the classes in the dataset is preserved in each fold, providing a more reliable estimate of the algorithm's performance. To achieve these results, the hybrid CMA-ES algorithm was optimized by fine-tuning its hyperparameters through a grid search, which involved iteratively exploring different combinations of hyperparameter values to identify the configuration that yielded the best classification performance. The RNN architecture was also optimized by adjusting the number of hidden layers, neurons per layer, and activation functions. The superior performance of the hybrid CMA-ES algorithm can be attributed to the combined strengths of the genetic algorithm and CMA-ES optimization.

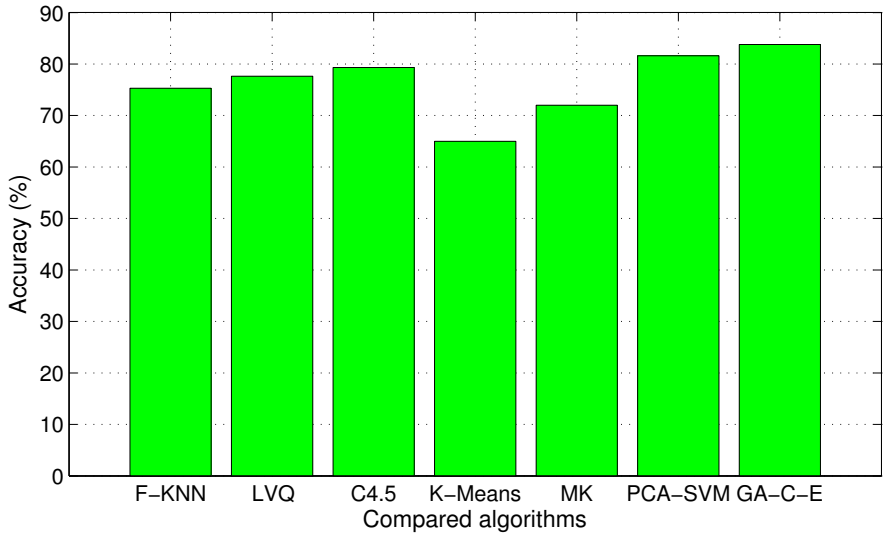
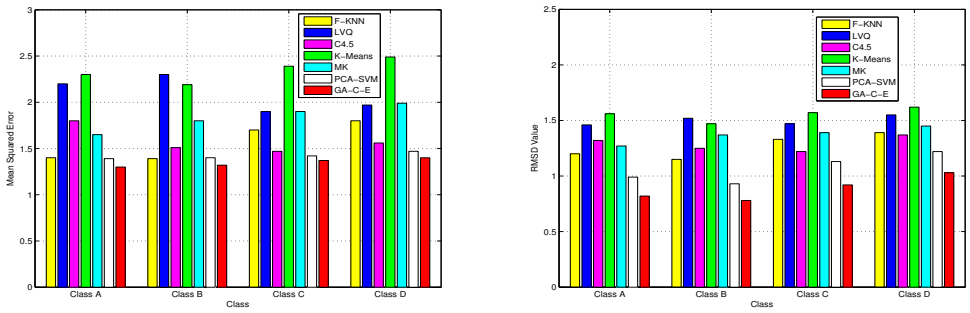


Figure 10. Accuracy scores of classification models. The accuracy of each algorithm was evaluated by measuring the percentage of correctly classified compounds out of the total number of test compounds. The algorithms compared in this study were the Fuzzy KNN (F-KNN) method [40], Learning Vector Quantization (LVQ) [41], C4.5 method [42], K-Means clustering method [43], momentum backpropagation (MK) method [44], PCA-SVM [45], and hybrid CMA-ES (GA-C-E) algorithm.

4.3.2. RMSD and MSE Comparison

Figure 11a shows that the proposed GA-CMA-ES-RNN algorithm consistently outperforms the other algorithms, achieving lower RMSD values in all classes, which indicates higher accuracy in predicting molecular compound classifications. From Figure 11b, it can be seen that our algorithm yields lower MSE values across all classes. This further validates

the effectiveness of the GA-CMA-ES-RNN algorithm in classifying molecular compounds with higher accuracy compared to the other methods.



