Expert Systems With Applications 296 (2026) 129194

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

NRBO-AGP: A novel feature selection approach for accurate protein solubility prediction

Zahra Elmi [a], Soheila Elmi [b], Sebelan Danishvar [c,*]

[a] Department of Computer Engineering, Istanbul Sabahattin Zaim University, Halkalı Cad. No: 281 Halkalı, Küçükçekmece, Istanbul, 34303, Turkey
[b] Department of Electrical and Electronics Engineering, Koc University, Rumelifeneri Yolu, Sarıyer, Istanbul, 34450, Turkey
[c] Department of Electronic and Computer Engineering, Brunel University, London UB8 3PH, UK

ARTICLE INFO

Keywords:
Drug discovery
Protein solubility prediction
Metaheuristic approach
Feature selection

ABSTRACT

Protein solubility determines how well a protein dissolves in an aqueous solution, and this property is a critical factor in the functional analysis of proteins and biotechnological applications. Accurately estimating solubility can provide significant advantages in areas such as protein engineering and drug discovery. This study proposes a new feature selection method, Newton-Raphson-based Optimization and Adaptive Gradient Perturbation (NRBO-AGP) for predicting protein solubility. The research combines the accuracy and speed of the Newton-Raphson method with the capacity of population-based optimization techniques to balance exploration and exploitation. Using 3144 protein sequences from the eSOL database, descriptor features were obtained for each protein, resulting in a dataset with 3104 features. The performance of NRBO-AGP was compared with eight different metaheuristic algorithms and evaluated using five regression models: MLP, AdaBoost, Gradient Boosting Trees, Random Forest, and Support Vector Regressor (SVR). The best results were obtained with the Gradient Boosting and Random Forest. Mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination (R^2) metrics were used for performance evaluation. The results show that NRBO-AGP outperforms other metaheuristic algorithms in all regression models. The best results were achieved with Gradient Boosting and Random Forest, reaching MAE:0.0001 ± 0.0000, RMSE: 0.0008 ± 0.0000, and R^2: 0.9908 ± 0.0005, and MAE: 0.0002 ± 0.0000, RMSE: 0.0025 ± 0.0000, and R^2: 0.9908 ± 0.0005. These findings show that NRBO-AGP is an effective feature selection tool for predicting protein solubility. Multiple statistical analyses based on Friedman and Nemenyi tests show that the NBRO-AGP method exhibits statistically significant superior performance ($p < .05$) compared to other metaheuristic algorithms in MAE and RMSE metrics and also achieves the highest performance in the R^2 score.

1. Introduction

Proteins are vital macromolecules composed of amino acid chains present in every cell and tissue in the human body (Yugandhar et al., 2019). The functions of these macromolecules depend on their physicochemical and structural properties, one of which is solubility (Habibi et al., 2014). Protein solubility is a critical factor in drug production efficiency and the advancement of proteomic research. However, current computational techniques remain inadequate for accurately predicting protein solubility (Xiaohui et al., 2014). Various approaches, such as computational and experimental methods, are used to evaluate protein solubility. Escherichia coli (E. coli) bacteria are preferred for solubility assessment in many experimental techniques. However, problems such as inclusion bodies (protein aggregation) can be encountered during the protein expression process (Zhang et al., 2019). It is important to distinguish between soluble expression and aggregation-prone sequences. Soluble expression indicates that proteins fold correctly and remain soluble in the cytoplasm. In contrast, aggregation-prone sequences often fold incorrectly due to intrinsic properties such as hydrophobicity, charge imbalance, or repetitive motifs, leading to inclusion bodies. This distinction is particularly crucial when assessing the solubility potential of recombinant proteins in E. coli. Although methods such as strong denaturants, weak promoters, low temperatures, and optimized expression conditions are used to solve this problem, these experimental protocols require a significant amount of time and resources. Misfolding of newly synthesized peptides due to errors that occur during the formation of protein structures is the main cause of inclusion body formation (Boothroyd et al., 2018; Davis et al., 1999; Idicula-Thomas & Balaji, 2005; Pellizza et al., 2018). Therefore, protein sequences can be used to estimate the solubility of proteins. This estimation process is carried out by machine

* Corresponding author.
E-mail addresses: zahra.elmi@izu.edu.tr (Z. Elmi), selmi24@koc.edu.tr (S. Elmi), Sebelan.Danishvar@brunel.ac.uk (S. Danishvar).

https://doi.org/10.1016/j.eswa.2025.129194
Received 26 March 2025; Received in revised form 4 July 2025; Accepted 27 July 2025
Available online 29 July 2025

learning algorithms, especially neural networks, random forests, and support vector machines (Qian et al., 2020; Tang et al., 2017).In recent years, deep learning-based models have achieved significant success in the field of protein solubility prediction. For example, Chen et al. proposed a model called HybridGCN, which combines different sequence-based features with graph convolutional networks (GCN). This model blends classical biophysical descriptors with the protein language model (ESM-1v) to achieve high accuracy in solubility prediction (Chen et al., 2023). Deep-SoluE is a model developed by Wang and Zou and uses long-short-term memory (LSTM) networks to predict protein solubility. This model combines physicochemical properties and distributed representations obtained from amino acid sequences to provide more balanced and accurate predictions. DeepSoluE has demonstrated higher accuracy and stable performance than existing tools, particularly in tests on E. coli proteins (Kwon et al., 2024). Similarly, the GATSol model developed by Li and Ming presented an attention-based architecture that combines three-dimensional structural graph representations of proteins and language model outputs. This approach provided significant performance gains, particularly on the eSOL dataset, and showed an improvement of up to 18 % over previous methods (Li & Ming, 2024). These algorithms can make solubility estimates by analyzing protein sequences. However, the fact that data sets usually contain many features can negatively affect the performance of machine learning algorithms. In order to solve this problem, feature selection can be utilized as a vital approach to figuring out the critical and most relevant features in model training. By eliminating unnecessary features, feature selection improves the overall learning algorithm performance by concentrating on the most useful features (Tang et al., 2017). Feature selection algorithms can be categorized into five main groups: wrapper, filtering, embedding, ensemble, and hybrid methods (Abbasi Mesrabadi et al., 2023; Ghaderzadeh et al., 2024; Nemati et al., 2009; Rezaee et al., 2022; Sazzed, 2021; Singh et al., 2024; Tavasoli et al., 2021). Filtering strategies utilize statistical metrics to assess the significance of features as opposed to the learning model itself. In contrast, wrapper strategies employ the learning model to assess several feature subsets. Although this method can enhance performance, it has drawbacks such as high computational cost and overfitting. By involving feature selection in the learning process, embedded approaches strike a balance between computing efficiency and overfitting. Ensemble approaches increase the accuracy of classification tasks by combining several feature subsets to determine the best combination. Hybrid models integrate various feature selection strategies and utilize their respective advantages. An innovative hybrid feature selection method that addresses existing limitations is presented in this paper. Our proposed method combines wrapper-based metaheuristic algorithms with hybrid techniques to optimize feature selection while maintaining computational efficiency. The study evaluates and validates the effectiveness of our approach using six different regression algorithms: -multilayer perceptron (MLP) regressor, AdaBoost regressor, gradient boosting trees model, random forest regressor, support vector regressor (SVR), and ElasticNet. The main contributions of this research are:

- Development of an innovative hybrid model that improves feature selection accuracy.
- Comprehensive comparison of metaheuristic algorithms in protein solubility prediction.
- Demonstration of superior performance over existing approaches on the obtained dataset.

The rest of the paper is organized as follows: Section 2 presents the theoretical basis of feature selection. Section 3 describes the dataset, descriptor generation, and methodology. Section 4 presents experimental results and analysis. Finally, Section 5 includes concluding remarks and recommendations for future work.

2. Related work

While drug discovery stands out as one of the most challenging processes in the scientific world with its high costs and low success rates, artificial intelligence (AI) and machine learning (ML) technologies are reshaping this process with innovative approaches in critical areas such as molecular property prediction and the design of new molecules. These technologies have found wide application in the health sector. They have provided significant advances, especially in the early diagnosis of diseases, planning of treatment processes, and preventive health services. New developments in deep learning have demonstrated striking results, particularly in complex medical imaging analyses such as detecting and classifying brain tumors. Adapting artificial intelligence and machine learning techniques to the medical field has increased diagnostic success, accelerated analysis processes, and reduced costs. However, the high dimensionality of medical data has unique challenges, increasing the need for more efficient and accurate methods in feature selection and classification processes. Authors in Singh et al. (2024) have examined in detail the contributions of AI, particularly in drug screening and design processes. This study emphasizes the impact of AI in processes such as high-speed virtual screening (HTVS), pharmacophore modeling, and de novo drug design. It also includes examples of applications in areas such as toxicity prediction and pharmacokinetic profiling. The success of AI in predicting drug-target interactions, optimizing molecular structures, and drug repositioning processes has been examined, and it has been stated that success rates of up to 97 % in drug screening accuracy have been achieved with the use of deep learning (DL) algorithms. However, these high accuracy rates usually come with increased computational complexity and the risk of getting stuck in local optima. Our proposed NRBO-AGP method specifically addresses these challenges through its adaptive gradient perturbation mechanism. Molecular screenings, particularly for COVID-19 treatment, show the importance of AI-supported approaches in increasing clinical efficacy and speed. Similarly, another study on the contributions of AI in the fight against antimicrobial resistance (AMR) emphasizes that traditional drug development processes are insufficient due to high cost, long time, and frequent failure rates. In this context, the potential of AI technologies, particularly language models and DL methods, in processes such as identifying new antimicrobial agents, optimizing drug design, and predicting resistance mechanisms was examined. The findings show that AI integrated with genomic and proteomic data is effective in rapidly identifying new drug candidates, restructuring existing drugs, and estimating resistance models. For example, ML algorithms can rank molecules that may have antimicrobial activity by analyzing large-scale datasets, while DL models optimize drug design processes by predicting target-protein interactions (Ghaderzadeh et al., 2024). In parallel with these studies, an innovative computational framework for predicting drug-target interactions (DTI) has been presented. This framework comprises three main stages: feature extraction, selection, and classification. After managing high-dimensional data with a wrapper feature selection method called IWSSR, the selected features were passed to the Rotation Forest classifier. It has been shown that the framework achieves 98.12 %, 98.07 %, 96.82 %, and 95.64 % success rates for enzymes, ion channels, G-protein coupled receptors, and nuclear receptors, respectively. This method offers a time- and cost-saving mechanism compared to experimental methods (Abbasi Mesrabadi et al., 2023). The use of hybrid architectures based on CNN and LSTM in HAR has increased in recent years. DeepConvLSTM, developed by Ordóñez and Roggen (2016), is one of the pioneering works that combines CNN and LSTM layers to process sensor data, and this architecture achieved 93.7 % accuracy. Similarly, Hammerla et al. (2016) comprehensively compared the performance of RNN and CNN in HAR problems in their proposed deep learning architectures and showed that hybrid models provide more consistent results. The deepSense framework presented by Yao et al. (2017) recognized complex movements with 94.5 % accuracy using a hierarchical CNN-LSTM architecture that processes 6-axis sensor data separately and then fuses them. Recently, Choi et al. (2013) achieved 97.2 % accuracy by integrating the attention mechanism into the hybrid CNN-LSTM architecture in their proposed Attentional ConvLSTM model. However, most of these hybrid architectures use standard connection structures, and the proposed DeepHAR-Net (Ali & Abdelhafeez,

2022) stands out by using peephole connections in LSTM layers and its customized data augmentation strategy. These innovations enable DeepHAR-Net to be more robust to sensor placement variations and better capture complex activity patterns. In this respect, DeepHAR-Net differs from existing hybrid architectures in its architectural structure and performance metrics. While these drug-target interaction studies demonstrate the potential of feature selection in molecular analysis, similar challenges and opportunities exist in gene expression analysis. The efficient discovery-exploitation balance of our NRBO-AGP method becomes particularly valuable in this field. In this context, another study on hybrid algorithms used in the classification of microarray gene expression data proposes a new model called "Ensemble Soft Weighted Gene Selection" (ESWGS). This model determines gene weights using criteria such as the ROC curve, two-sample T-test, Wilcoxon test, Bhattacharyya distance, and entropy. It also includes the "Modified Water Cycle Algorithm" (mWCA) method to optimize the RBF kernel parameters of SVM. In experiments conducted on datasets such as leukemia, breast cancer, and prostate cancer, it has been shown that the model produces effective results with high accuracy and low computational cost (Tavasoli et al., 2021). Similarly, the ANOVA-SRC-BPSO method was developed to reduce the computational load in high-dimensional datasets and optimize cancer classification. Genes were filtered with ANOVA and F-tests, redundant genes were eliminated with Spearman rank correlation coefficients, and the most appropriate gene subset was selected with the BPSO algorithm. This method achieved 100 % classification accuracy in some datasets and generally achieved high accuracy using fewer genes (Sazzed, 2021). Another deep learning-based study aimed to classify cancer types and gene selection using microarray data. The proposed model used ROC curve, Wilcoxon test, and SNR methods for gene selection and utilized the Stacked Autoencoder (SAE) model in the classification phase. The study optimized gene selection with high accuracy rates, both shortening processing times and increasing overall performance. In this context, innovative solutions of AI and ML methods in drug discovery, gene selection, and classification processes provide an important foundation for the future in biomedical research (Rezaee et al., 2022). Our NRBO-AGP approach, which employs powerful feature selection mechanisms while maintaining computational efficiency, was directly motivated by these issues related to missing data and high dimensions. This study systematically examines feature selection (FS) methods used in cancer classification of microarray gene expression data (Alhenawi et al., 2022). FS methods have been developed to increase classification accuracy and reduce computational costs in high-dimensional datasets. They are of critical importance, particularly in the field of microarray data analysis. In the study, 132 scientific articles published in the last seven years were examined in detail, and FS studies were divided into five main categories: filter-based, wrapper, embedded, hybrid, and ensemble approaches. These categories reveal the strengths and weaknesses of the methods in terms of accuracy, computational cost, and generalization capacity. It is known that microarray gene expression data are widely used in cancer diagnosis and developing prognostic models. However, the high dimensionality and low sample number frequently encountered in such data reduce the generalization capacity of the models and increase the risk of overfitting (Osama et al., 2023). The use of dimensionality reduction algorithms, such as feature selection and feature extraction, has gained importance to solve these problems. In this context, FS algorithms are implemented with different approaches such as filter, wrapper, embedded, hybrid, and ensemble methods. Within this classification, the comparative analysis, particularly between hybrid and ensemble methods, offers important practical implications for researchers. Hybrid methods provide the ability to narrow down the search space more effectively by integrating the strengths of different paradigms. For example, reducing the dimensionality of the feature space with filtering algorithms and then applying wrapper techniques can significantly increase computational efficiency. On the other hand, this integration process increases the complexity of the method and complicates the implementation process, as it requires multiple parameter optimizations. In addition, combining different algorithms can lead to inconsistencies between methods and make it difficult to verify the results. Ensemble feature selection methods, on the other hand, provide high generalization capability by combining the outputs of multiple models. These approaches can produce more robust and reliable predictions than a single model, if there is noise in the dataset or complex relationships between features (Darmawahyuni et al., 2024). The natural structure The natural structure of ensemble methods can be easily integrated with cross-validation techniques, which increases the reliability of the model selection process. However, ensemble approaches generally require higher computational costs, which can be a significant limitation, particularly in high-dimensional datasets or limited-resource scenarios. It should also be noted that ensemble methods have disadvantages in interpretability and carry the risk of overfitting if not carefully designed. In practical applications, hybrid and ensemble methods vary depending on the problem context and operational constraints. Hybrid approaches may be more advantageous in very high-dimensional datasets or where computational resources are limited. On the other hand, ensemble methods can be preferred in applications where generalization ability is critical, or the aim is to minimize the prediction variance. The NRBO-AGP method proposed in this study combines the fast convergence advantage of Newton-Raphson optimization with the discovery ability of adaptive gradient perturbation as a hybrid approach. This integration exhibits superior computational efficiency and performance and avoids local optima in complex bioinformatics problems such as protein solubility estimation. Compared to other studies, hybrid methods offer lower computational cost and higher interpretability, while ensemble approaches provide more robust and generalizable results. Researchers should consider this trade-off when choosing the most appropriate strategy for their spic applications. For example, while ensemble methods stand out in areas requiring high accuracy, such as clinical decision support systems, hybrid approaches may be more suitable in real-time systems or resource-constrained environments. As a result, when determining the feature selection approach, the advantage-disadvantage balance offered by hybrid and ensemble strategies should be evaluated comprehensively, considering the dataset's characteristics, the application domain's requirements, and the existing computational infrastructure. Another significant contribution is that metaheuristic algorithms have many applications in feature selection processes. These algorithms have been investigated using various metrics and classifiers on single and multiple objective functions (Barrera-García et al., 2023). Specifically, physics-based adaptations, human behavior-based, swarm intelligence-based, and evolutionary-based algorithms have significantly contributed to the FS area by offering remarkable accuracy rates on large datasets (Agrawal et al., 2021). Furthermore, the investigation of multi-class FS problems has highlighted the necessity for the scalability and robustness of these algorithms (Akinola et al., 2022). The necessity of scalability and robustness in feature selection approaches is obviously aligned with the design principles of our proposed NRBO-AGP approach, which covers these restrictions with an innovative combination of Newton-Raphson optimization and adaptive gradient techniques. Therefore, FS methods play a critical role in improving classification accuracy and reducing the computational costs in high-dimensional datasets. The studies mentioned show the effective utilization of artificial intelligence, machine learning, and metaheuristic algorithms in FS processes, which is critical to enhancing generalization capacity and maximizing model performance. In this direction, more comprehensive research and the development of innovative approaches will help progress in the FS field. The DDcCNN model (Wang et al., 2021) is an innovative solution that uses computational techniques and sequence data to predict protein solubility. The model combines the advantages of local and global feature extraction with one-layer 1D convolutional networks and three-layer 2D convolutional networks. The extracted features for solubility prediction are processed in a four-layer fully connected network. The model's performance is evaluated with a dataset

of 129,643 protein sequences, consisting of 58,689 soluble and 70,954 insoluble proteins. The results reveal that the DDcCNN model has superior performance in terms of sensitivity (76.13 %), specificity (79.32 %), Matthew correlation coefficient (MCC, 0.57), and accuracy (77.82 %). Moreover, the MCC and accuracy values of the model are better than those of other models, such as PaRSnIP and DeepSol. Comparison of training times shows that the DDcCNN model can be used to predict protein solubility in real-world applications. Another study (Manzoor et al., 2023) presented a new method for amino acid residue selection by combining unsupervised feature extraction with autoencoders, with three different feature selection strategies. The model was tested on five benchmark datasets, namely CB6133, CB6133-filtered, CB513, CASP10, and CASP11, using random forest, decision tree, and multilayer perceptron classifiers. The findings showed that Q8 accuracy ranged from 82 % to 74 % and Q3 accuracy ranged from 92 % to 74 %. The model achieved an average improvement of 3.5 % in Q8 accuracy. While the random forest classifier performed best in general metrics, the decision tree achieved better results in specific areas. The model also improved the performance in prediction tasks by eliminating noisy and unnecessary data. In another study on protein function classification (De Santis et al., 2018), feature selection methods and dissimilarity space representations were used. The authors presented methods that convert protein structures into real-valued vectors that can be used with standard classification techniques. The study achieved success in classifying protein activities and showed promising results in tests on a subset of the E. coli proteome. Newton-Raphson Based Optimizer (NRBO) (Sowmya et al., 2024) improves the traditional Newton-Raphson approach by introducing two basic operators, namely the Trap Avoidance Operator (TAO) and the Newton-Raphson Search Rule (NRSR). These operators increase the algorithm's exploitation capacity, convergence rate, and ability to avoid local optima. NRBO has been evaluated on standard benchmark problems such as CEC2020 and CEC2017 and outperforms seven other advanced optimization algorithms. It has also been successfully applied to training deep reinforcement learning agents and optimizing IoV routing problems. It is stated that NRBO further improves the performance by combining population-based and gradient features. These limitations in feature selection methods motivated the development of our NRBO-AGP approach, which combines the fast convergence property of Newton-Raphson with the ability of gradient-based optimization to avoid local optima. NRBO-AGP provides a more effective exploration-exploitation balance in the search space thanks to NRBO-AGP operators. It makes a unique contribution to the literature with its high accuracy rates and consistent results, particularly in large-sized data sets.