(**a**) RMSD Comparison (**b**) MSE Comparison

Figure 11. Comparison of experimental results for different algorithms. (**a**) presents a comparison of the Root Mean Squared Deviation (RMSD) [46] and Mean Squared Error (MSE) [47] for the other five algorithms (Fuzzy KNN, LVQ, C4.5, K-means, and Backpropagation) across the four datasets (Class A, Class B, Class C, and Class D).

4.3.3. Run-Time Performance Evaluation

As depicted in Figure 12, the CMA-ES and K-means algorithms consistently outperform the other five algorithms in terms of runtime efficiency. This signifies the superior efficiency of our method. Notably, the hybrid CMA-ES algorithm demonstrates competitive performance overall, underscoring its potential as a viable solution for chemical compound classification from SMILES notation. These findings highlight the promising prospects of our hybrid approach in enhancing the efficiency of classification tasks in chemical compound analysis.

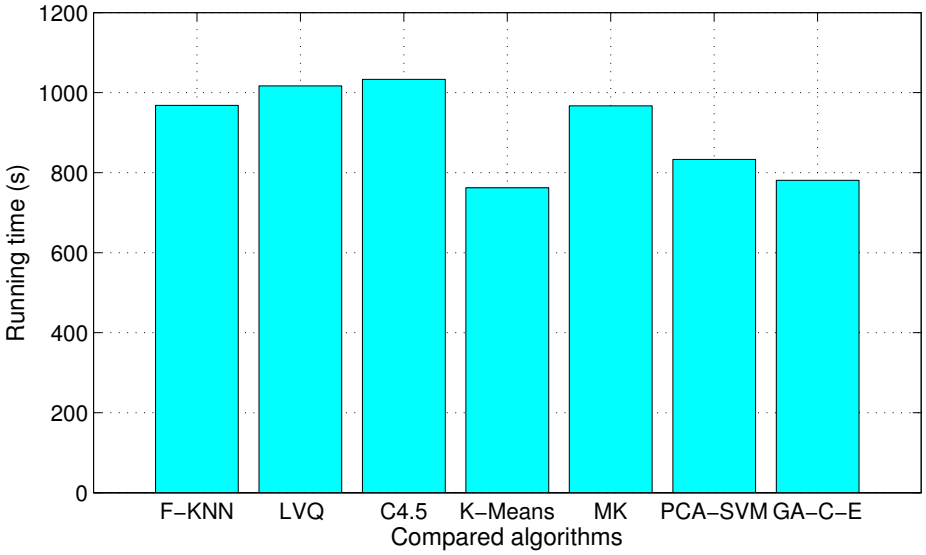


Figure 12. Results of the runtime performance evaluation of all algorithms for classifying chemical compounds from SMILES notation, focusing on a dataset comprising 100 compounds.

5. Discussion

In our simulation experiment, the classification accuracy soared to 83%, a marked improvement over the baseline method. This underscores the efficacy of our hybrid approach merging a Genetic Algorithm (GA) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES). First, the hybrid utilization of GA and CMA-ES enables a more comprehensive exploration of the solution space, allowing for the identification of optimal

solutions with higher precision. The GA excels in global exploration, while the CMA-ES focuses on local exploitation, synergistically improving the search process. Additionally, the adaptive nature of the CMA-ES dynamically adjusts the covariance matrix to prioritize promising candidate solutions, enhancing the efficiency of the optimization process. Not only does this result validate the potency of our method, it highlights the power of machine learning in refining prediction accuracy amidst intricate datasets. The implications of our findings are profound, particularly in the realm of drug research, where they expedite the drug discovery process. Our hybrid strategy coupling an RNN, GA, and CMA-ES offers a promising avenue for bolstering the efficiency and precision of molecular classification. By harnessing the synergies between these techniques, our approach navigates the diversity and complexity of molecular chemical space more adeptly than conventional methods. This discussion section will delve into a deeper theoretical justification, elucidating why the GA-CMA-ES hybrid is well-suited for our task and how it surpasses existing methodologies.