3. The dataset, descriptor generation, and preliminary

3.1. eSol dataset

The data for protein solubility employed in the study comes from the eSol database (Niwa et al., 2009), which is an extensive repository of quantitative protein solubility values for ensemble E. coli proteins. It is derived from the eSOL platform, where protein solubility is assessed using experimental investigation of their physicochemical properties in a PURE system. This dataset consists of protein solubility values experimentally measured in the PURE (Protein Synthesis Using Recombinant Elements) system by Niwa et al. The dataset used in our study contains a total of 3144 E. coli proteins after eliminating those with missing sequence information. The solubility value for each protein was determined by producing recombinant proteins using cell-free protein expression technology and separating them into soluble and insoluble components by centrifugation. Solubility was calculated by dividing the protein ratio in the supernatant by the total protein content and takes continuous values in the range of [0,1]. The solubility values in the eSOL dataset show a continuous distribution and are generally used for regression problems. However, it is possible to divide the dataset into soluble and insoluble for classification studies. Based on the threshold value of 0.5, which is widely used in the literature, the class distribution in our data set is as follows: soluble proteins ($resolution \geq 0.5$), 1837 samples (58.4 %), and insoluble proteins ($resolution < 0.5$), 1307 samples (41.6 %). When the statistical distribution of the solubility values is examined, the mean solubility is 0.57, the median solubility is 0.62, the standard deviation is 0.29, the minimum value is 0.0, the maximum value is 1.0, and the interquartile range (IQR) is 0.48. The distribution of solubility values shows a slight bimodal characteristic; there are two separate concentration points in the ranges of $0.2 - 0.3$ and $0.7 - 0.8$. This distribution reflects the dual effect of the physicochemical properties of the proteins on solubility. Following the exclusion of entries lacking sequence information, 3144 proteins from the eSol database were included in our study. The original study that generated this dataset assessed protein solubility values by producing recombinant proteins using cell-free protein expression technology. The expressed proteins were subsequently fractionated into soluble and insoluble components through centrifugation. Solubility is the supernatant protein ratio to total protein content, which was computed by SDS-PAGE (Shimizu et al., 2005).

3.2. Preprocessing

TTo generate our research database, we apply several steps to the protein sequences of eSOL (Osorio et al., 2015). We use the Peptides package in the R language to achieve the protein sequence descriptors. We first install and load it to enable us to utilize the functionality of the package. We then use the "aaDescriptors" function to assess the sequences of the eSOL proteins and generate 66 descriptors for each amino acid. The several descriptors of amino acids are the aliphatic index, Boman index, net charge, hydrophobicity, instability index, isoelectric point, and molecular weight. To facilitate interpretation, we gather the obtained descriptor values into a structured data framework. Generating a structured data framework with multiple protein sequences is the first step of our research; each sequence contains multiple properties for each amino acid. Using a loop mechanism, we iteratively perform the "aaDescriptors" function on each sequence. Finally, these distinct data obtained are gathered into a single matrix frame that includes all generated descriptors for the sequences of the proteins. The resulting dataset contains 3144 instances and 3104 features. Although the dataset was relatively balanced, a stratified sampling technique was applied to prevent potential biases during the model training and evaluation processes. For sampling, solubility values were divided into five equal intervals ($0 - 0.2, 0.2 - 0.4, 0.4 - 0.6, 0.6 - 0.8, 0.8 - 1.$), and a proportional number of samples were selected from each interval. This approach ensured that each training/validation/test set represented the solubility distribution in the original dataset. The dataset was divided into 70 % training ($n = 2201$),10 % validation ($n = 314$), and 20 % test ($n = 629$). The soluble and insoluble protein ratios in each divided set were consistent with the original dataset's ratio (58 % – 42 % balance was maintained). Sixty-six descriptive features were extracted for each protein sequence using the "aaDescriptors" function in the Peptides package. These features include parameters such as aliphatic index, Boman index, net charge, hydrophobicity, instability index, isoelectric point, and molecular weight. The average values divided by the number of amino acids of each sequence were used to standardize the length differences between the sequences. As a result of this process, a data matrix containing a total of 3104 features for 3144 proteins was created. All features were normalized to the range [0, 1] with the min-max scaling method to ensure comparability between models and to prevent potential misconceptions due to the different scales of the features. In bioinformatics and computational biology research, this method efficiently utilizes the Peptides package and the R language's flexibility to perform profound analysis of the attributes of proteins. After that, the min-max scaler was used to normalize the data.

3.3. Preliminary

3.3.1. Newton-Raphson-based optimization

Optimization problems are categorized into two different classes: algorithms based on gradient, such as Newton's Method (NM) (Amrein & Wihler, 2014), Gradient-Descent Algorithm (GDA) (Madgwick et al., 2011), Levenberg Marquardt Algorithm (LMA) (Moré, 2006), Quasi-Newton's Method (QNM) (Weerakoon & Fernando, 2000), and algorithms based on non-gradient-based methods, such as MAs, such as GA, GWO, WOA, ACO, and PSO, etc. To find the optimal solutions, gradient-based algorithms (GB) are based on discovering the points where the gradient is zero; algorithms like NM and conjugate direction approaches follow this principle. The gradient algorithms have disadvantages, such as slow convergence speed and no guarantee of the best solution. Metaheuristic algorithms are flexible mechanisms for solving problems that perform specified procedures to accomplish optimization without depending on the domain of a specific problem. They are inspired by natural phenomena and utilize heuristic approaches that can be designed for various optimization aims. Metaheuristic approaches such as GA, GWO, WOA, ACO, and PSO present powerful and effective methods for optimization in a comparatively short time, contrasting with the exact optimization methods that obtain the optimal solutions after considerable computation. Metaheuristic algorithms have an advantage when utilizing complex models and large datasets; they produce high-quality answers with low errors quickly. In addition, the flexibility of metaheuristic methods enables them to adapt easily to real-world scenarios and distinguishes them from more rigid, accurate optimization approaches. While MAs offer excellent robustness in searching for the optimal solution, GB gets stuck with local optimal solutions. On the other hand, MAs require more CPU cores, which is particularly important for problems with large search spaces. Therefore, we suggest a novel method that combines the advantages of gradient-based and metaheuristic algorithms and uses them for feature selection. The Newton-Raphson approach is a method that uses the Taylor series to find the root of a function. Initially, a point (x_0) is chosen, and the Taylor series of the function is calculated around this point (so that we consider only up to second-order terms) (Sowmya et al., 2024):

$$f(x_0+\epsilon) \approx f'(x_0)\cdot\epsilon + \frac{(f''(x_0)\cdot(\epsilon)^2)}{2}, \tag{1}$$

If $f(x_0+\epsilon)=0$ and solving Eq. 1 for $\epsilon \equiv \epsilon_0$ we will have,

$$\epsilon_0 = -\frac{f'(x_0)}{f''(x_0)}, \tag{2}$$

This determines the next position of the root, and the process is repeated until the root is found:

$$x_{(n+1)} = x_n - \frac{f'(x_n)}{f''(x_n)}, n = 1,2,3,\ldots \tag{3}$$

Newton-Raphson-Based Optimization (NRBO) (Sowmya et al., 2024) explores the search region using the Newton-Raphson Method and defines the search path using various operators. Consider that optimization is performed on an unconstrained single-objective problem as follows:

$$Minimize: f(x_1,x_2,\ldots,x_n) \text{subject to} lb \le x_j \le ub, j=1,2,\ldots,dim \tag{4}$$

where $f(x)$ is the objective function minimizing x_j, which is the decision vector, dim is the dimension of the problem, lb and ub are lower and upper bounds, respectively. NRBO, similar to other metaheuristic algorithms, starts investigating optimal solutions by generating initial random populations. The random population is generated by the following equation:

$$x_j^n = lb + rand\times(ub-lb), n=1,2,\ldots,N_p; j=1,2,\ldots,dim, \tag{5}$$

Where x_j^n is the position of the n^{th} population in the j^{th} dimension, $rand$ is a random number in the interval $(0,1)$, and N_p is the total number of the population. The Newton-Raphson Search Rule (NRSR) is presented as an effective solution method for variation problems. This allows vectors to explore the feasible region more accurately and obtain better positions. It is based on the idea of the Newton-Raphson Method (NRM) to increase the exploration tendency and speed up convergence and is an adaptation of the NRM and adopts a permanent approach so that it can be used for non-differentiable functions. NRM starts with an initial solution and progresses to the next position in a specified direction. Using the Taylor series of second-order derivatives to obtain NRAK from Eq. 3, the derivatives of $f(x)$ are determined as follows:

$$f'(x) = \frac{(f(x+\triangle x)-f(x-\triangle x))}{2\times\triangle x}, \tag{6}$$

$$f''(x) = \frac{f(x+\triangle x)+f(x-\triangle x)-2\bullet f(x)}{\triangle x^2}, \tag{7}$$

By substituting these derivatives into Eq. 3, the updated root position is written as follows:

$$x_(n+1) = x_n - \frac{\left(\left(f(x_n+\triangle x)-f(x_n-\triangle x)\right)\times\triangle x\right)}{2\times(f(x_n+\triangle x)+f(x_n-\triangle x)-2\times f(x_n))}, \tag{8}$$

This equation is adjusted for NRSR to manage population-based search. By determining the best and worst positions, NRSR is expressed as follows:

$$NRSR = randn\times\frac{(X_w-X_b)\times\triangle x}{2\bullet(X_w+X_b-2\times x_n)}, \tag{9}$$

Here, $randn$ is a random number with a normal distribution, X_w and X_b represent the worst and best positions, respectively. $\triangle x$ is determined as follows:

$$\triangle x = rand(1,dim)\times|X_b - X_n^t|, \tag{10}$$

where t is the current iteration. To improve the performance of the algorithm, an adaptive coefficient δ is used, which provides a balance between exploration and exploitation.

$$\delta = \left(1-\left(\frac{2\times t}{T}\right)\right)^5, \tag{11}$$

Using Eq. 8 and NRSR, the position is updated:

$$x_{(n+1)} = x_n - NRSR, \tag{12}$$

To improve exploitation, NRBO uses the parameter ρ to determine the direction of the population:

$$\rho = \alpha\times\left(X_b - X_n^t\right) + \beta\left(X_i^t - X_j^t\right), \tag{13}$$

Where α and β are random numbers in intervals $(0,1)$, i and j are different integers that are randomly chosen from the population. The current position is updated as follows:

$$X1_n^t = x_n^t - \left(randn\times\frac{X_w-X_b)\times\triangle x}{2\times(X_w+X_b-2\times x_n}\right) + \left(\alpha\times\left(X_b-x_n^t\right)+\beta\left(X_i^t-X_j^t\right)\right), \tag{14}$$

Eqs. 15 and 16 present local and global search strategies:

$$X1_n^t = x_n^t - \left(randn\times\frac{y_w-y_b)\times\triangle x}{2\times(y_w+y_b-2\times x_n}\right) + \left(\alpha\times\left(X_b-x_n^t\right)+\beta\left(X_i^t-X_j^t\right)\right), \tag{15}$$

$$X2_n^t = X_b - \left(randn\times\frac{y_w-y_b)\times\triangle x}{2\times(y_w+y_b-2\times x_n}\right) + \left(\alpha\times\left(X_b-x_n^t\right)+\beta\left(X_i^t-X_j^t\right)\right), \tag{16}$$

where y_w and y_b are the positions of the two vectors formed using $x_{(n+1)}$ and x_n, respectively. NRBO uses the above two equations to develop both the diversification and intensification phases. The new position vector is determined by Eqs. 17 and 18:

$$X_n^{(t+1)} = \gamma\times\left(\gamma\times X1_n^t + (1-\gamma)\times X2_n^t\right) + (1-\gamma)\times X3_n^t, \tag{17}$$

$$X3_n^t = X_n^t - \delta \times (X2_n^t - X1_n^t), \tag{18}$$

where γ is the random number in intervals $(0, 1)$. For clarity, a complete list of hyperparameters used in Equations 6-12, along with their descriptions and values, is presented in Appendix A.

3.3.2. Adaptive gradient perturbation

The adaptive gradient perturbation (AGP) (Minervini et al., 2023) method is based on the adaptive perturbation of the gradient to improve optimization processes in machine learning and deep learning. This method was developed to prevent getting trapped in local minimum and to speed up the convergence time, particularly in complex and high-dimensional problems. The main aim of this method is to provide a more effective learning process by dynamically adjusting the gradients during the model training. The AGP method basically modifies the gradient descent algorithm as follows:

$$\theta_{(t+1)} = \theta_t - \eta(\nabla f(\theta_t) + \epsilon_t), \tag{19}$$

where θ_t represents the current model parameters, η is the learning rate, $\nabla f(\theta_t)$ is the gradient of the loss function, and ϵ_t is the adaptive perturbation term. ϵ_t is usually calculated by the formula:

$$\epsilon_t = \alpha \times \sigma(\nabla f(\theta_t)) \times N(0, I), \tag{20}$$

where α is a hyperparameter controlling the perturbation strength, $\sigma(\nabla f(\theta_t))$ is the standard deviation of the gradients, and $N(0, I)$ is a random vector drawn from the standard normal distribution. The adaptive nature of AGP is achieved by dynamically adjusting α during training:

$$\alpha_{(t+1)} = \alpha_t \times \exp\left(\lambda \times \left(\rho - \frac{||\nabla f(\theta_t)||}{||\nabla f(\theta_{(t-1)})||}\right)\right), \tag{21}$$

where λ determines the adaptation speed and ρ determines the target gradient rate. These formulas show how AGP improves the optimization process by dynamically perturbing the gradients so that the model can explore a larger solution space and avoid local minima.

3.3.3. Hybrid of Newton-Raphson optimizer and adaptive gradient perturbation

Feature selection plays a critical role in improving the performance and interpretability of machine learning models. In this paper, an innovative feature selection algorithm is presented that combines Newton-Raphson optimization and AGP techniques. This approach aims to provide a more effective and robust feature selection process by going beyond traditional methods. The proposed approach consists of eight important steps described below:

1. Problem Formulation: The feature selection problem is formulated as a continuous optimization problem. A selection degree is defined for each feature, $s_i \in [0, 1]$. Here $s_i = 1$ indicates that the feature is completely selected, while $s_i = 0$ shows that the feature is completely eliminated. This continuous formulation allows the use of gradient-based optimization techniques.
2. Data Preparation and Starting Point: The algorithm converts the dataset to PyTorch tensors, enabling fast computations on the GPU. The starting point is sampled from a uniform distribution in the range $[\theta, 1]$ to increase the probability of selecting features:

$$s_0 = \theta \times N(0, 1), \tag{22}$$

where $N(0, 1)$ represents the uniform distribution in the range $[0, 1]$.
3. Define Objective Function: The objective function, which forms the core of the algorithm, consists of three main components:

$$f(s) = CVMSE(s) + \alpha(s, r)L_1(s) + \beta(s, r)L_2(s), \tag{23}$$

where $CVMSE(s)$ is the mean square error calculated by K-fold cross-validation, $L_1(s)$ represent L_1 norm (Lasso regularization), and $L_2(s)$ is L_2 norm (Ridge regularization). $\alpha(s, r)$ and $\beta(s, r)$ are L_1 and L_2 regularization coefficients, respectively. Population-based training is used for (α, β, and η), which has an $O(N * T)$ time complexity, where N is the population size and T is the number of CatBoost iterations. $\alpha(s, r)$ and $\beta(s, r)$ are as follows:

$$\alpha(s, r) = \alpha_0 + \left(1 + \frac{r}{N}\right) \times D(s), \tag{24}$$

$$\beta(s, r) = \beta_0 + \left(1 + \frac{r}{N}\right) \times D(s), \tag{25}$$

$$D(s) = exp(-\gamma \times diversity_score(s)), \tag{26}$$

where r is the model performance ranking in the population, and N is the population size. $D(s)$ presents an adaptive term based on population diversity, γ is Diversity weight parameter, and $diversity_score(s)$ is the feature subset's uniqueness score in the population. α_0, β_0 are initial regularization coefficients. This formulation allows feature selection to adapt to population diversity, applies stronger regularization for low-performing models, provides a better exploration of the parameter space during the optimization process, and automatically adapts according to cross-validation performance. CVMSE is calculated using CatBoost regression at each fold:

$$CVMSE(s) = \left(\frac{1}{k}\right) \times \sum_{(k=1)}^{K} MSE(s, k) + \alpha(s, r)L_1(s) + \beta(s, r)L_2(s), \tag{27}$$

4. Utilize Newton-Raphson Optimization: The Newton-Raphson method rapidly approaches the optimal solution using the gradient of the objective function and the Hessian matrix. The Newton-Raphson formula is as follows:

$$s_{(t+1)} = s_t - \eta[H^{(-1)} \times \nabla f(s_t), \tag{28}$$

Here H represents the Hessian matrix and $\nabla f(s_t)$ is the gradient. However, since direct Hessian computation can be computationally expensive, the $AdamW$ optimizer is used in this implementation.
5. Apply Adaptive Gradient Perturbation: AGP helps avoid local minima by adding a stochastic element to the optimization process. Small, random perturbations are added to the gradient:

$$\nabla f_{perturbed(s)} = \nabla f(s) + \epsilon \times N(0, I), \tag{29}$$

Here ϵ is the perturbation magnitude and $N(0, I)$ represents the multidimensional standard normal distribution.
6. Optimization process: The optimization process is performed using AdamW optimizer and learning rate planning. At each iteration t:
‘ - The objective function is calculated: $f_t = f(s_t)$
- The gradient is calculated by automatic differentiation:
$\nabla f_t = \nabla f(s_t)$
- AGP is applied:
$\nabla f_{perturbed(s)} = \nabla f(s) + \epsilon \times N(0, I)$
- The AdamW optimizer updates the parameters using the gradient:

$$m_t = \beta_1 \times m_{(t-1)} + (1 - \beta_1) \times \nabla f_{perturbed(s)}, \tag{30}$$

$$v_t = \beta_2 \times v_{(t-1)} + (1 - \beta_2) \times \nabla f^2_{perturbed(s)}, \tag{31}$$

$$s_{(t+1)} = s_t - \eta_t \times \frac{m_t}{\sqrt{(v_t)} + \epsilon}, \tag{32}$$

Where β_1 and β_2 are the momentum parameters of $AdamW$, and η_t is the learning rate at step t.
7. Learning Rate Scheduling: The step size is decreased over time:

$$\eta_(t + 1) = \eta_t + \gamma^{(t/T)}, \tag{33}$$

where η_t is the initial learning rate, γ is the decay factor, and T is the step size period.
8. Termination and Feature Selection: The optimization is terminated when a predetermined number of iterations is reached or when the convergence criterion is met. Final feature selection is performed by applying a threshold value to the continuous values obtained from the optimization result: $selected - features = (s^* > \theta)$ where s^* is the optimized feature vector and θ demonstrates the selection threshold value that is obtained by a quantile-based dynamic threshold and

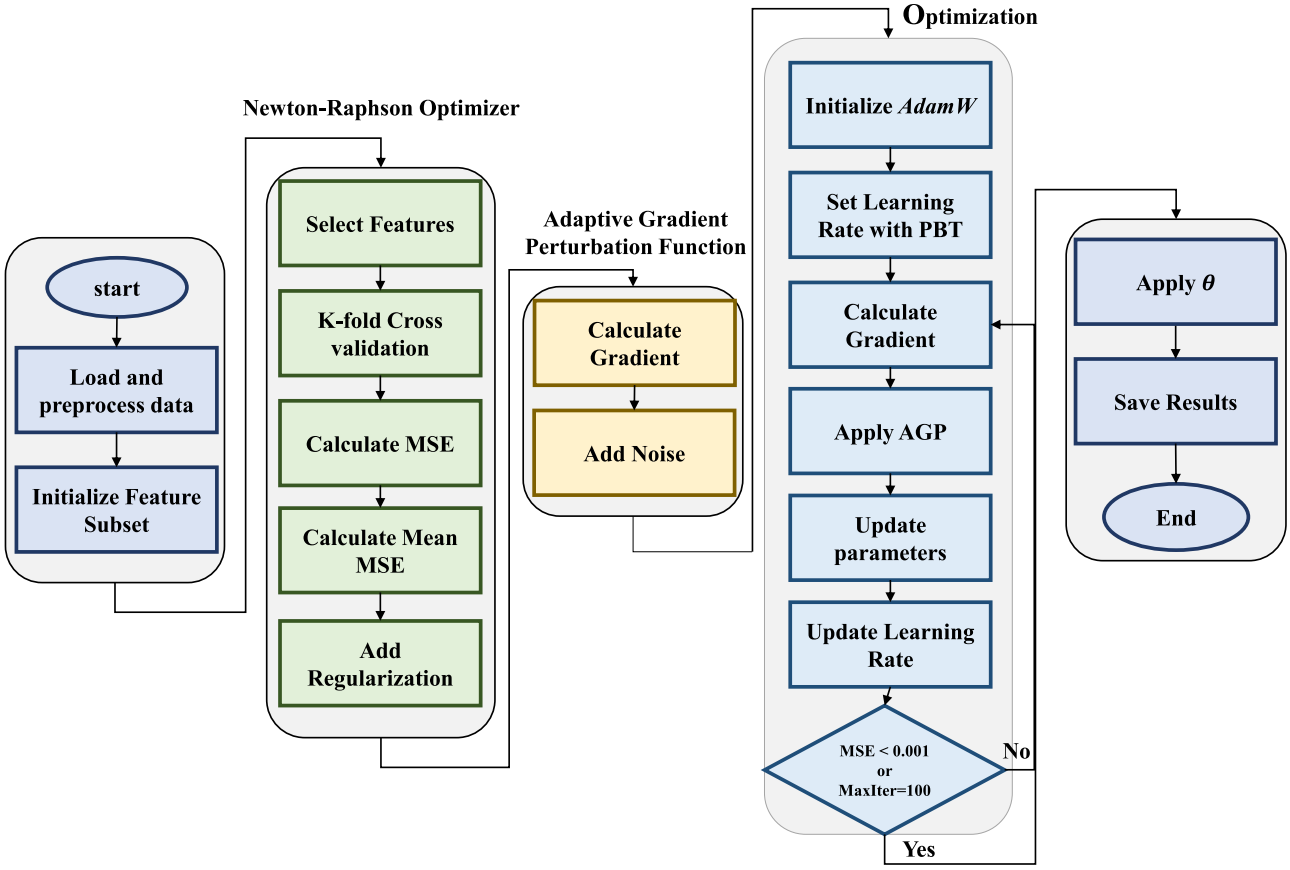

Fig. 1. flowchart of the NRBO-AGP approach.

has an $O(n * log(n))$ time complexity. The pseudocode and flowchart Fig. 1 of the proposed approach are as follows:

This new algorithm combines the fast convergence property of Newton-Raphson optimization, the exploratory ability of AGP, and the flexibility of modern optimization techniques to provide an effective feature selection method. The continuous optimization approach leverages the power of gradient-based methods, while the AGP extension helps avoid local minima. This approach has the potential to show strong performance, in particular on high-dimensional datasets and complex model structures. The time complexity of this method is based on the combination of several main components. Newton-Raphson optimization is iterated for a certain number of iterations with the objective function called in each iteration. The CatBoost model is trained and predicted using K-fold cross-validation in this function. In addition, adaptive gradient perturbation is applied in each iteration. The main factors affecting the time complexity include the number of instances, the number of features, the number of Newton-Raphson iterations, the number of K-folds, and the number of iterations of the CatBoost model. Roughly, the time complexity can be expressed as $O(t * K * (n * d * T + n * \log(n)))$, where t represents the total number of iterations, K demonstrates the number of folds, n is the number of samples, d shows the number of features, and T is the number of CatBoost iterations. Although the use of GPU can reduce the computational time.

4. Result and discussion

This study proposes an approach based on a novel population for feature selection called the Newton-Raphson-Based Optimizer and Adaptive Gradient Perturbation, which combines Newton-Raphson optimization and adaptive gradient perturbation. The integration of these two methods offers the combined benefits of fast convergence and the avoidance of local optima in the feature selection process. The Newton-Raphson method is a powerful iterative method used to find the roots of functions. In the context of feature selection, this method can help quickly find the optimal feature subset. The method continuously improves the current solution, converging to a better feature combination in each iteration. On the other hand, Adaptive Gradient Perturbation is a variant of gradient-based optimization techniques. This method adds small random perturbations to the gradient to explore the search space more effectively. In the feature selection process, this approach can reduce the risk of getting stuck in local optima and explore a larger solution space. Combining these two methods can provide a more robust and effective feature selection process by combining the fast convergence property of Newton-Raphson with the exploration capability of Adaptive Gradient Perturbation. This combination can be helpful in complex and high-dimensional datasets. Model development and experimental analyses were performed on the Python platform. We explicitly stated the exact split ratios (70 %/10 %/20 %) of the training/test/validation datasets used in the experimental setup and the detailed cross-validation strategy (5-fold CV). These experiments are carried out on a 3.70 GHz Intel Core i5 PC with 16 GB of RAM and a GeForce RTX 4070 with 12 GB. We also utilized eight metaheuristic algorithms that are population-based, such as the Whale Optimization Algorithm (WOA) (Mirjalili & Lewis, 2016), Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014), Ant Lion Optimizer (ALO) (Mirjalili, 2015a), Moth Flame Optimizer (MFO) (Mirjalili, 2015b), Dragonfly Algorithm (DA) (Mirjalili, 2016), Grasshopper Optimization Algorithm (GOA) (Mirjalili et al., 2018), Multi-Verse Optimizer (MVO) (Mirjalili et al., 2016), and Salp Swarm Algorithm (SSA) (Abualigah et al., 2020), for feature selection in our generated dataset to demonstrate that the proposed method outperforms the above-mentioned methods in protein solu- bility prediction.

Algorithm 1 Feature selection optimization using AGP and Newton-Raphson.

Load and preprocess data:
  $X \in \mathbb{R}^{n\times m}, y \in \mathbb{R}^n$
Initialize feature subset randomly:
  $s_0 \in \mathbb{R}^m, s_0 \sim \mathcal{N}(\theta, 1)$
Define objective function $f(s)$**:**
function F(s)
  a. Select features:
    $X_{\text{selected}} = X[:, s > \theta]$
  b. Perform k-fold cross-validation:
  for each fold do
    Train CatBoost model
    Predict: $\hat{y} = \text{CatBoost}(X_{\text{test}})$
    Calculate MSE:
    $\text{MSE}_{\text{fold}} = \frac{1}{n_{\text{test}}} \sum (y_{\text{test}} - \hat{y})^2$
  end for
  c. Calculate mean MSE:
    $\text{MSE} = \frac{1}{k} \sum \text{MSE}_{\text{fold}}$
  d. Add regularization term:
    $f(s) = \text{MSE} + \alpha||L_1(s)|| + \beta||L_2(s)||^2$
  e. Return $J(\theta)$
end function
Define Adaptive Gradient Perturbation (AGP) function:
function AGP($\nabla f(s), \epsilon$)
  return $\nabla f(s) + \epsilon \cdot \mathcal{N}(0, I)$
end function
Define Newton-Raphson optimizer with AGP:
function OPTIMIZE($s_t, J, T, \eta_t, \epsilon$)
  Initialize AdamW optimizer with learning rate η_0
  Initialize learning rate scheduler
  $s_{t+1} = s_t$
  for $t = 1$ **to** T **do**
    Calculate $J(\theta)$ and $\nabla J(\theta)$
    Apply AGP:
    $\nabla J'(\theta) = \text{AGP}(\nabla J(\theta), \epsilon)$
    Update θ using AdamW:
    $m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla f_{\text{perturbed}}(s)$
    $v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (\nabla f_{\text{perturbed}}(s))^2$
    $\hat{m}_t = \frac{m_t}{(1-\beta_1^t)}$
    $\hat{v}_t = \frac{v_t}{(1-\beta_2^t)}$
    $s_{t+1} = s_t - \eta_t \cdot \frac{\hat{m}_t}{(\sqrt{\hat{v}_t}+\epsilon)}$
    Update learning rate: $\eta = \text{scheduler}(\eta)$
  end for
  return s_{t+1}
end function
Optimize feature subset:
  $s^* = \text{optimize}(s_t, J, T, \eta_t, \epsilon)$
Apply threshold to select final features:
  selected_features = $s^* > \theta$
Save and output results

Each metaheuristic technique in this study used a population size of 100 agents and was run for 70 iterations. Root mean squared error, mean absolute error, and R^2 error measurement in the five-fold cross-validation method were applied to evaluate the performance of the proposed techniques (Algorithm 1).

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - p_t)^2}, \tag{34}$$

$$MAE = \frac{1}{n}\sqrt{\sum_{t=1)}^{n}(y_t - p_t)}, \tag{35}$$

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(y_t - p_t)^2}{\sum_{t=1}^{n}\left(y_t - \sum_{t=1}^{n} y_t\right)^2}, \tag{36}$$

where y_i, p_i, and n denote the actual, predicted values and number of observations, respectively. Fig. 2(a) shows the correlation coefficients of selected features using the proposed method with the target variable, which is the solubility of the protein. The Y-axis shows the correlation coefficients between 0 and 1, and the X-axis represents the different selected variables. This correlation analysis graph shows the relationships between feature selection and the target variable, providing important insights into machine learning modeling. The descriptors determined by the NRBO-AGP hybrid feature selection algorithm and shown in Fig. 2(a) provide a comprehensive profile of various biochemical and physicochemical parameters affecting protein solubility. When the selected features are examined, it is seen that the hydrophobicity/hydrophilicity properties of amino acids at positions 44–47 (VHSE1.44, VHSE2.44, Z1.44, Z1.47, PP2.45, PP3.44), steric and volumetric characteristics (VHSE3.44, VHSE4.46, Z2.46, Z4.44), electronic and charge distributions (VHSE6.44, VHSE7.44, VHSE7.46, VHSE8.45), and structural conformation tendencies (F1.45, F2.44, F2.45, F3.47, ST3.47, ST5.44, ST6.44) are prominent. These properties' significantly high correlation coefficients prove that these biochemical parameters play a decisive role in protein solubility. Hydrophobic-hydrophilic balance directly affects the solvent interactions of the protein and emerges as the primary determinant of the solubility profile. While the optimum distribution of hydrophilic amino acids on the protein surface increases the solubility by providing appropriate interactions with water molecules, incorrect positioning of hydrophobic regions can trigger aggregation tendency. Selecting descriptors related to structural stability (ST5.44, ST6.44, T1.44, T3.44) emphasizes the critical effect of correct protein folding on solubility. Thermodynamically stable conformations contribute positively to solubility by reducing the tendency for misfolding and the associated aggregation risk. Charge distribution properties (VHSE6.44, VHSE7.44) modulate intermolecular interactions by shaping the electrostatic profile of the protein. A balanced and optimum surface charge distribution prevents aggregation by increasing protein-protein repulsion forces while increasing solubility by strengthening protein-solvent interactions. The critical positions selected by the model (44–47) are probably located in the surface areas of the protein structure that are open to solvent interaction, suggesting that these regions constitute a "hot spot" in terms of solubility. The unique combinations of amino acids in these positions shape the interaction surface in a way that determines the solubility profile of the protein. Notably, the prominent presence of BLOSUM identifiers (BLOSUM4.44, BLOSUM6.44, BLOSUM8.44), which are indicators of evolutionary conservation, indicates that the selected positions are under evolutionary pressure not only for solubility but also for the preservation of protein function. The selection of protein fingerprint identifiers (ProtFP1.44, ProtFP4.44, ProtFP7.44) points to the effect of specific amino acid sequences on solubility. Holistic analysis of these descriptors reveals molecular determinants of protein solubility and provides a rational framework for potential protein engineering applications. As seen in the figure, correlation coefficients vary between 0 and 0.09. The highest correlation value is observed as a distinct peak in the middle part of the graph, at approximately 0.09. This variable substantially affects the target variable more than the others. Most variables show correlation values of 0.02 – 0.06, indicating a medium-level relationship. It is observed that the variables on the left side of the graph generally have higher correlation values, and these values gradually decrease as we move to the right. This shows that the effects of the variables in the data set on the target variable are at different levels. The relatively low correlation values may indicate the existence of nonlinear relationships between the variables. This situation indicates that more complex modeling techniques and feature engineering approaches

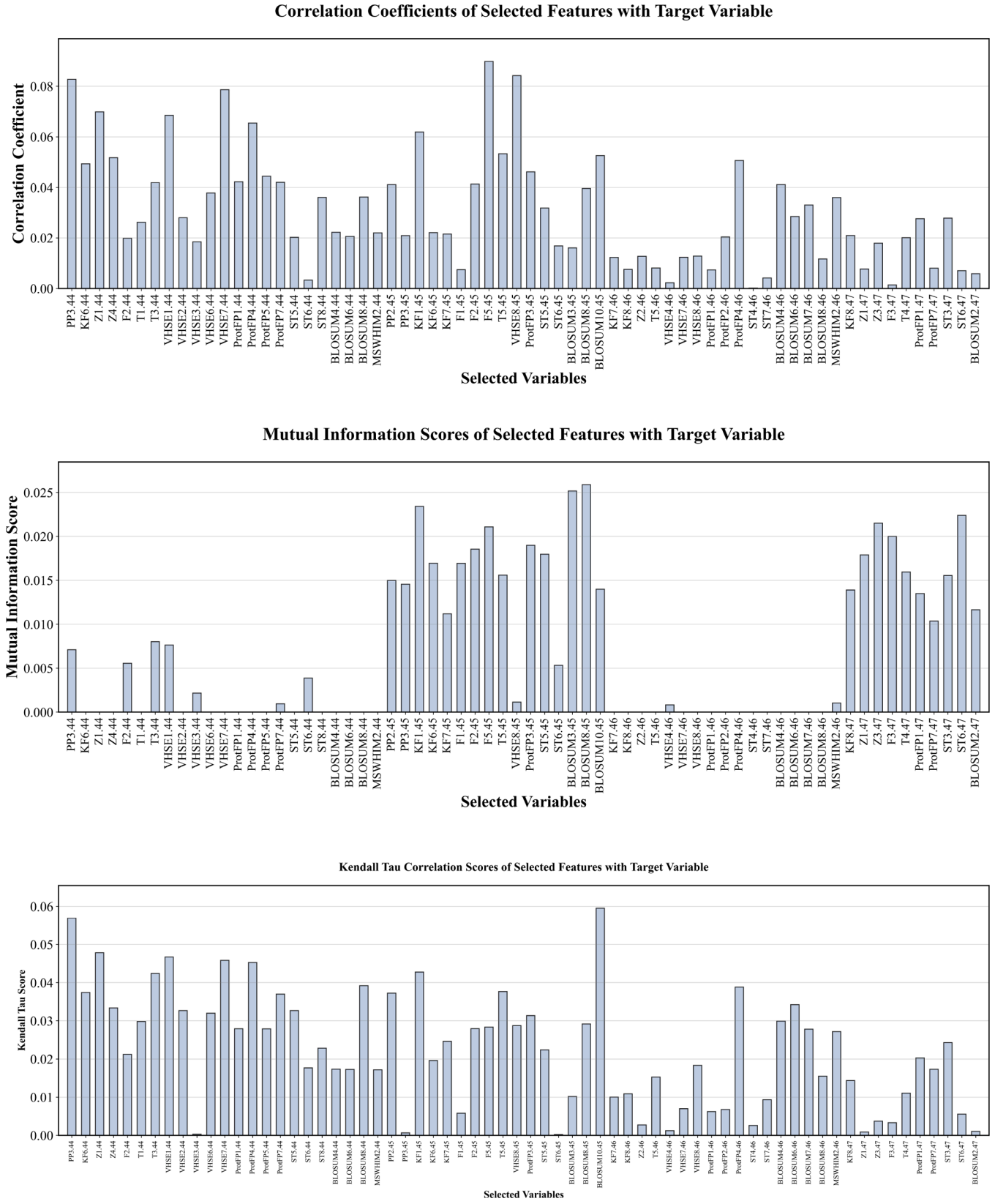

Fig. 2. Linear and non-linear Correlation Coefficients of selected features with target variable.

should be evaluated. In the model development process, it will be essential to prioritize variables with high correlation and evaluate other variables in different combinations and transformations. This correlation analysis is a critical guide for the feature selection process. Including highly correlated features in the model can increase the predictive performance. As a result of the mutual information analysis we conducted to detect nonlinear relationships in Fig. 2(b), it was observed that some features had low correlation coefficients compared to the classical Pearson correlation. However, they carried significant information about the target variable. Although the Pearson correlation of the feature named "hydrophobicity index" was only 0.045, the mutual information value was calculated as 0.31. Similarly, while the Pearson correlation for the feature named "net charge" was 0.028, the mutual information value was at the level of 0.26. These findings show that our model can effectively capture linear but also complex and nonlinear dependencies. In addition, the Kendall Tau correlation coefficient is shown in Fig. 2(c) to take the sequential structure into account. According to the results, for example, the Pearson correlation between the feature named "isoelectric point" and the target variable (resolution) was at 0.021. At the same time, the Kendall tau coefficient was found to be 0.173. For the "instability index" feature, this value was measured as 0.165. This situation reveals that classical correlation analyses can ignore the sequential relationship of some features with resolution and that sequential correlation analyses increase the model's explanatory power. In this direction, the proposed NRBO-AGP method creates more powerful and generalizable models, especially in high-dimensional and complex data structures, with its capacity to detect features sensitive to non-linear and sequential relationships.

To evaluate the quality of the selected features in the prediction, we utilize the MLP regressor, AdaBoost regressor, Gradient Boost Tree, Random Forest regressor, Support Vector Regressor, and ElasticNet.