6. Conclusions

In conclusion, our novel GA-CMA-ES-RNN molecular classification algorithm presents significant advancements over traditional machine learning methods in terms of efficiency, accuracy, and adaptability. Through a synergistic fusion of a Genetic Algorithm (GA) and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), our approach adeptly captures the intricate nonlinear relationships inherent within SMILES data. This integration harnesses the collective power of GAs' global exploration capability and the local exploitation mechanisms of the CMA-ES. The GA facilitates the exploration of diverse solution spaces, while the CMA-ES dynamically adjusts the covariance matrix to focus on promising candidate solutions. By leveraging these complementary strategies, our approach efficiently navigates the complex landscape of molecular data, ensuring robust modeling of intricate relationships for enhanced classification accuracy. Compared to conventional methods such as Fuzzy KNN, LVQ, C4.5, K-means clustering, and backpropagation, our method demonstrates superior performance in molecular classification. The versatility of our algorithm extends its potential applications across various domains within pharmacology and drug discovery, where precise molecular classification is imperative for advancing research and development endeavors.

Author Contributions: Methodology, Z.G., D.H. and Q.H.; Software, Z.G.; Formal analysis, Q.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 62202089; the National Natural Science Foundation of China under Grant U22A2004; the General Project of Liaoning Provincial Department of Education (No. LJKZ0005); the Doctor Startup Foundation of Liaoning Province under Grant No. 2021-BS-055; and the China Postdoctoral Science Foundation under Grant No. 2021M693318.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: Author Dianlong Hou was employed by the company Dongying United Petroleum & Chemicals Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The Dongying United Petroleum & Chemicals Co., Ltd. had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Pognan, F.; Beilmann, M.; Boonen, H.; Czich, A.; Dear, G.; Hewitt, P.; Mow, T.; Oinonen, T.; Roth, A.; Steger-Hartmann, T.; et al. The evolving role of investigative toxicology in the pharmaceutical industry. *Nat. Rev. Drug Discov.* **2023**, *22*, 317–335. [CrossRef]
2. Schuhmacher, A.; Gassmann, O.; Hinder, M. Changing R&D models in research-based pharmaceutical companies. *J. Transl. Med.* **2016**, *14*, 105. [PubMed]