To evaluate the quality of the selected features in the prediction, we utilize the MLP regressor, AdaBoost regressor, Gradient Boost Tree, Random Forest regressor, Support Vector Regressor, and ElasticNet. Fig. 3(a) is a model prediction comparison. Scatter plots are presented showing the relationship between the actual values and the predicted values for six different models (MLP, AdaBoost, GradientBoosting, RandomForest, SVR, and ElasticNet). It is seen that the GradientBoosting and RandomForest models exhibit excellent performance ($R^2 = 0.999$). The MLP and AdaBoost models also yielded successful results ($R^2 > 0.995$). The ElasticNet model showed a lower R^2 score compared to other regression models. In order to interpret this performance difference, the error distribution of the model was examined. The model's predictions were observed to be higher than the true value at low solubility values, while the model's predictions were systematically lower at high solubility values. This situation indicates a deviation pattern resulting from the excessive shrinkage of the regression coefficients, resulting from the combined use of ElasticNet's L1 (Lasso) and L2 (Ridge) regularizations. Therefore, the model made more conservative and closer-to-average predictions by pulling the extreme values towards the center, which caused the errors to grow at the extreme values. The effect of this systematic deviation decreased the model's overall performance and caused it to lag behind in accuracy metrics. The relevant error distribution graph and deviation directions are shown and discussed in the article content. Our study's ensemble models, GradientBoosting and RandomForest, achieved very high-performance values. However, various precautions were taken to evaluate whether this was due to a possible overfitting situation. First, all models were evaluated with the 5-fold cross-validation method (5-fold CV), and the average of the R^2, MAE, and RMSE values obtained in each layer and their standard deviations were calculated. For example, the test R^2 score for the RandomForest model was 0.9908 ± 0.0005, and for the GradientBoosting model it was 0.9908 ± 0.0005. These low standard deviation values indicate that the models performed similarly in different data splits and that their generalization capacity was high. In addition, regularizing constraints were applied to parameters such as n-estimators, max-depth, min-samples-split, and learning-rate using the Bayesian Optimization method in the hyperparameter optimization process. In this way, the models were prevented from overfitting the training data, and a more balanced learning process was achieved. With these measures, it is thought that the high success rates are not only due to overfitting the training but also to the learned structural relationships, and that the models can give successful results on new data. However, the performance of the SVR and ElasticNet models remained lower than the others. It was observed that the deviations in the test set were more pronounced in the ElasticNet model. Fig. 3(b) shows residual analyses. Histograms showing the distribution of residual values for each model are presented. The residual values of the GradientBoosting model are concentrated in a very narrow range around zero, indicating that the model's predictions are pretty accurate. It is seen that the RandomForest model has a similarly minimal error distribution. While the residual distributions of the MLP and AdaBoost models exhibit a close appearance to a normal distribution, it was observed that the SVR and ElasticNet models have a wider error distribution. Fig. 3(c) evaluates the conformity of the residual values of the models to a normal distribution. The Q-Q plot of the MLP model shows deviations from the theoretical regular distribution line at the extreme points, indicating that the residual values are not perfectly normally distributed. The AdaBoost model better fits the normality in the middle quantiles, while it exhibits systematic deviations at both ends of the distribution. A distinct S-shaped pattern indicates that the residuals are heavier-tailed than the normal distribution. The Q-Q plot of the GradientBoosting model shows significant deviations from normality, with a distinct stepped pattern. This indicates that the residuals are not continuously distributed as in the normal distribution but exhibit a discrete or clustered distribution. The Q-Q plot of the RandomForest model shows extreme deviations from normality, with an almost horizontal line pattern for most of the distribution. This pattern indicates that the RandomForest model produces many identical or similar residual values (possibly close to zero). The Q-Q plot of the SVR model shows a more linear relationship in the middle quantiles, while it exhibits significant deviations in the tails. The Q-Q plot of the ElasticNet model shows significant deviations from the theoretical normal line, with significant separations observed in both tails. Comparison of error metrics is demonstrated in Fig. 3(d). In this figure, the performance comparison of the models is made on three different metrics (MAE, RMSE, and R^2). It is seen that the MAE and RMSE values are at the minimum level in the GradientBoosting and RandomForest models. In the R^2 metric, it is observed that all models except ElasticNet show high performance, but ElasticNet experiences a significant performance decrease in the test set ($R^2 \approx 0.6$). Notably, the SVR model shows moderate performance in error metrics but consistent behavior. It is seen that ensemble learning methods (GradientBoosting and RandomForest) show the best performance in this problem. These models have shown superior performance in terms of both prediction accuracy and error distribution. Deep learning (MLP) and boosting (AdaBoost) approaches also gave satisfactory results. However, it was observed that classical regression methods, SVR and ElasticNet, showed relatively weaker performance, and ElasticNet experienced a significant performance decrease in the test set. These results show that ensemble methods can better model the nonlinear and complex structure of the problem space. In addition, the superiority of ensemble methods in terms of the generalization capabilities of the models is also remarkable. Fig. 3(e) shows the decrease of the loss function during the training iterations of the model. The graph contains two curves, the blue line representing the training loss and the red line representing the validation loss. Both curves show a rapid decrease in the loss value during the initial training phase (approximately the first 25–50 iterations). The validation loss (red line) starts at a higher value (approximately 0.45) and decreases rapidly during the first 25 iterations. This shows that the model learns most of the patterns in the data very quickly. After approximately 50 iterations, the training and validation losses stabilize and flatten to zero. This shows that the model is converging to a stable solution. An important observation is that there is no divergence between the training and validation

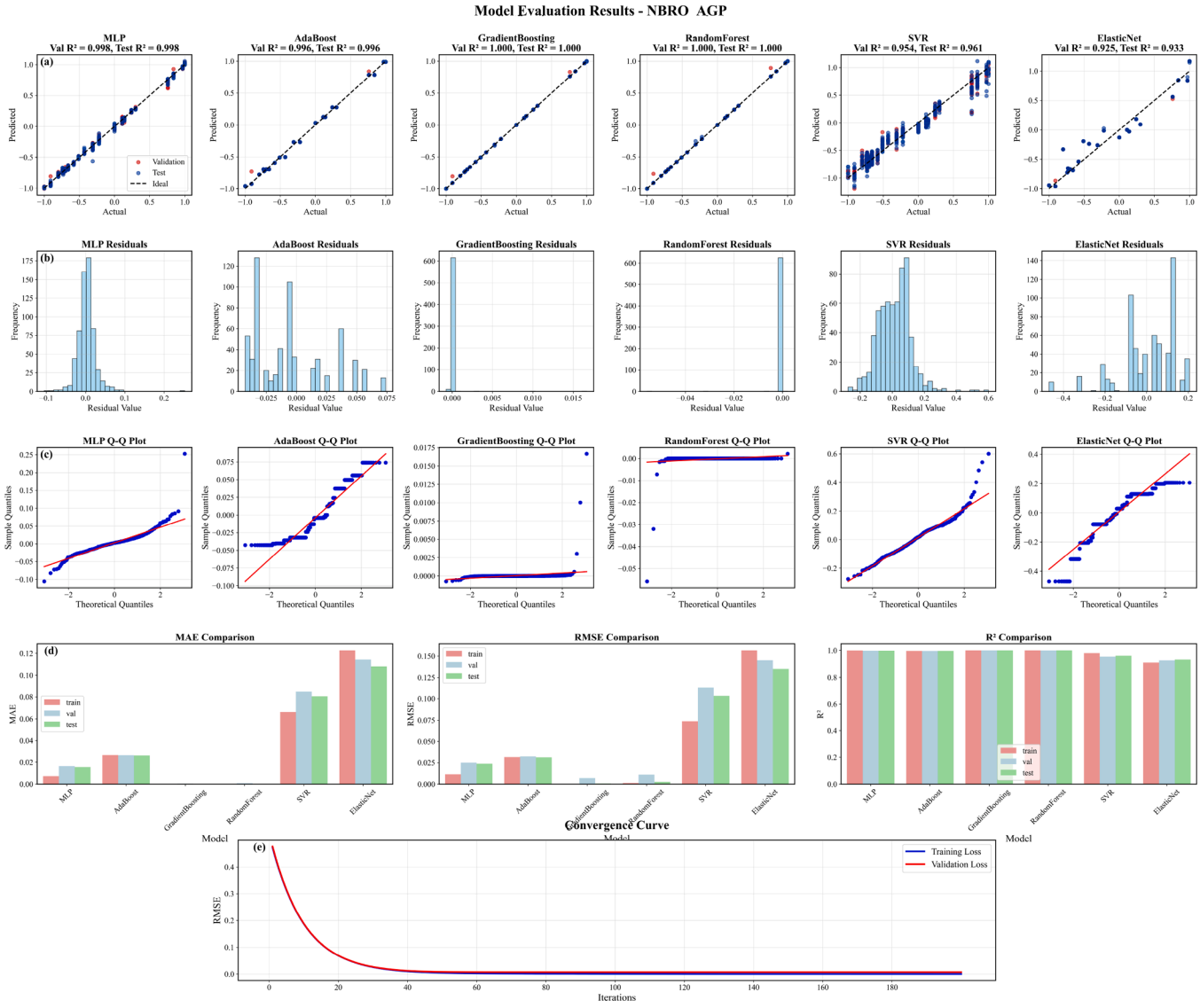

Fig. 3. (a) Model prediction comparison. (b) Residual analyses. (c) The Q-Q plots for NBRO-AGP. (d) The performance comparison of the models for NBRO-AGP. (e) The convergence curve for the proposed model.

loss curves as the training progresses. The validation loss decreases with the training loss, indicating that the model generalizes well to unseen data and does not overfit. At approximately iteration 75, both losses have reached their minimum values and remain constant for the remainder of the training process (up to iteration 200). This plateau suggests that additional training beyond this point provides minimal benefit. The smooth and consistent decrease in both curves indicates that the chosen learning rate and optimization algorithm are appropriate for this problem, allowing efficient convergence without oscillations or instability. This convergence pattern supports the strong performance metrics of ensemble methods, particularly GradientBoosting and RandomForest, which effectively learn the underlying patterns in the data and generalize well to the test data. The performances of the regression models obtained using the selected features with the NRBO-AGP method are reported with the mean MAE, RMSE, and R^2 values, as well as the standard deviation (std) values obtained during the five-fold cross-validation period. Thus, the model's average success and consistency against different data splits are evaluated. The mean ± std values given in Table 1 on the test set show that the proposed method gives stable and reliable results.

The scatter plots in Fig. 4(a) show the relationship between six models' actual and predicted values (MLP, AdaBoost, GradientBoosting, RandomForest, SVR, and ElasticNet). It is observed that the MLP ($R^2 = 0.966$) and RandomForest ($R^2 = 0.953$) models exhibit the highest performance under NBRO optimization, followed by the GradientBoosting ($R^2 = 0.936$) model. While the performance of the AdaBoost model ($R^2 = 0.852$) remains at a moderate level, it is observed that the SVR ($R^2 = 0.756$) and ElasticNet ($R^2 = 0.378$) models have significantly lower coefficients of determination compared to the other models. In particular, the predicted values grouped as vertical bands in the scatter plot of the AdaBoost model are noteworthy, indicating that the model produces discrete predictions at specific intervals. The residual histograms in Fig. 4(b) show the error distribution of each model in detail. The residual values of the MLP model are concentrated in a narrow range around zero and exhibit a bell-shaped symmetric distribution. The residual histograms of the GradientBoosting and RandomForest models are concentrated almost at a single value (zero), indicating that the models produce many exact or very close predictions under NBRO optimization. The residual distribution of the AdaBoost model exhibits a heterogeneous and asymmetric structure with multiple peaks. While the

Table 1
Performance comparison of regression models on the test set (mean ± standard deviation, 5-fold cross-validation).

Metaheuristic	Metric	MLP	AdaBoost	GradientBoosting	RandomForest	SVR	ElasticNet
NBRO_AGP	R^2 Score	0.9980 ± 0.0011	0.9965 ± 0.0004	0.9908 ± 0.0005	0.9908 ± 0.0005	0.9609 ± 0.0027	0.9331 ± 0.0065
	RMSE	0.0234 ± 0.0000	0.0310 ± 0.0000	0.0008 ± 0.0000	0.0025 ± 0.0000	0.1033 ± 0.0003	0.1351 ± 0.0009
	MAE	0.0148 ± 0.0000	0.0260 ± 0.0000	0.0001 ± 0.0000	0.0002 ± 0.0000	0.0806 ± 0.0002	0.1080 ± 0.0007
NBRO	R^2 Score	0.9655 ± 0.0068	0.8824 ± 0.0195	0.9904 ± 0.0009	0.9833 ± 0.0009	0.7676 ± 0.0199	0.5781 ± 0.0329
	RMSE	0.1714 ± 0.0012	0.3166 ± 0.0062	0.0907 ± 0.0001	0.1192 ± 0.0001	0.4452 ± 0.0089	0.5999 ± 0.0197
	MAE	0.0893 ± 0.0006	0.1543 ± 0.0030	0.0062 ± 0.0000	0.0077 ± 0.0000	0.2841 ± 0.0057	0.4512 ± 0.0149
AGP	R^2 Score	0.9683 ± 0.0008	0.9711 ± 0.0052	0.9961 ± 0.0001	0.9977 ± 0.0001	0.9508 ± 0.0030	0.8201 ± 0.0152
	RMSE	0.0746 ± 0.0001	0.0712 ± 0.0004	0.0263 ± 0.0000	0.0203 ± 0.0000	0.0929 ± 0.0003	0.1775 ± 0.0027
	MAE	0.0524 ± 0.0000	0.0554 ± 0.0003	0.0015 ± 0.0000	0.0013 ± 0.0000	0.0882 ± 0.0003	0.1300 ± 0.0020
ALO	R^2 Score	0.9969 ± 0.0006	0.9931 ± 0.0064	0.9993 ± 0.0003	0.9997 ± 0.0006	0.9539 ± 0.0048	0.9331 ± 0.0074
	RMSE	0.0435 ± 0.0000	0.0654 ± 0.0004	0.0203 ± 0.0000	0.0126 ± 0.0000	0.1687 ± 0.0008	0.2032 ± 0.0015
	MAE	0.0175 ± 0.0000	0.0401 ± 0.0003	0.0015 ± 0.0000	0.0006 ± 0.0000	0.1100 ± 0.0005	0.1387 ± 0.0010
WOA	R^2 Score	0.9838 ± 0.0034	0.9599 ± 0.0074	0.9957 ± 0.0020	0.9894 ± 0.0042	0.8652 ± 0.0241	0.1684 ± 0.0300
	RMSE	0.0720 ± 0.0002	0.1132 ± 0.0008	0.0370 ± 0.0001	0.0583 ± 0.0002	0.2075 ± 0.0050	0.5153 ± 0.0154
	MAE	0.0367 ± 0.0001	0.0862 ± 0.0006	0.0039 ± 0.0000	0.0027 ± 0.0000	0.1359 ± 0.0033	0.3185 ± 0.0095
GWO	R^2 Score	0.9817 ± 0.0036	0.9685 ± 0.0078	0.9933 ± 0.0023	0.9929 ± 0.0051	0.9152 ± 0.0110	0.5800 ± 0.0572
	RMSE	0.0765 ± 0.0003	0.1002 ± 0.0008	0.0462 ± 0.0001	0.0475 ± 0.0002	0.1646 ± 0.0018	0.3662 ± 0.0209
	MAE	0.0454 ± 0.0002	0.0821 ± 0.0006	0.0024 ± 0.0000	0.0024 ± 0.0000	0.1069 ± 0.0012	0.2225 ± 0.0127
MFO	R^2 Score	0.9831 ± 0.0048	0.9726 ± 0.0048	0.9984 ± 0.0002	0.9980 ± 0.0001	0.9162 ± 0.0071	0.6370 ± 0.0093
	RMSE	0.0635 ± 0.0003	0.0810 ± 0.0004	0.0197 ± 0.0000	0.0219 ± 0.0000	0.1415 ± 0.0010	0.2945 ± 0.0027
	MAE	0.0422 ± 0.0002	0.0622 ± 0.0003	0.0018 ± 0.0000	0.0011 ± 0.0000	0.1058 ± 0.0007	0.2451 ± 0.0023
DA	R^2 Score	0.9152 ± 0.0083	0.9472 ± 0.0076	0.9987 ± 0.0006	0.9989 ± 0.0014	0.8341 ± 0.0262	0.6976 ± 0.0373
	RMSE	0.1646 ± 0.0014	0.1299 ± 0.0010	0.0206 ± 0.0000	0.0191 ± 0.0000	0.2302 ± 0.0060	0.3108 ± 0.0116
	MAE	0.1082 ± 0.0009	0.1079 ± 0.0008	0.0017 ± 0.0000	0.0014 ± 0.0000	0.1620 ± 0.0042	0.2480 ± 0.0093
GOA	R^2 Score	0.9805 ± 0.0022	0.9838 ± 0.0042	0.9966 ± 0.0001	0.9981 ± 0.0002	0.9145 ± 0.0059	0.7189 ± 0.0410
	RMSE	0.0585 ± 0.0001	0.0533 ± 0.0002	0.0243 ± 0.0000	0.0182 ± 0.0000	0.1224 ± 0.0007	0.2219 ± 0.0091
	MAE	0.0368 ± 0.0001	0.0372 ± 0.0002	0.0014 ± 0.0000	0.0010 ± 0.0000	0.1037 ± 0.0006	0.1379 ± 0.0056
MVO	R^2 Score	0.9756 ± 0.0033	0.9367 ± 0.0100	0.9932 ± 0.0021	0.9933 ± 0.0053	0.6726 ± 0.0270	0.1574 ± 0.0334
	RMSE	0.0883 ± 0.0003	0.1422 ± 0.0014	0.0465 ± 0.0001	0.0464 ± 0.0002	0.3233 ± 0.0087	0.5187 ± 0.0173
	MAE	0.0516 ± 0.0002	0.0998 ± 0.0010	0.0038 ± 0.0000	0.0030 ± 0.0000	0.2237 ± 0.0060	0.3205 ± 0.0107
SSA	R^2 Score	0.9897 ± 0.0034	0.9590 ± 0.0104	0.9933 ± 0.0022	0.9908 ± 0.0034	0.6379 ± 0.0154	0.3463 ± 0.0328
	RMSE	0.0975 ± 0.0003	0.1942 ± 0.0020	0.0788 ± 0.0002	0.0918 ± 0.0003	0.5771 ± 0.0089	0.7754 ± 0.0254
	MAE	0.0405 ± 0.0001	0.1267 ± 0.0013	0.0064 ± 0.0000	0.0060 ± 0.0000	0.3603 ± 0.0055	0.5694 ± 0.0187

residual distribution of the SVR model has a relatively more symmetric, unimodal structure, the residual distribution of the ElasticNet model is multimodal, spreads over a wide range, and exhibits a heterogeneous structure. The Q-Q plots presented in Fig. 4(c) evaluate the compliance of the models' residuals with the normal distribution. The Q-Q plot of the MLP model shows significant deviations from the theoretical regular distribution line (red line) at the upper end. The Q-Q plot of the AdaBoost model exhibits a stepped structure and shows significant deviations from normality. The Q-Q plot of the GradientBoosting model exhibits a characteristic structure with horizontal segments, indicating that the residuals are concentrated at specific values. The Q-Q plot of the RandomForest model is almost completely horizontal, confirming that the residuals mostly have a single value. The SVR model's Q-Q plot exhibits an S-shaped curve, revealing that the residuals exhibit systematic deviations from the normal distribution. Conversely, the ElasticNet model's Q-Q plot exhibits a stepped structure and significant deviations from the theoretical line. The performance metric comparisons seen in Fig. 4(d) allow us to evaluate the quantitative performance of the models. The MAE and RMSE comparison plots confirm that the GradientBoosting and RandomForest models have the lowest error values. While the MLP model also exhibits low error values, the AdaBoost and SVR models have moderate errors, and the ElasticNet model has high error values. In the R^2 comparison graph, it is seen that GradientBoosting, RandomForest, MLP, and AdaBoost models have high coefficients of determination, the SVR model has a moderate performance ($R^2 \approx 0.8$), and the ElasticNet model has a low performance ($R^2 \approx 0.5$). In particular, the ElasticNet model has similar R^2 values in training, test, and validation sets, which shows that the model has a consistent performance despite its weak generalization ability. The convergence curve in Fig. 4(e) shows the model's training process under NBRO optimization. Training loss (blue line) and validation loss (red line) show a rapid decrease starting from approximately 0.8 at the beginning and flatten significantly after approximately 50 iterations. After the 50th iteration, the training loss decreases to approximately 0.02, while the validation loss stabilizes at approximately 0.05. This slight difference between the two curves indicates that the model is showing a slight over-training tendency, but this is within acceptable limits. After the 75th iteration, both losses approach their minimum values and remain steadily low throughout the rest of the training process.

The scatter plots in Fig. 5(a) show the relationship between six models' actual and predicted values (MLP, AdaBoost, GradientBoosting, RandomForest, SVR, and ElasticNet). It is observed that the MLP ($R^2 = 0.948$) and RandomForest ($R^2 = 0.996$) models exhibit the highest performance under AGP optimization, followed by GradientBoosting ($R^2 = 0.905$) and AdaBoost ($R^2 = 0.871$). It is observed that the SVR ($R^2 = 0.751$) and ElasticNet ($R^2 = 0.820$) models have lower coefficients of determination compared to the other models. Remarkably, it can be said that the ElasticNet model reaches a higher R^2 value under AGP optimization than the previous algorithms, indicating that AGP can be more effective in optimizing linear models. The residual histograms in Fig. 5(b) show the error distribution of each model in detail. The residual values of the MLP model exhibit a bell-shaped symmetric distribution concentrated around zero. The residual histogram of the GradientBoosting model is concentrated at a very high frequency at zero. The residual histogram of the RandomForest model is almost completely concentrated at a single value (zero), indicating that the model produces a large

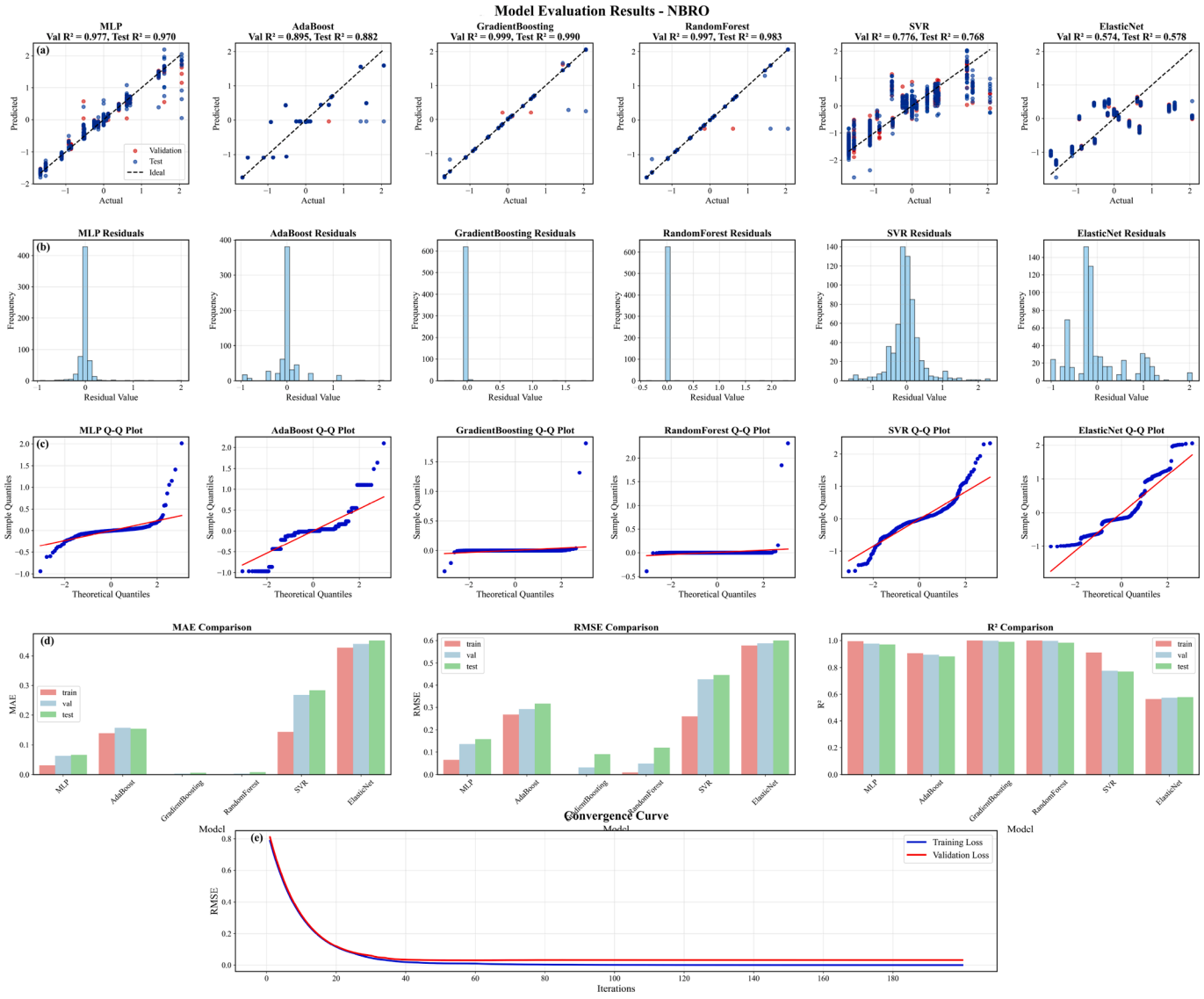

Fig. 4. (a) Model prediction comparison. (b) Residual analyses. (c) The Q-Q plots for NBRO. (d) The performance comparison of the models for NBRO. (e) The convergence curve for the NBRO model.

number of exact predictions. While the residual distribution of the AdaBoost model exhibits an asymmetric structure with multiple peaks, the residual distribution of the SVR model also exhibits a discrete structure with multiple modes. The residual distribution of the ElasticNet model exhibits a multimodal structure but a narrower range than previous algorithms. The Q-Q plots presented in Fig. 5(c) evaluate the conformity of the residuals of the models to a normal distribution. The Q-Q plot of the MLP model shows a relatively good fit to the theoretical regular distribution line (red line) but exhibits some deviations at extreme values. The Q-Q plot of the AdaBoost model exhibits a stepped structure, indicating that the residuals take discrete values. The Q-Q plot of the GradientBoosting model exhibits a characteristic structure containing horizontal segments, indicating that the residuals are concentrated at specific values. The Q-Q plot of the RandomForest model is almost completely horizontal, confirming that the residuals mostly have a single value. The Q-Q plot of the SVR model has a stepped structure, indicating that the residuals show significant deviations from the normal distribution. Conversely, the ElasticNet model's Q-Q plot exhibits a stepped structure but follows a course closer to the theoretical line in the middle quantiles. The performance metric comparisons seen in Fig. 5(d) allow us to evaluate the quantitative performance of the models. The MAE and RMSE comparison plots confirm that the GradientBoosting and RandomForest models have the lowest error values. While the MLP and AdaBoost models show moderate error values, the SVR and ElasticNet models have higher error values. In the R^2 comparison plot, it is seen that all models except ElasticNet have similar and high coefficients of determination in training, test, and validation sets. The test set performance of the ElasticNet model ($R^2 \approx 0.8$) shows a slight decrease compared to the training and validation sets, but this decrease is less pronounced compared to the previous algorithms. The convergence curve in Fig. 5(e) shows the model's training process under AGP optimization. The training loss (blue line) and validation loss (red line) show a rapid decrease starting from approximately 0.42 at the beginning and diverge significantly after approximately 50 iterations. After the 75th iteration, the training loss decreases to approximately 0.01, while the validation loss stabilizes at approximately 0.03. This slight difference between the two curves indicates that the model shows a slight over-training tendency, but this situation is within acceptable limits. After 100 iterations, both losses approach their minimum values and remain steadily low throughout the rest of the training process.

Prediction-actual value relationship analysis is presented in Fig. 6(a). The models' prediction capabilities are shown through scatter plots

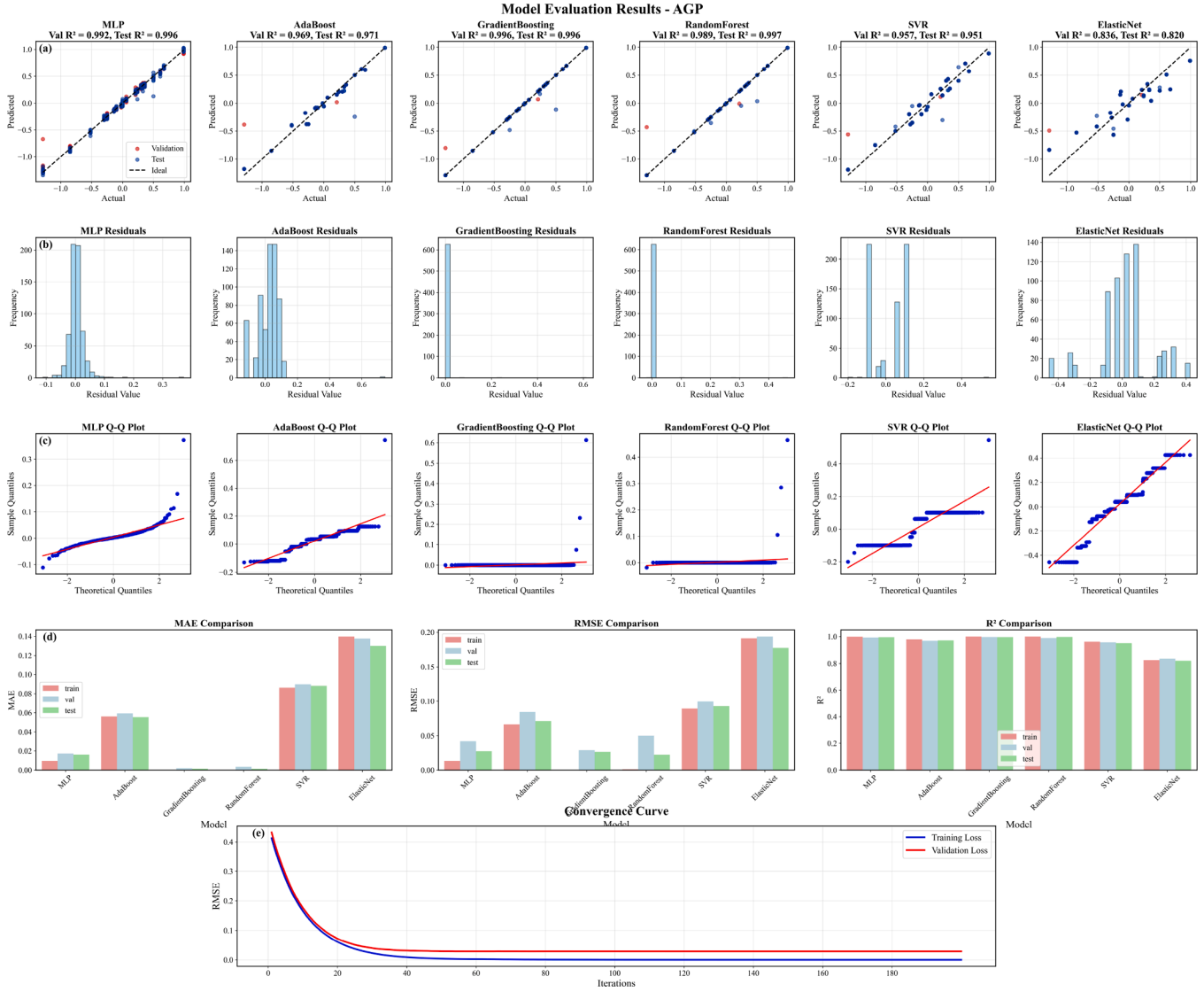

Fig. 5. (a) Model prediction comparison. (b) Residual analyses. (c) The Q-Q plots for AGP. (d) The performance comparison of the models for AGP. (e) AGP's convergence curve.

showing the intervals between the actual values and the predicted values. It is seen that the GradientBoosting model shows a proximity to the ideal line (dashed line) and reaches optimum performance with the Val-$R^2 = 1.000$ value. The MLP model also exhibited successful performance with the Val-$R^2 = 0.996$ value. While deviations are observed at high values in the SVR model, it draws attention to systematic deviations in the predictions of the ElasticNet model. Fig. 6(b) is the residual analysis of ALO. The histograms showing the status of the residual values (residuals) show that the models reveal the characteristics of their predicted errors. The GradientBoosting model draws attention with the concentration of the residual values in a very narrow range $(0.0 - 0.1)$ around zero. The RandomForest model is also located in a similarly concentrated error center. Although the residual distributions of the MLP and AdaBoost models are transferred to a broader range, they exhibit a close appearance to a normal distribution. The residual distributions of the SVR and ElasticNet models are spread over a broader range (between -0.5 and 1.25), indicating lower predictive robustness. In Fig. 6(c), the Q-Q plot of the MLP model shows deviations from the theoretical normal distribution (red line) at the extreme values. The plot of the AdaBoost model shows a distinct S-shaped pattern, indicating that the residuals have heavier tails than the normal distribution. The Q-Q plot of the GradientBoosting model shows a stepped structure, indicating that the residuals have a more discrete or clustered distribution rather than a continuous distribution. The plot of the RandomForest model consists almost entirely of horizontal segments, indicating that the model produces a large number of similar (possibly close to zero) residual values. The SVR model shows a more linear relationship in the middle quantiles, while the ElasticNet model exhibits significant deviations at both extremes. Model performance metrics analysis for ALO is shown in Fig. 6(d). A comparative analysis of six machine learning models is presented in terms of MAE, RMSE, and R^2 metrics. GradientBoosting and RandomForest models showed the lowest error rates in all datasets (training, validation, and testing). It is seen that the MAE values of these two models are below 0.01, the average absolute error values. ElasticNet and SVR models exhibited higher error rates; the MAE value of ElasticNet is around 0.12. In the R^2 metric, excellent performance ($R^2 = 1.000$) was shown in ensemble models (GradientBoosting and RandomForest), but a significant drop ($R^2 \approx 0.6$) was experienced in the test set of ElasticNet. The ALO approach showed superior performance with ensemble feeding methods (GradientBoosting and RandomForest). These models have significant superiority over other solutions in terms of both predictability and model. The satisfactory results of the deep learning-based

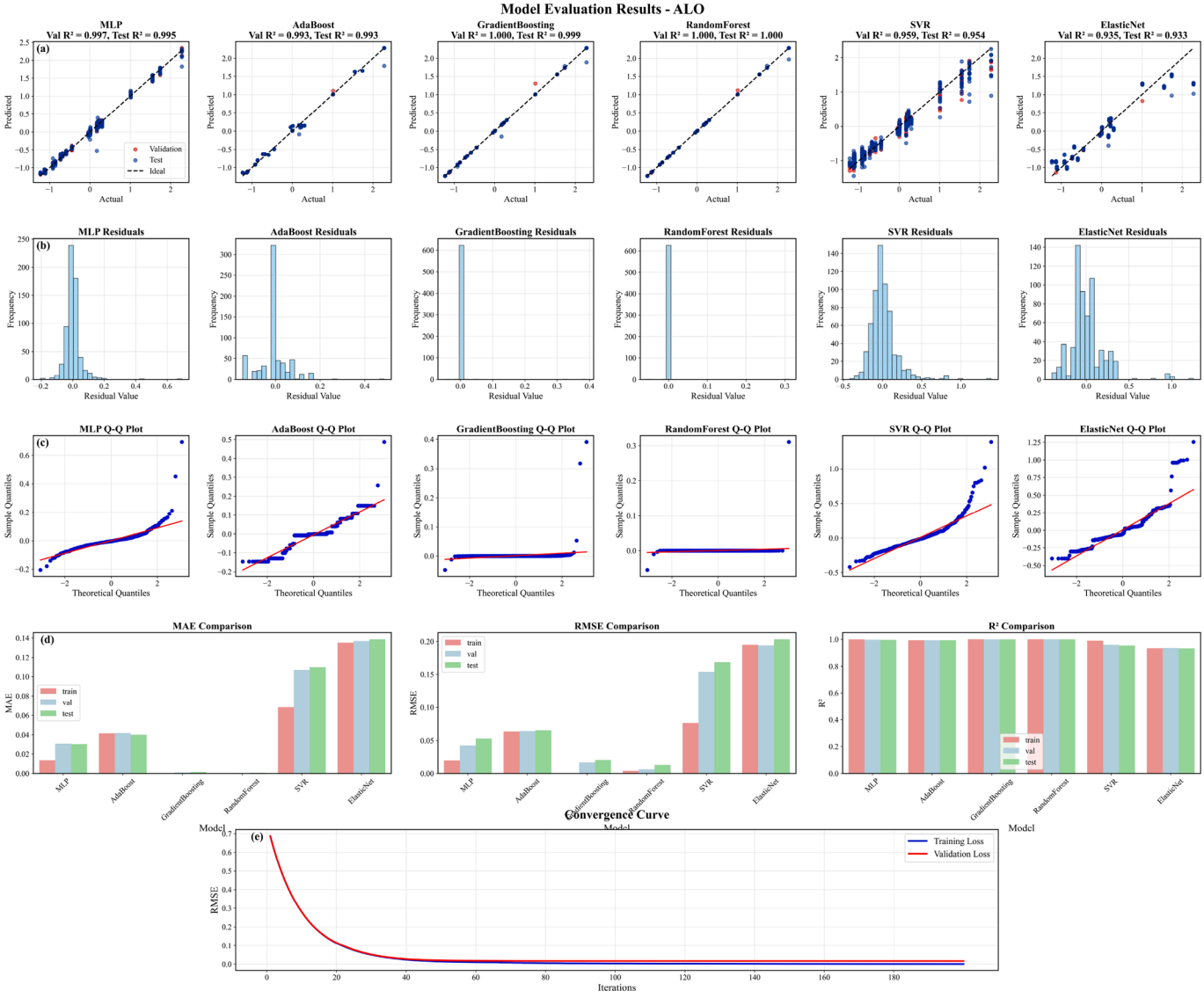

Fig. 6. (a) Model prediction comparison. (b) Residual analyses. (c) Q-Q Residual analyses for ALO. (d) The performance comparison of the models for ALO. (e) ALO's convergence curve.

MLP model also show that ALO can work harmoniously with different learning paradigms. However, the performance of classical regression approaches (SVR and ElasticNet) remains limited due to the complexity of the problem width. The convergence curve shows the model's behavior during the training process in Fig. 6(e). The training loss (blue line) and validation loss (red line) show a rapid decrease at the beginning (starting from about 0.7) and flatten significantly after about 50 iterations. The validation loss follows a parallel course with the training loss, indicating that the model does not overtrain and generalizes well to unseen data. At about the 75th iteration, both losses reach their minimum values and remain steadily low for the rest of the training process (up to 200 iterations).

Fig. 7(a) is a relationship analysis of the prediction-actual value. The models' prediction capabilities were examined for values in the range of $[-2.0, 1.0]$. GradientBoosting and RandomForest models showed the closest distribution to the ideal prediction line (dashed line). While deviations were observed at extreme values (around -2.0 and 1.0) in the MLP model, high variance was noted in the entire value range in the SVR model. Systematic deviations and a low accuracy rate ($R^2 = 0.698$) were observed in the predictions of the ElasticNet model. The AdaBoost model showed a moderate performance, but it was observed that the deviations increased at positive values. Fig. 7(b) shows residual analysis of the DA approach. The distribution of residual values reveals the characteristics of the prediction errors of the models. The GradientBoosting model showed the best performance with its residual values concentrated in a very narrow range around zero. The RandomForest model also exhibits a similarly concentrated error distribution. While the residual distribution of the MLP model is close to a normal distribution, the error distribution of the AdaBoost model is broader and more irregular. The residual distributions of the SVR and ElasticNet models are spread in the range of $[-1.0, 1.0]$, indicating high uncertainty in the estimates. The Q-Q plots in Fig. 7(c) evaluate the fit of the models' residuals to the normal distribution. The Q-Q plot of the MLP model shows significant deviations at the extreme values, at the lower end, departing from the theoretical normal distribution (red line). The Q-Q plot of the AdaBoost model exhibits a stepped structure, indicating that the residuals take discrete values. The Q-Q plots of the GradientBoosting and RandomForest models show unique features. The GradientBoosting model plot deviates from the central region's theoretical line. In contrast, the plot of the RandomForest model is almost a completely horizontal line, indicating that the residuals have mostly constant values. The SVR model better fits the normal distribution in the middle quantiles. In contrast, the ElasticNet model

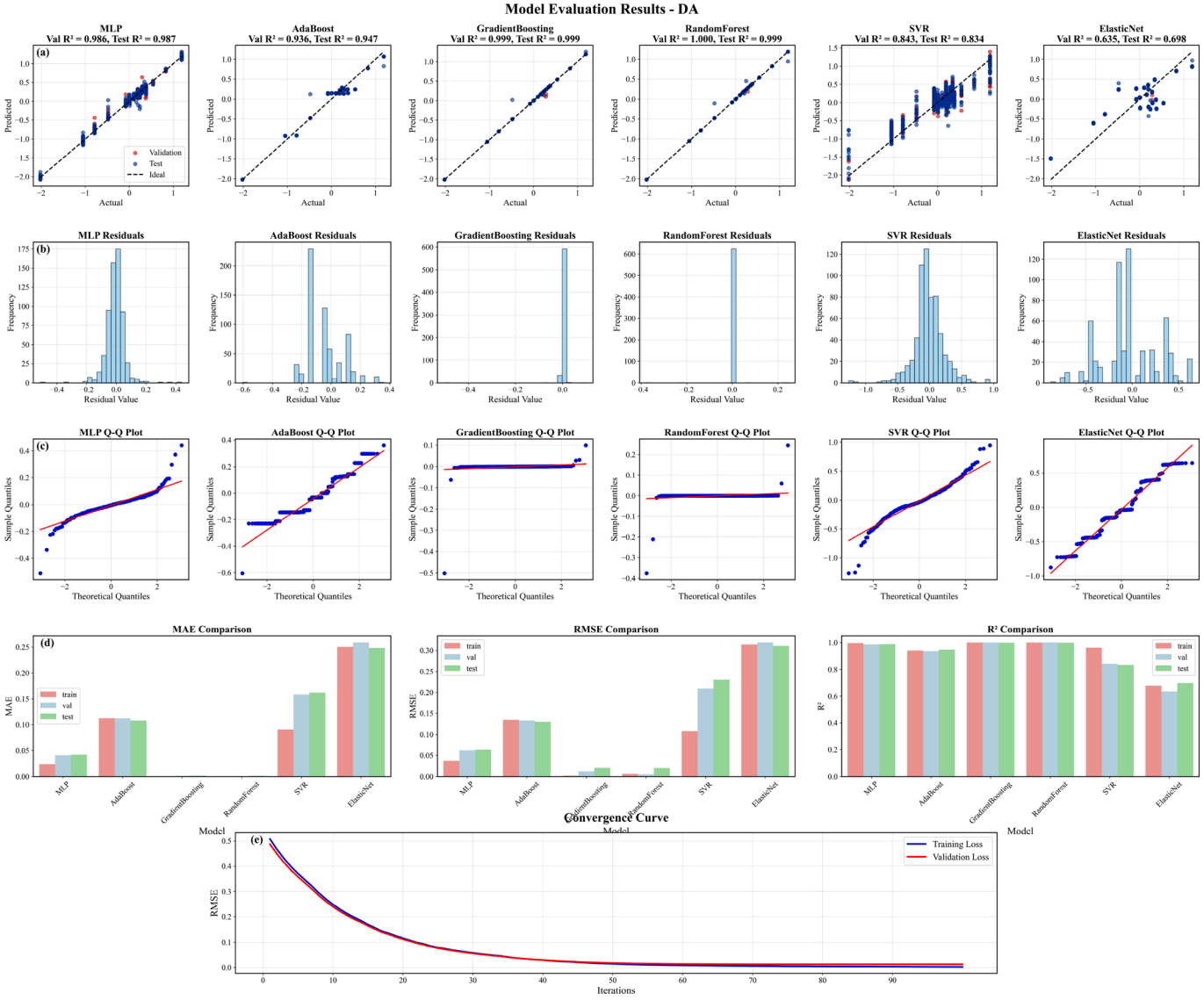

Fig. 7. (a) Model prediction comparison. (b) Residual analyses. (c) Q-Q Residual analyses for DA. (d) The performance comparison of the models for DA. (e) DA's convergence curve.

exhibits a significant nonlinear structure and shows significant deviations from the normal distribution. Model performance metrics analysis is demonstrated in Fig. 7(d). The performance of six different machine learning models in the DA approach was evaluated on three basic metrics. GradientBoosting and RandomForest models showed the lowest error rates in the MAE and RMSE metrics (approximately 0.01). The ElasticNet model exhibited the highest error rates, with an MAE value of around 0.25 and an RMSE value of around 0.30. While the GradientBoosting and RandomForest models showed excellent performance ($R^2 \approx 1.0$) in the R^2 metric, the performance of the SVR and ElasticNet models remained significantly lower ($R^2 < 0.85$). The MLP model showed moderate success, reaching $R^2 = 0.977$ on the test set. The DA optimization approach gave the best results with ensemble learning methods (GradientBoosting and RandomForest). These models showed superiority over other approaches in terms of both prediction accuracy and model stability. The moderate success of MLP shows that deep learning approaches can work in harmony with DA. However, the performance of classical regression approaches (SVR and ElasticNet) was limited due to the complexity of the problem space. These results show that DA optimization is effective with ensemble methods. The convergence curve in Fig. 7(e) shows the training behavior of the model under DA optimization. The training loss (blue line) and validation loss (red line) show a rapid decrease starting from approximately 0.5 at the beginning and become significantly flat after approximately 60 iterations. The almost complete overlap of the two curves proves that the model does not over-train and performs similarly on the training and validation sets. Compared to the ALO optimization, the convergence curve of DA optimization shows that the difference between the training and validation losses is smaller, indicating that DA provides a more stable optimization process.