3. Muhammed, M.T.; Aki-Yalcin, E. Molecular docking: Principles, advances, and its applications in drug discovery. *Lett. Drug Des. Discov.* **2024**, *21*, 480–495. [CrossRef]
4. Lin, X.; Li, X.; Lin, X. A review on applications of computational methods in drug screening and design. *Molecules* **2020**, *25*, 1375. [CrossRef]
5. Chugh, H.; Singh, S. Machine learning applications in rational drug discovery. *Drug Des. Using Mach. Learn.* **2022**, 97–116.
6. He, Q.; Qiao, W.; Fang, H.; Bao, Y. Improving the identification of miRNA–disease associations with multi-task learning on gene–disease networks. *Briefings Bioinform.* **2023**, *24*, bbad203. [CrossRef]
7. Bajorath, J.; Chávez-Hernández, A.L.; Duran-Frigola, M.; Fernández-de Gortari, E.; Gasteiger, J.; López-López, E.; Maggiora, G.M.; Medina-Franco, J.L.; Méndez-Lucio, O.; Mestres, J.; et al. Chemoinformatics and artificial intelligence colloquium: Progress and challenges in developing bioactive compounds. *J. Cheminform.* **2022**, *14*, 82. [CrossRef] [PubMed]
8. Newman, D.J.; Cragg, G.M. Natural products as sources of new drugs over the nearly four decades from 01/1981 to 09/2019. *J. Nat. Prod.* **2020**, *83*, 770–803. [CrossRef] [PubMed]
9. Islam, M.A.; Pillay, T.S. Simplified molecular input line entry system-based descriptors in QSAR modeling for HIV-protease inhibitors. *Chemom. Intell. Lab. Syst.* **2016**, *153*, 67–74. [CrossRef]
10. Šegota, S.B.; Anđelić, N.; Lorencin, I.; Musulin, J.; Štifanić, D.; Car, Z. Preparation of simplified molecular input line entry system notation datasets for use in convolutional neural networks. In Proceedings of the 2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE), IEEE, Kragujevac, Serbia, 25–27 October 2021; pp. 1–6.
11. Slowik, A.; Kwasnicka, H. Evolutionary algorithms and their applications to engineering problems. *Neural Comput. Appl.* **2020**, *32*, 12363–12379. [CrossRef]
12. Alorf, A. A survey of recently developed metaheuristics and their comparative analysis. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105622. [CrossRef]
13. Sadeeq, H.T.; Abdulazeez, A.M. Giant trevally optimizer (GTO): A novel metaheuristic algorithm for global optimization and challenging engineering problems. *IEEE Access* **2022**, *10*, 121615–121640. [CrossRef]
14. Wang, L.; Cao, Q.; Zhang, Z.; Mirjalili, S.; Zhao, W. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105082. [CrossRef]
15. Sadybekov, A.V.; Katritch, V. Computational approaches streamlining drug discovery. *Nature* **2023**, *616*, 673–685. [CrossRef]
16. Rush, T.S.; Grant, J.A.; Mosyak, L.; Nicholls, A. A shape-based 3-D scaffold hopping method and its application to a bacterial protein-protein interaction. *J. Med. Chem.* **2005**, *48*, 1489–1495. [CrossRef] [PubMed]
17. Ferreira, J.D.; Couto, F.M. Semantic similarity for automatic classification of chemical compounds. *PLoS Comput. Biol.* **2010**, *6*, E1000937. [CrossRef]
18. Lo, Y.C.; Senese, S.; France, B.; Gholkar, A.A.; Damoiseaux, R.; Torres, J.Z. Computational cell cycle profiling of cancer cells for prioritizing FDA-approved drugs with repurposing potential. *Sci. Rep.* **2017**, *7*, 11261. [CrossRef]
19. Bajorath, J. Molecular similarity concepts for informatics applications. In *Bioinformatics: Volume II: Structure, Function, and Applications*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 231–245.
20. Poroikov, V.V.; Filimonov, D.A.; Borodina, Y.V.; Lagunin, A.A.; Kos, A. Robustness of biological activity spectra predicting by computer program PASS for noncongeneric sets of chemical compounds. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1349–1355. [CrossRef]
21. Pang, X.; Ge, Y.F.; Wang, K.; Traina, A.J.; Wang, H. Patient assignment optimization in cloud healthcare systems: A distributed genetic algorithm. *Health Inf. Sci. Syst.* **2023**, *11*, 30. [CrossRef]
22. Sarkate, P.A.; Deorankar, A.V. Classification of chemical medicine or drug using K nearest neighbor (KNN) and genetic algorithm. *Int. Res. J. Eng. Technol.* **2018**, *5*, 833–834.
23. Karakaplan, M.; Avcu, F.M. Classification of some chemical drugs by genetic algorithm and deep neural network hybrid method. *Concurr. Comput. Pract. Exp.* **2021**, *33*, E6242. [CrossRef]
24. Feng, H.; Jiang, J.; Wei, G.W. Machine-learning repurposing of DrugBank compounds for opioid use disorder. *Comput. Biol. Med.* **2023**, *160*, 106921. [CrossRef] [PubMed]
25. Ezugwu, A.E.; Shukla, A.K.; Nath, R.; Akinyelu, A.A.; Agushaka, J.O.; Chiroma, H.; Muhuri, P.K. Metaheuristics: A comprehensive overview and classification along with bibliometric analysis. *Artif. Intell. Rev.* **2021**, *54*, 4237–4316. [CrossRef]
26. AlMuhaideb, S.; Ghannam, R.B.; Alhelal, N.; Alduheshi, S.; Alkhamees, F.; Alsuhaibani, R. IWD-miner: A novel metaheuristic algorithm for medical data classification. *Comput. Mater. Contin.* **2020**, *66*, 1329–1346. [CrossRef]
27. Hansen, N.; Müller, S.D.; Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol. Comput.* **2003**, *11*, 1–18. [CrossRef]
28. Akimoto, Y.; Hansen, N. Diagonal acceleration for covariance matrix adaptation evolution strategies. *Evol. Comput.* **2020**, *28*, 405–435. [CrossRef] [PubMed]
29. Bank, P.D. Protein data bank. *Nat. New Biol.* **1971**, *233*, 223.
30. Wilantho, A.; Tongsima, S.; Jenwitheesuk, E. Pre-docking filter for protein and ligand 3D structures. *Bioinformation* **2008**, *3*, 189. [CrossRef]
31. Abriata, L.A. Structural database resources for biological macromolecules. *Briefing. Bioinform.* **2017**, *18*, 659–669. [CrossRef]
32. Krotko, D.G. Atomic ring invariant and modified CANON extended connectivity algorithm for symmetry perception in molecular graphs and rigorous canonicalization of SMILES. *J. Cheminform.* **2020**, *12*, 48. [CrossRef]