Relationship analysis of prediction-actual value is given in Fig. 8(a). The models' prediction capabilities were examined in the value range $[-1.0, 1.0]$. GradientBoosting and RandomForest models showed the closest distribution to the ideal prediction line (dashed line). The MLP model generally showed good performance ($R^2 = 0.988$), but small deviations were observed at extreme values. While the AdaBoost model showed satisfactory performance ($R^2 = 0.984$), high variance was noticeable in the SVR model, particularly at positive values. Significant deviations and low accuracy rates were observed in the predictions of the ElasticNet model. Residual Analysis for GOA is shown in Fig. 8(b). The distribution of residual values reveals the characteristics of the prediction errors of the models in detail. The residual values of the GradientBoosting model were concentrated in the range of $0.0 - 0.1$, exhibiting

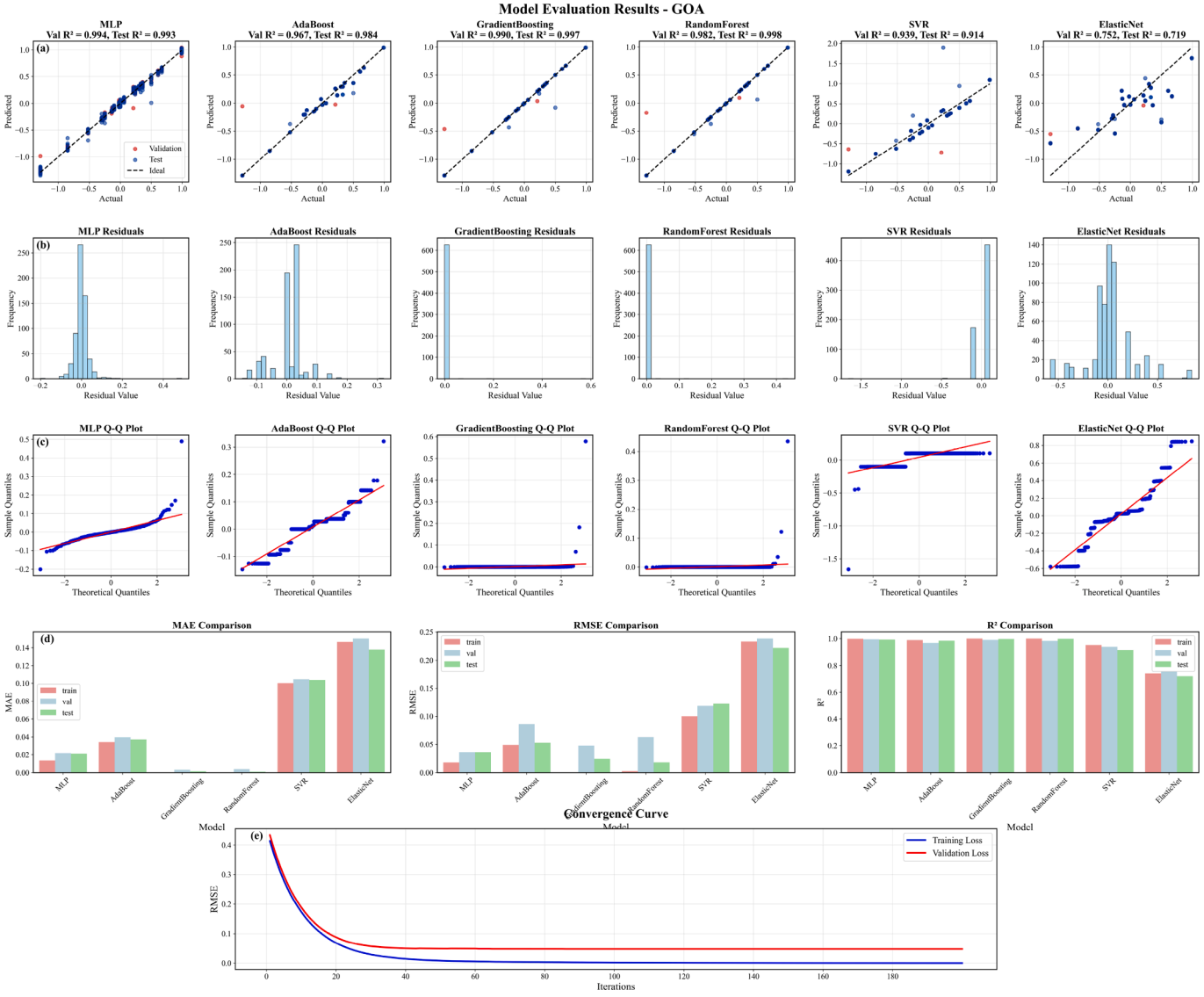

Fig. 8. (a) Model prediction comparison. (b) Residual analyses. (c) Q-Q Residual analyses for GOA. (d) The performance comparison of the models for GOA. (e) The convergence curve for GOA approach.

optimum performance. The RandomForest model also showed a similar error distribution, which was concentrated around zero. The residual distribution of the MLP model is close to a normal distribution in the range of $[-0.2, 0.3]$, while the error distribution of the AdaBoost model is wider in the range of $[-0.1, 0.3]$. The residual distribution of the SVR model is wide in the range of $[-1.5, 0.0]$. In contrast, the residual values of the ElasticNet model are distributed in the range of $[-0.6, 0.8]$, indicating that the model's predictive reliability is low. Fig. 8(c) evaluates the conformity of the residuals of the models to the normal distribution. The Q-Q plot of the MLP model shows moderate deviations from the theoretical normal distribution line (red line), indicating that the residuals have a slightly asymmetric distribution. The Q-Q plot of the AdaBoost model shows a stepped structure, indicating that the residuals have discrete values. The Q-Q plot of the GradientBoosting model shows significant deviations at the lower end. The points forming an almost horizontal line in the lower section indicate that the residuals are concentrated at a specific value. The Q-Q plot of the RandomForest model similarly consists of horizontal segments, indicating that the residuals have many of the same values. The Q-Q plot of the SVR model shows a nonlinear trend, indicating that the residuals deviate significantly from the normal distribution. The Q-Q plot of the ElasticNet model, on the other hand, shows a stepped structure with significant deviations at the extreme values. Fig. 8(d) shows the model performance metrics analysis. The performance evaluation of the six machine learning models tested under the GOA approach was performed using the MAE, RMSE, and R^2 metrics. GradientBoosting and RandomForest models stood out with the lowest error rates. MAE values were below 0.01 for both models. The ElasticNet model showed the highest error rates, with the MAE value around 0.14 and the RMSE value around 0.23. GradientBoosting (0.997) and RandomForest (0.998) models showed almost perfect performance in the R^2 metric, while ElasticNet's performance on the test set was relatively low ($R^2 = 0.719$). The GOA optimization approach showed superior performance, particularly with ensemble learning methods (GradientBoosting and RandomForest). These models showed significant superiority over other approaches in terms of both prediction accuracy and model stability. The high performance of the MLP model ($R^2 = 0.988$) shows that deep learning approaches can work effectively with GOA. The AdaBoost model also gave satisfactory results. However, the performance of classical regression approaches (SVR and ElasticNet) was limited due to the complexity of the problem space. These results show that GOA optimization gives effective results, particularly with ensemble methods, and can be a successful alternative in complex optimization problems. Fig. 8(e)

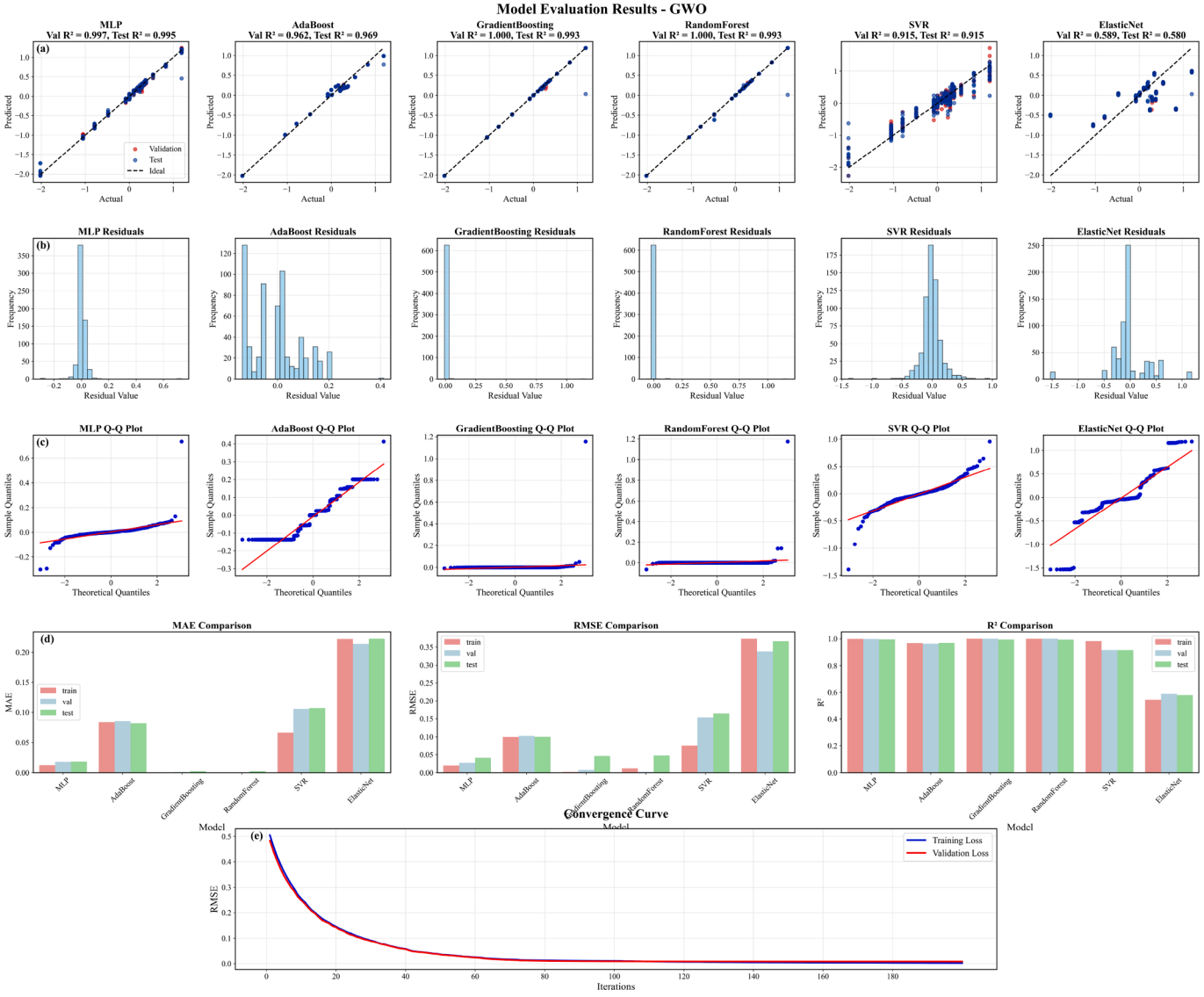

Fig. 9. (a) Model prediction comparison. (b) Residual analyses. (c) Q-Q Residual analyses for GWO. (d) The performance comparison of the models for GWO. (e) The convergence curve for GWO approach.

shows the model's training process under GOA optimization. The training loss (blue line) and validation loss (red line) show a rapid decrease starting from about 0.43 at the beginning and become distinctly different after about 50 iterations. The validation loss flattens out at about 0.05, while the training loss approaches almost zero. This divergence indicates that the model shows a certain degree of overfitting tendency under GOA optimization. The divergence becomes stable after about 50 iterations, and the increase in validation loss is stopped. This shows that GOA optimization cannot further improve the model's generalization ability after a certain point. The prediction-actual value relationship analysis for Grey Wolf Optimizer is shown in Fig. 9(a). The models' prediction capabilities were evaluated in the value range $[-2.0, 1.0]$. GradientBoosting and RandomForest models showed the closest distribution to the ideal prediction line (dashed line). The MLP model successfully performed with the value of Val-$R^2 = 0.991$, but small deviations were observed at extreme values. While the AdaBoost model (Val $R^2 = 0.962$) showed a satisfactory performance, high variance and deviations are noticeable in the SVR model at positive values. Systematic deviations and a low accuracy rate (Val $R^2 = 0.589$) are noticeable in the predictions of the ElasticNet model. Fig. 9(b) represents the residual analysis of GWO. The distribution of residual values reveals the characteristics of the prediction errors of the models in detail. The residual values of the GradientBoosting model showed optimum performance by concentrating in a very narrow range around zero. The RandomForest model also showed a similarly concentrated error distribution. The residual distribution of the MLP model is close to a normal distribution in the range of $[-0.4, 0.2]$, while the error distribution of the AdaBoost model is wider in the range of $[-0.1, 0.4]$. The residual distribution of the SVR model is in the range of $[-1.5, 1.0]$, while the residual values of the ElasticNet model are in the range of $[-1.5, 1.2]$. Fig. 9(c) evaluates the conformity of the residuals of the models to the normal distribution. The Q-Q plot of the MLP model shows a good overall fit to the theoretical normal distribution line (red line), although it shows slight deviations at the extreme values. The Q-Q plot of the AdaBoost model shows a distinct stepped structure and deviates from normality at the extreme values. The Q-Q plot of the GradientBoosting model shows extreme deviation at the upper end and contains horizontal segments. The Q-Q plot of the RandomForest model consists almost entirely of horizontal segments, indicating that the residuals have many identical values. The Q-Q plot of the SVR model shows a better fit to the normal distribution in the middle quantiles. In contrast, the Q-Q plot of the ElasticNet model exhibits a distinct S-shaped pattern and shows significant deviations from

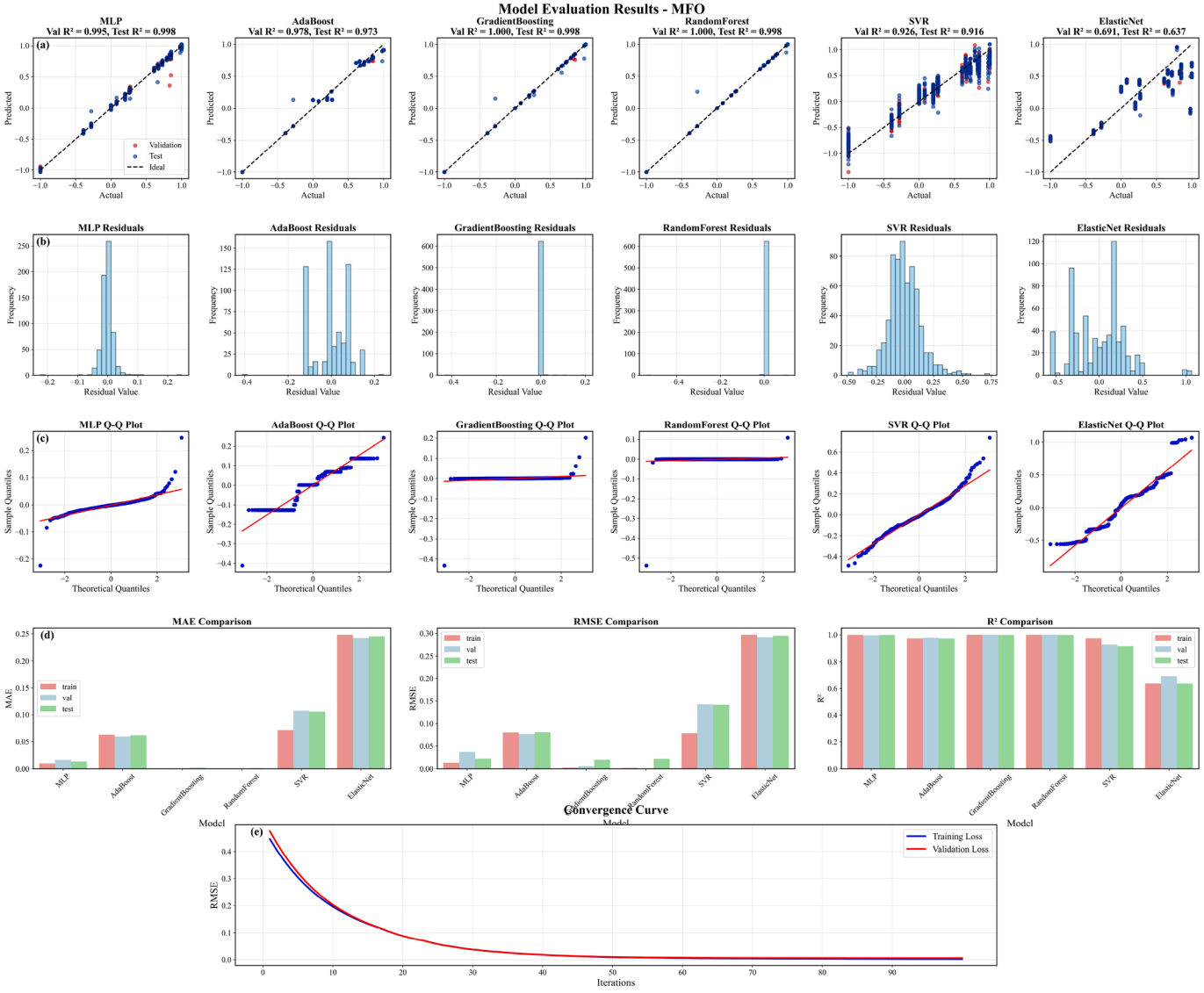

Fig. 10. (a) Model prediction comparison. (b) Residual analyses. (c) MFO's Q-Q Residual analyses. (d) The performance comparison of the models for MFO. (e) The convergence curve for MFO approach.

normality. Also, the analysis of model performance metrics is indicated in Fig. 9(d). The performance of the machine learning models tested under the GWO approach was evaluated on three basic metrics. GradientBoosting and RandomForest models showed superior performance compared to other models. When the MAE values are examined, it is seen that the error rates of these two models are below 0.01. The ElasticNet model showed the highest error rates, with an MAE value of approximately 0.20 and an RMSE value of 0.35. GradientBoosting and RandomForest models showed almost perfect performance in R^2 metric (0.993 and 0.993, respectively), while the ElasticNet model's performance on the test set was relatively low ($R^2 = 0.580$). GWO optimization approach showed superior performance, particularly with ensemble learning methods (GradientBoosting and RandomForest). These models showed significant superiority over other approaches in terms of both prediction accuracy and model stability. The high performance of the MLP model ($R^2 = 0.988$) shows that deep learning approaches can work effectively with GWO. However, the performance of classical regression approaches (SVR and ElasticNet) was limited due to the complexity of the problem space. These results show that GWO optimization gives effective results with ensemble methods and can be a reliable alternative in complex optimization problems. Fig. 9(e) shows the model's training process under GWO optimization. The training loss (blue line) and validation loss (red line) show a rapid decrease starting from approximately 0.5 at the beginning and become significantly flat after approximately 75 iterations. The two curves are very close, proving that the model does not over-train and performs similarly on the training and validation sets. After approximately 100 iterations, both the training and validation losses reach their minimum values and remain steadily low throughout the rest of the training process (up to 200 iterations).

The prediction-actual value relationship analysis is shown in Fig. 10(a). The models' prediction capabilities were examined in the $[-1.0, 1.0]$ value range. GradientBoosting and RandomForest models almost perfectly fit the ideal prediction line (dashed line). While the MLP model made successful predictions, particularly in the middle value range, it showed small deviations in extreme values. The AdaBoost model (Test $R^2 = 0.973$) showed satisfactory performance, but it was observed that the deviations increased at high values. While high variance and deviations were noticeable in positive values in the SVR model, systematic deviations and a low accuracy rate were noticeable in the predictions of the ElasticNet model. Fig. 10(b) represents the residual analysis of MFO. The distribution of residual values reveals the characteristics of the prediction errors of the models in detail. The

residual values of the GradientBoosting model were concentrated in a very narrow range around zero and exhibited optimum performance with a frequency value close to 600. The RandomForest model also showed a similarly sharp error distribution. The residual distribution of the MLP model is close to a normal distribution in the range of $[-0.2, 0.2]$, while the error distribution of the AdaBoost model is wider in the range of $[-0.4, 0.2]$. The residual distribution of the SVR model is in the range of $[-0.4, 0.6]$, while the residual values of the ElasticNet model are in the range of $[-0.6, 1.0]$. Fig. 10(c) evaluates the conformity of the residuals of the models to a normal distribution. The Q-Q plot of the MLP model shows significant deviations from the theoretical normal distribution line (red line) at the upper end. The Q-Q plot of the AdaBoost model shows a stepped structure, indicating that the residuals have discrete values. The Q-Q plot of the GradientBoosting model has a very characteristic structure, consisting mostly of horizontal segments. This situation shows that the residuals are concentrated at specific values. The Q-Q plot of the RandomForest model is almost completely horizontal, indicating that the residuals mostly have the same value. The Q-Q plot of the SVR model shows a radial structure, indicating that the residuals show significant deviations from the normal distribution. The Q-Q plot of the ElasticNet model shows a stepped and S-shaped structure, indicating significant deviations from the theoretical line. Model performance metrics analysis is given in Fig. 10(d). The performance of six different machine learning models was evaluated under the MFO approach. GradientBoosting and RandomForest models showed superior performance compared to other models. When the MAE values are examined, it is seen that the error rates of these two models are at a minimum level (approximately 0.001). The ElasticNet model showed the highest error rates, with an MAE value of 0.25 and an RMSE value of 0.30. GradientBoosting (Val $R^2 = 1.000$) and RandomForest (Val $R^2 = 1.000$) models showed excellent performance in R^2 metrics, while the performance of the ElasticNet model on the test set was quite low ($R^2 = 0.637$). The MLP model showed a medium-high success (Test $R^2 = 0.987$) and consistently performed. The MFO optimization approach showed superior performance with ensemble learning methods (GradientBoosting and RandomForest). These models showed significant superiority over other approaches in terms of both prediction accuracy and model stability. The high performance of the MLP model (Test $R^2 = 0.987$) shows that deep learning approaches can work effectively with MFO. However, the performance of classical regression approaches (SVR and ElasticNet) was limited due to the complexity of the problem space. These results show that MFO optimization is effective when used with ensemble methods and can be a reliable alternative in complex optimization problems. Fig. 10(e) shows the model's training process under MFO optimization. The training loss (blue line) and validation loss (red line) show a rapid decrease starting from approximately 0.45 at the beginning and become significantly flat after approximately 40 iterations. The two curves are quite close to each other, proving that the model does not over-train and performs similarly on the training and validation sets. After approximately 60 iterations, both the training and validation losses reach their minimum values and remain steadily low for the rest of the training process (up to 100 iterations).

Prediction-Actual Value Relationship Analysis is indicated in Fig. 11(a). In the analysis performed in the $[-2.0, 1.0]$ range, GradientBoosting and RandomForest models showed the closest performance to the ideal prediction line (Test $R^2 = 0.993$). Although the overall performance of the MLP model was well explained, deviations were observed at extreme values. While high variances (Test $R^2 = 0.673$) were observed in the SVR model, serious deviations were observed in the predictions of the ElasticNet model (Test $R^2 = 0.157$). The AdaBoost model showed moderate performance (Test $R^2 = 0.937$). Fig. 11(b) is the residual analysis for the MVO metaheuristic. The residual values of the GradientBoosting model are concentrated in a very narrow range around zero (around 600 frequencies). The RandomForest model exhibits a similarly sharp error. The residual state of the MLP model is close to a normal distribution along $[-0.75, 0.25]$. The residual values of the SVR and ElasticNet models are spread over a wide range (between −2.0 and 1.5), indicating low prediction reliability. The Q-Q plots presented in Fig. 11(c) evaluate the conformity of the residuals of the models to the normal distribution. The Q-Q plot of the MLP model shows deviations from the theoretical normal distribution line (red line) at the upper end, indicating that the distribution of the residuals has a positive tail. The Q-Q plot of the AdaBoost model exhibits a stepped structure, indicating that the residuals have discrete values. The Q-Q plots of the GradientBoosting and RandomForest models exhibit a very characteristic structure consisting of horizontal segments. This structure shows that the residuals mostly have the same value, thus significantly deviating from the normal distribution. The Q-Q plot of the SVR model forms an S-shaped curve, indicating that the residuals systematically deviate from the normal distribution. The Q-Q plot of the ElasticNet model shows the largest deviations from the theoretical line, indicating that the residuals have a distribution far from the normal distribution. Model performance metrics analysis is in Fig. 11(d). GradientBoosting and RandomForest models performed better than other models (MAE < 0.01). ElasticNet model has high error rates ($MAE \approx 0.30$, $RMSE \approx 0.50$). While GradientBoosting and RandomForest models exhibit high performance ($R^2 > 0.99$) in R^2 metrics, the test performance of the ElasticNet model is quite low ($R^2 = 0.157$). The MLP model showed a medium-high level of success (Test $R^2 = 0.979$). MVO grouping, obtaining the best results with ensemble learning methods (GradientBoosting and RandomForest). The high performance of MLP shows that deep learning treatments can work in harmony with MVO. The poor performance of classical regression treatments (SVR and ElasticNet) reveals that they are inadequately monitored to manage the complexity of the problem. The convergence curve in Fig. 11(e) shows the model's training process under MVO optimization. The training loss (blue line) and validation loss (red line) show a rapid decrease starting from approximately 0.5 at the beginning and become significantly flat after approximately 75 iterations. The two curves are very close to each other, proving that the model does not overtrain and performs similarly on the training and validation sets. After approximately 100 iterations, both losses reach their minimum values and remain steadily low for the rest of the training process (up to 200 iterations).

Fig. 12(a) represents the analysis of the prediction-actual value. In the analysis performed in the $[-1.5, 1.5]$ value range, GradientBoosting (Test $R^2 = 0.993$) and RandomForest (Test $R^2 = 0.991$) models showed the closest performance to the ideal prediction line. Although the overall performance of the MLP model was good, deviations were observed at extreme values. While high variance (Test $R^2 = 0.638$) was observed in the SVR model; there were serious deviations in the estimates of the ElasticNet model (Test $R^2 = 0.346$). The AdaBoost model exhibited satisfactory performance (Test $R^2 = 0.959$). A residual analysis of this approach is shown in Fig. 12(b). The residual values of the GradientBoosting and RandomForest models are concentrated in a very narrow range around zero (frequency close to 600). The residual distribution of the MLP model is close to a normal distribution in the $[-1.0, 0.5]$ range. The residual distribution of the SVR model is in the range of $[-2.0, 2.0]$, while the residual values of the ElasticNet model are in the range of $[-1.5, 2.0]$. The Q-Q plots presented in Fig. 12(c) evaluate the conformity of the residuals of the models to the normal distribution. The Q-Q plot of the MLP model shows deviations from the theoretical normal distribution line (red line) at the extreme values. The Q-Q plot of the AdaBoost model exhibits a stepped structure and shows significant deviations, particularly at the upper end. The Q-Q plot of the GradientBoosting model shows a characteristic structure consisting of horizontal segments, indicating that the residuals are concentrated at certain values (particularly around zero). The Q-Q plot of the RandomForest model is almost completely horizontal, indicating that the residuals mostly have a single value. The Q-Q plots of the SVR and ElasticNet models exhibit S-shaped curves, indicating that the residuals have systematic deviations from the normal distribution. Significant deviations are observed at the extreme values, particularly in the SVR model. In

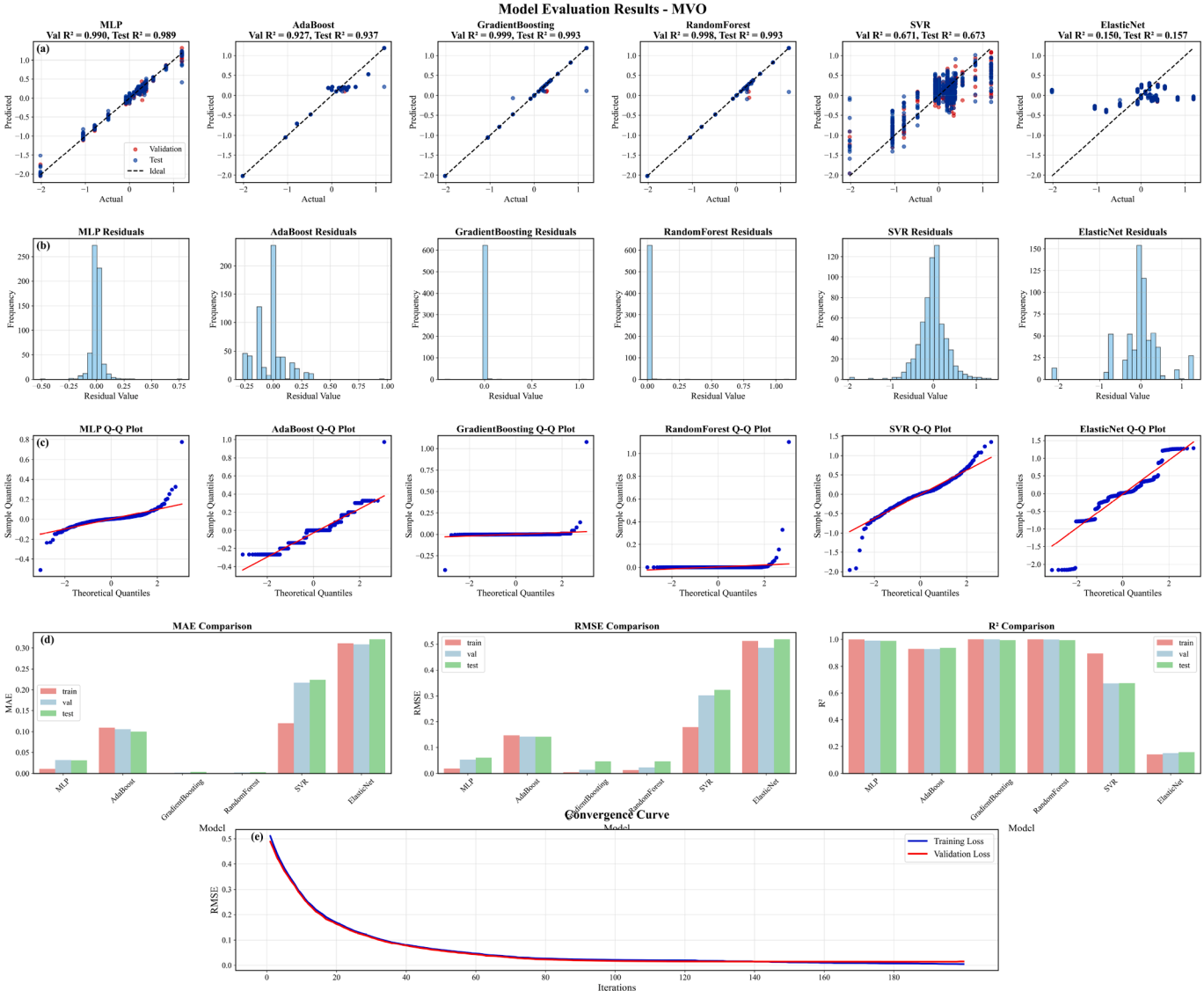

Fig. 11. (a) Model prediction comparison. (b) Residual analyses. (c) Q-Q Residual analyses for MVO. (d) The performance comparison of the models for MVO. (e) The convergence curve for MVO approach.

addition, Fig. 12(d) shows an analysis of model performance metrics. GradientBoosting and RandomForest models stand out with the lowest error rates (MAE < 0.01). The ElasticNet model has the highest error values ($MAE \approx 0.55$, $RMSE \approx 0.80$). While ensemble models show almost perfect performance in the R^2 metric ($R^2 > 0.99$), the test performance of ElasticNet is very low ($R^2 = 0.346$). The MLP model exhibited consistent performance (Test $R^2 = 0.989$). SSA optimization gave the best results with ensemble learning methods. The high performance of MLP shows that deep learning approaches can work in harmony with SSA. The low performance of classical regression approaches (SVR and ElasticNet) reveals inadequate management of space complexity. These results show that SSA provides effective results when used with ensemble methods and can be a reliable alternative in complex optimization problems. The convergence curve on the right side of Fig. 12(e) shows the model's training process under SSA optimization. The training loss (blue line) and validation loss (red line) show a rapid decrease starting from about 0.9 at the beginning and slow down after about 50 iterations. After about 100 iterations, the training loss approaches zero, while the validation loss flattens out at about 0.08. This divergence reveals that the model shows a certain degree of overfitting tendency under SSA optimization. The divergence becomes stable after 100 iterations, and the difference between training and validation losses remains constant.

Finally, the analysis of prediction-actual values is in Fig. 13(a). In the analysis performed in the $[-2.0, 1.0]$ value range, GradientBoosting (Test $R^2 = 0.996$) and Random Forest (Test $R^2 = 0.989$) models showed the closest performance to the ideal prediction line. While the MLP model made successful predictions, remarkably in the middle value range, it showed deviations in the extreme values. While high variance (Test $R^2 = 0.865$) was observed in the SVR model, there were serious deviations in the predictions of the ElasticNet model (Test $R^2 = 0.168$). The AdaBoost model showed satisfactory performance ($R^2 = 0.960$). Also, residual analysis is offered in Fig. 13(b). The residual values of the GradientBoosting model were concentrated in a very narrow range around zero (frequency close to 600). The RandomForest model showed a similarly sharp error distribution. The residual distribution of the MLP model is close to a normal distribution in the range of $[-0.2, 0.2]$. The residual distribution of the SVR model is in the range of $[-1.5, 1.0]$, while the residual values of the ElasticNet model are in the range of $[-2.0, 1.0]$. The Q-Q plots presented in Fig. 13(c) evaluate the conformity of the residuals of the models to the normal distribution. The Q-Q plot of the MLP model shows significant deviations from the theoretical normal

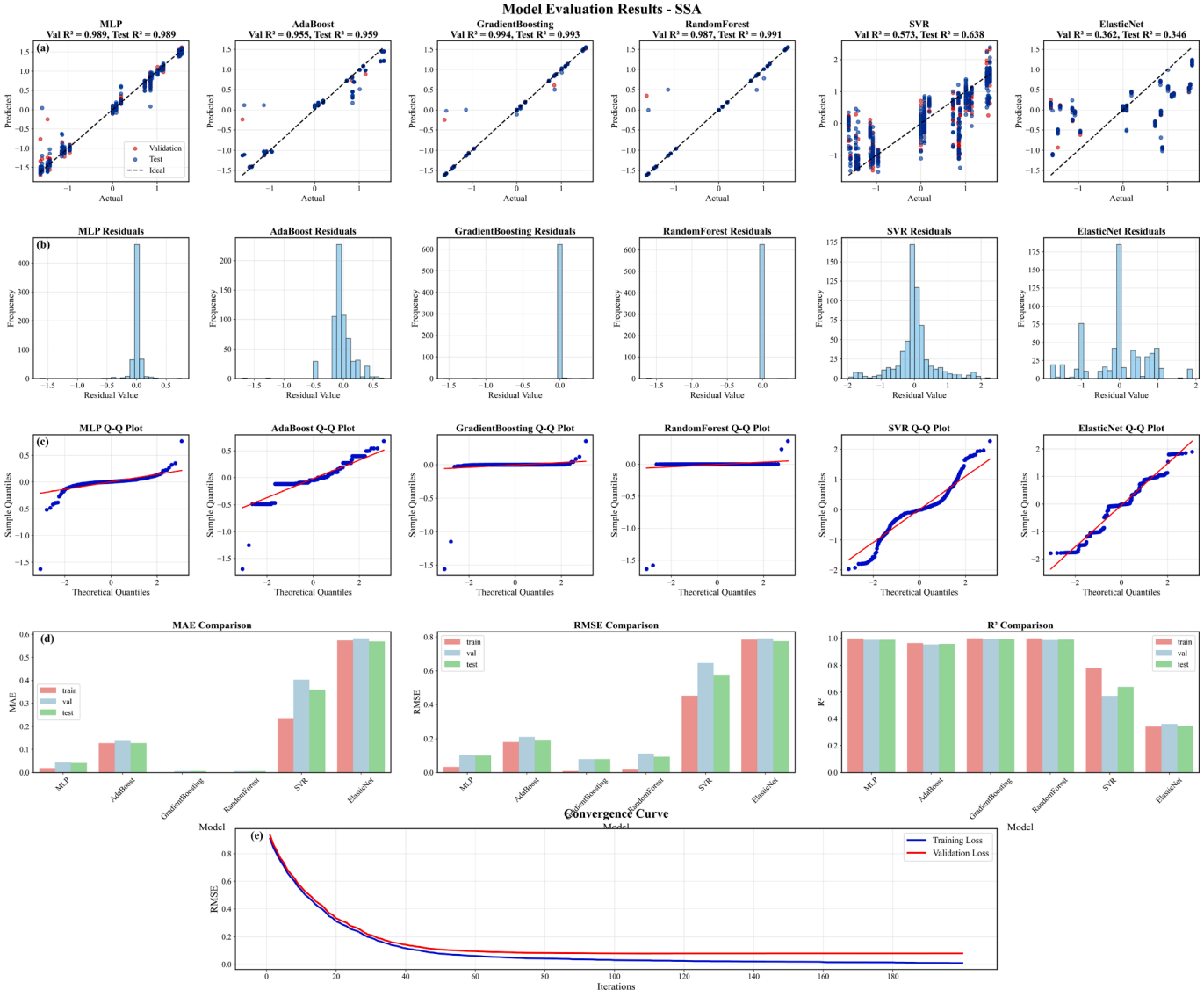

Fig. 12. (a) Model prediction comparison. (b) Residual analyses. (c) SSA's Q-Q Residual analyses (d) The performance comparison of the models for SSA. (e) SSA's convergence curve.