33. Ferrandi, F.; Lanzi, P.L.; Loiacono, D.; Pilato, C.; Sciuto, D. A multi-objective genetic algorithm for design space exploration in high-level synthesis. In Proceedings of the 2008 IEEE Computer Society Annual Symposium on VLSI, Montpellier, France, 7–9 April 2008; pp. 417–422.
34. Graves, A.; Mohamed, A.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649.
35. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
36. Zuo, C.; Qian, J.; Feng, S.; Yin, W.; Li, Y.; Fan, P.; Han, J.; Qian, K.; Chen, Q. Deep learning in optical metrology: A review. *Light Sci. Appl.* **2022**, *11*, 39. [CrossRef] [PubMed]
37. Kotsias, P.C.; Arús-Pous, J.; Chen, H.; Engkvist, O.; Tyrchan, C.; Bjerrum, E.J. Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nat. Mach. Intell.* **2020**, *2*, 254–265. [CrossRef]
38. Liang, Y.; Wang, X.; Zhao, H.; Han, T.; Wei, Z.; Li, Y. A covariance matrix adaptation evolution strategy variant and its engineering application. *Appl. Soft Comput.* **2019**, *83*, 105680. [CrossRef]
39. Ros, R.; Hansen, N. A simple modification in CMA-ES achieving linear time and space complexity. In Proceedings of the Parallel Problem Solving from Nature–PPSN X: 10th International Conference, Dortmund, Germany, 13–17 September 2008; Proceedings 10; Springer: Berlin/Heidelberg, Germany, 2008; pp. 296–305.
40. Kumbure, M.M.; Luukka, P. Using the generalized fuzzy k-nearest neighbor classifier for biomass feedstocks classification. *Ann. Comput. Sci. Inf. Syst.* **2021**, *29*, 29–35.
41. de Boer, J.; Dedja, K.; Vens, C. SurvivalLVQ: Interpretable supervised clustering and prediction in survival analysis via Learning Vector Quantization. *Pattern Recognit.* **2024**, *153*, 110497. [CrossRef]
42. Chauhan, H.; Chauhan, A. Implementation of decision tree algorithm c4.5. *Int. J. Sci. Res. Publ.* **2013**, *3*, 1–3.
43. Teknomo, K. K-means clustering tutorial. *Medicine* **2006**, *100*, 3.
44. Hamid, N.A.; Nawi, N.M.; Ghazali, R. The effect of adaptive gain and adaptive momentum in improving training time of gradient descent back propagation algorithm on classification problems. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2011**, *1*, 178–184. [CrossRef]
45. Han, S.; Jin, Z.; Deji, D.; Han, T.; Zhang, Y.; Feng, M.; Hasi, W. Study on the classification and identification of various carbonate and sulfate mineral medicines based on Raman spectroscopy combined with PCA-SVM algorithm. *Anal. Sci.* **2023**, *39*, 241–248. [CrossRef]
46. Kuzmanic, A.; Zagrovic, B. Determination of ensemble-average pairwise root mean-square deviation from experimental B-factors. *Biophys. J.* **2010**, *98*, 861–871. [CrossRef] [PubMed]
47. Das, K.; Jiang, J.; Rao, J.N.K. Mean squared error of empirical predictor. *Ann. Stat.* **2004**, 32, 818–840. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.