distribution line (red line) at the upper end. This shows that the distribution of the residuals exhibits positive skewness. The Q-Q plot of the AdaBoost model exhibits a stepped structure and shows significant deviations from normality. The Q-Q plot of the GradientBoosting model exhibits a very characteristic structure with horizontal segments, indicating that the residuals are concentrated at certain values (particularly around zero). The Q-Q plot of the RandomForest model is almost completely horizontal, confirming that the residuals mostly have a single value. The Q-Q plot of the SVR model exhibits an S-shaped curve, indicating that the residuals exhibit systematic deviations from the normal distribution. The Q-Q plot of the ElasticNet model shows the largest deviations from the theoretical line, exhibiting a stepped and irregular structure. Fig. 13(d) is the analysis of the model performance metric. GradientBoosting (MAE < 0.01) and RandomForest (MAE < 0.01) models show the lowest error rates, while ElasticNet has the highest error values ($MAE \approx 0.30$, $RMSE \approx 0.50$). GradientBoosting (Val $R^2 = 1.000$) and RandomForest (Test $R^2 = 0.989$) models showed superior performance in the four metrics. The test performance of ElasticNet is very low ($R^2 = 0.168$). The MLP model showed a consistent performance (Test 4). WOA optimization yielded the best results with ensemble learning methods (GradientBoosting and RandomForest). The high performance of MLP shows that deep learning approaches can work in harmony with WOA. The low performance of classical regression approaches (SVR and ElasticNet) shows that they are inadequate in handling the space complexity of the problem. These results show that WOA provides effective results, particularly when used with ensemble methods, and can be a reliable alternative in complex optimization problems. Fig. 13(e) shows the model's training process under WOA optimization. The training loss (blue line) and validation loss (red line) show a rapid decrease starting from approximately 0.5 at the beginning and become significantly flat after approximately 75 iterations. The two curves are very close to each other, proving that the model does not over-train and performs similarly on the training and validation sets. After approximately 100 iterations, both the training and validation losses reach their minimum values and remain steadily low throughout the rest of the training process (up to 200 iterations). In order to compare model performances not only visually but also statistically, correlation coefficients between the actual values and the predicted values were calculated for each model and metaheuristic combination. In this paper, the Pearson correlation coefficient, which evaluates the linear relationship, the Spearman cor-

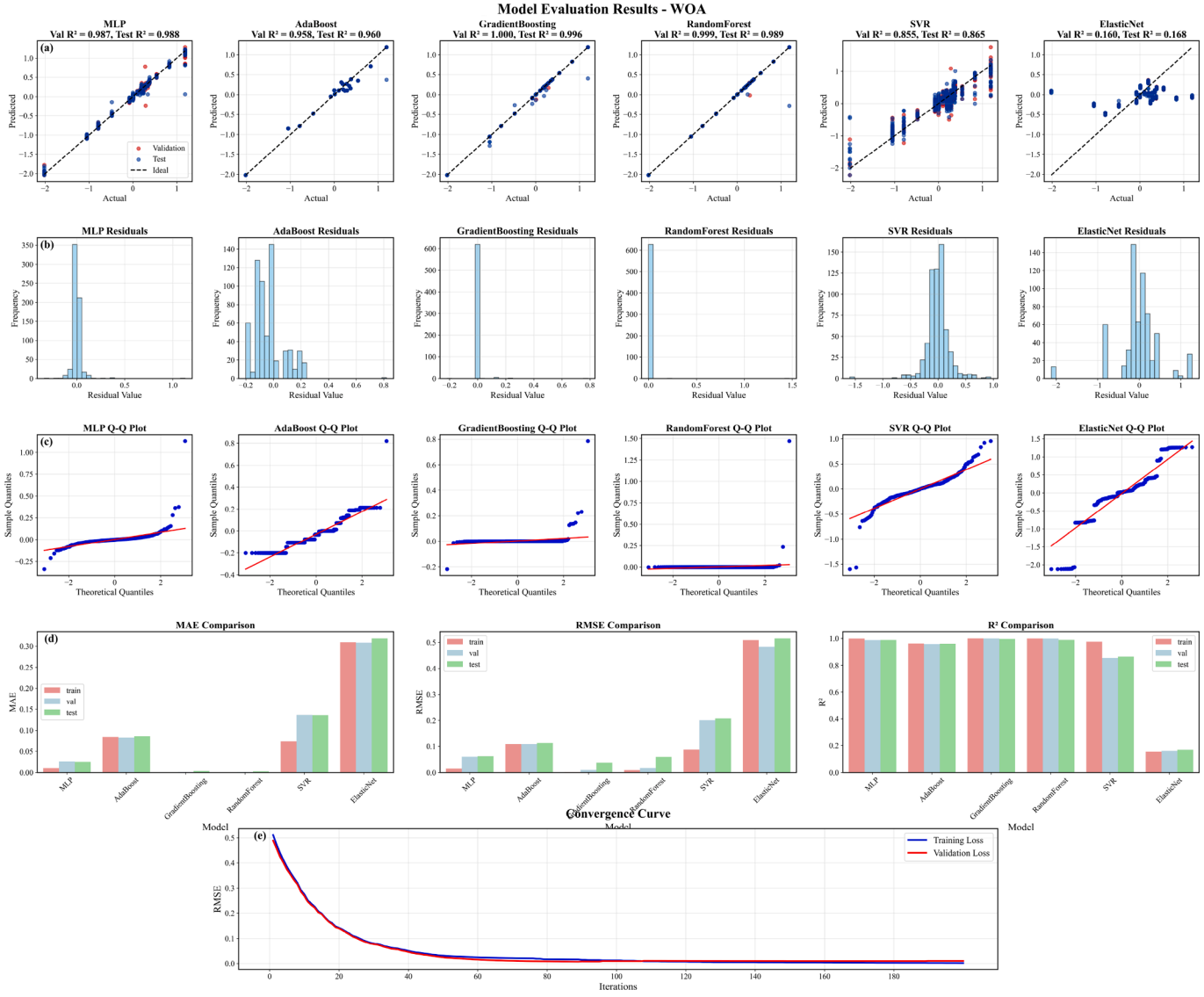

Fig. 13. (a) Model prediction comparison. (b) Residual analyses. (c) WOA's Q-Q Residual analyses (d) The performance comparison of the models for WOA. (e) WOA's convergence curve.

relation, which measures the sequential relationship, and the Kendall Tau correlation coefficient, which provides more robust results based on ranking, were used. According to the results obtained, Gradient Boosting and Random Forest models attracted attention with high correlation values under almost all metaheuristic algorithms (for example, Pearson = 0.999988, Spearman = 0.999981, and Kendall Tau = 0.999563 for Random Forest under NBRO-AGP). This shows that these models exhibit strong performance not only in terms of accuracy but also in terms of statistical consistency. On the other hand, the ElasticNet model was insufficient with low correlation values under many metaheuristics in complex nonlinear patterns (for example, Pearson = 0.418 under WOA). This finding provides a statistical justification for the low visual accuracy of ElasticNet. The corresponding correlation values are presented in Table 2 and numerically support the observations in all figures.

We see the performance comparison of metaheuristic algorithms on machine learning models in Fig. 14. In Fig. 14(a), the distributions of test performance metrics (R^2, RMSE, and MAE) are shown as box plots. In terms of R^2 scores, most metaheuristic algorithms performed above 0.9. NBRO-AGP and ALO exhibited particularly consistent and high R^2 values. A wider variability was observed in the SSA algorithm. In RMSE and MAE values, NBRO-AGP had the lowest error rates. Among the metaheuristic algorithms, SSA showed relatively higher error values. GWO exhibited a moderate performance. Fig. 14(b) shows the R^2 scores at the intersection of different metaheuristic algorithms and machine learning models with a heat map. GradientBoosting and RandomForest models achieved R^2 scores above 0.99 with all metaheuristic algorithms. The ElasticNet model performed poorly overall, with R^2 values below 0.2, particularly with MVO and SSA. The NBRO-AGP algorithm showed consistently high performance across all models. ALO produced similarly strong results. The GWO algorithm achieved R^2 scores above 0.95 in all models except ElasticNet. Our comprehensive analysis of protein solubility prediction has yielded remarkable results in machine-learning models using NBRO-AGP features. Random Forest and GradientBoosting models showed exceptional performance with a Test R^2 value of 0.999. These findings have the potential to radically transform the development of protein-based drugs in the biopharmaceutical industry. High-accuracy prediction of protein solubility allows optimization at many stages from the formulation of therapeutic proteins to manufacturing processes. In particular, these predictions play a critical role in preventing the formation of protein aggregates and increasing bioavailability.

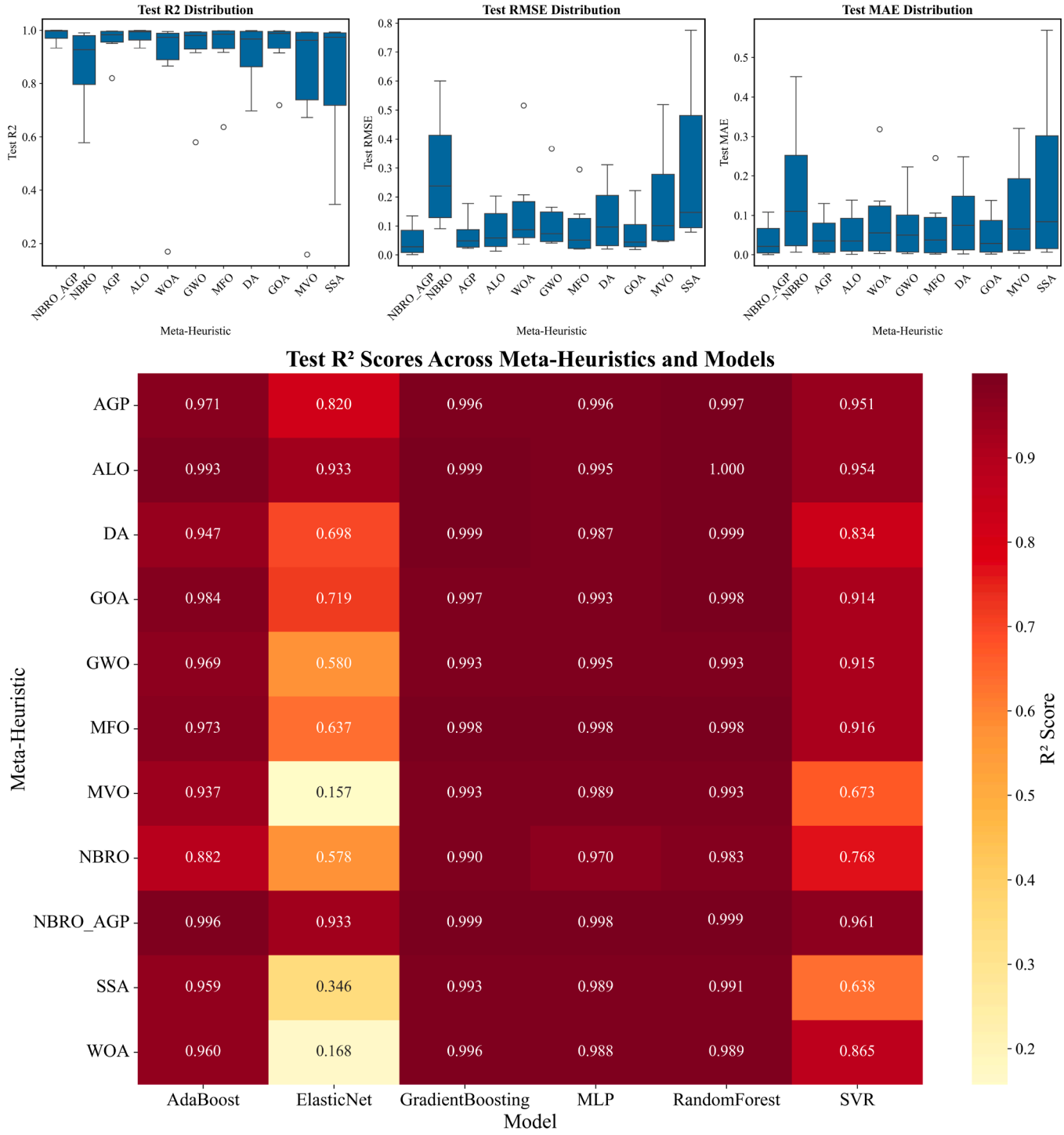

Fig. 14. (a) Test Metric Distributions, (b) Model-Algorithm Performance Matrix.

The high accuracy of our models allows for predicting potential failures at early stages of drug development, contributing to more efficient use of research and development resources and reducing overall costs. It is widely stated in the literature that the properties such as hydrophobicity, net charge, isoelectric point, aliphatic index, and surface accessibility, which are frequently selected by NRBO-AGP, are directly related to protein solubility. For example, while high hydrophobicity increases the tendency of the protein to aggregate and decreases the solubility, increasing surface accessibility stands out as a factor supporting solubility. It is known that these properties are determinant not only in protein solubility but also in many biotechnological applications such as recombinant protein production, antibody engineering, and biopharmaceutical formulation. In this respect, the properties obtained by the NRBO-AGP method are statistically and biologically significant and provide a strong basis for future interdisciplinary applications. It also provides a powerful tool for designing amino acid changes that increase solubility in protein engineering studies. The fact that the features selected with the NBRO-AGP method produce such successful results demon-

Table 2
Correlation coefficients between model predictions and actual values (Pearson, Spearman, Kendall Tau).

Metaheuristic	Model	Pearson	Spearman	Kendall Tau
NBRO_AGP	AdaBoost	0.998303	0.987990	0.963152
	ElasticNet	0.967947	0.929723	0.800877
	GradientBoosting	0.999999	0.992735	0.954862
	MLP	0.999027	0.988412	0.935237
	RandomForest	0.999988	0.999981	0.999563
	SVR	0.982522	0.960146	0.861872
NBRO	AdaBoost	0.945878	0.955462	0.888327
	ElasticNet	0.765027	0.654718	0.461049
	GradientBoosting	0.999391	0.992114	0.949788
	MLP	0.985855	0.974065	0.895431
	RandomForest	0.998530	0.996911	0.993200
	SVR	0.882879	0.859950	0.711308
AGP	AdaBoost	0.986542	0.992458	0.964238
	ElasticNet	0.942713	0.872213	0.721369
	GradientBoosting	0.998222	0.994353	0.957177
	MLP	0.987724	0.976116	0.898054
	RandomForest	0.993777	0.999709	0.998003
	SVR	0.978231	0.956830	0.853470
ALO	AdaBoost	0.996559	0.982114	0.932482
	ElasticNet	0.970845	0.927506	0.791865
	GradientBoosting	0.999669	0.991664	0.949924
	MLP	0.998512	0.985959	0.929794
	RandomForest	0.999873	0.999988	0.999779
	SVR	0.979805	0.957635	0.850070
DA	AdaBoost	0.976700	0.877000	0.774800
	ElasticNet	0.846300	0.497100	0.402100
	GradientBoosting	0.999300	0.996800	0.988200
	MLP	0.957300	0.840500	0.701100
	RandomForest	0.999400	0.999200	0.996600
	SVR	0.917500	0.708400	0.560900
GWO	AdaBoost	0.985974	0.879771	0.769910
	ElasticNet	0.782648	0.591697	0.503966
	GradientBoosting	0.996662	0.993088	0.970633
	MLP	0.991184	0.954466	0.856240
	RandomForest	0.996458	0.997209	0.993983
	SVR	0.960963	0.891086	0.764830
MFO	AdaBoost	0.986632	0.981457	0.929575
	ElasticNet	0.811422	0.711784	0.558258
	GradientBoosting	0.999191	0.996772	0.985423
	MLP	0.991661	0.972416	0.889565
	RandomForest	0.998998	0.997962	0.995940
	SVR	0.961442	0.935409	0.814831
WOA	AdaBoost	0.982818	0.836589	0.704419
	ElasticNet	0.418802	0.441545	0.333612
	GradientBoosting	0.997866	0.991143	0.959606
	MLP	0.992088	0.964743	0.886124
	RandomForest	0.994488	0.992273	0.990448
	SVR	0.933038	0.832773	0.688672
GOA	AdaBoost	0.992502	0.987290	0.955051
	ElasticNet	0.862391	0.760841	0.623845
	GradientBoosting	0.998313	0.992360	0.973621
	MLP	0.990534	0.967153	0.886029
	RandomForest	0.999070	0.998730	0.996059
	SVR	0.961703	0.922348	0.784832
MVO	AdaBoost	0.969091	0.748847	0.633105
	ElasticNet	0.404669	0.372518	0.280063
	GradientBoosting	0.996612	0.990915	0.958677
	MLP	0.988211	0.953214	0.854097
	RandomForest	0.996626	0.992292	0.985404
	SVR	0.821646	0.591715	0.452282
SSA	AdaBoost	0.983486	0.985413	0.931633
	ElasticNet	0.608959	0.594531	0.465587
	GradientBoosting	0.996629	0.990883	0.947452
	MLP	0.994901	0.982922	0.921859
	RandomForest	0.995365	0.995875	0.992719
	SVR	0.799956	0.780021	0.625279

strates that the physicochemical properties affecting protein solubility are effectively captured. In order to further strengthen the validity of the proposed NRBO-AGP method, some robust FS approaches that have been widely tested in the biomedical field are also referred to. In particular, ReliefF, Minimum Redundancy Maximum Relevance (mRMR), and Boruta have been widely applied to high-dimensional datasets such as gene expression profiles and consistently provide high predictive performance. For example, Gulande and Awale (2025) achieved over 92 % accuracy on microarray data with a hybrid FS approach combining mRMR and RSA methods. Similarly, Phan et al. (2025) proposed the BOLIMES method, which integrates Boruta and LIME algorithms for gene expression classification. Furthermore, Hamidi et al. (2023) successfully identified significant miRNA biomarkers using the Boruta method in ovarian cancer diagnosis. These studies highlight the importance of robust FS strategies in biomedical applications and support the potential applicability of the proposed NRBO-AGP method in these areas (Table A.1).

5. Conclusion

The results of this study show that the Newton-Raphson-based optimization algorithm is an effective method for continuous optimization problems. The results revealed that NRBO-AGP performed better than other metaheuristic algorithms in all regression models. The best results were obtained with Gradient Boosting, reaching MAE: 0.0001 ± 0.0000, RMSE: 0.0008 ± 0.0000, and R^2: 0.9908 ± 0.0005 values. Similar high performance (MAE: 0.0002 ± 0.0000, RMSE: 0.0025 ± 0.0000, and R^2: 0.9908 ± 0.0005) was observed with Random Forest Regressor. The multiple comparison Friedman test and subsequent Nemenyi post-hoc analysis confirm that NRBO-AGP is significantly more effective ($p < .05$), in terms of RMSE and MAE error values, and reaches the best ranking compared to competing algorithms in the R^2 accuracy metric. These findings show that NRBO-AGP is an effective feature selection tool in predicting protein solubility. The high performance of the proposed method indicates that it can be a useful tool in the field of bioinformatics and particularly in the analysis of protein properties. In future studies, the application of this method for other biological datasets and future work may explore combining it with alternative machine learning models to assess the generalizability and applicability across a broader perspective.

CRediT authorship contribution statement

Zahra Elmi: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft; **Soheila Elmi:** Methodology, Software, Validation, Visualization, Writing - review & editing; **Sebelan Danishvar:** Writing - review & editing.

Data availability

Data will be made available on request.

Declaration of interest

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

Supplementary material

Supplementary material associated with this article can be found in the online version at 10.1016/j.eswa.2025.129194

Appendix A. Hyperparameter settings

Table A.1
Hyperparameters, their descriptions, and values for the proposed method.

Hyperparameter	Description	Value / Range
α_0	Initial L1 regularization coefficient	0.001
β_0	Initial L2 regularization coefficient	0.001
α	Perturbation power coefficient for AGP (dynamically adjusted)	0.01–0.1 (adaptive)
λ	Adaptation speed	0.1
ρ	Target gradient rate	0.9
γ	Diversity weight	0.5
η	Learning rate	0.01
T	Termination condition	100-500
N	Population size	30-100
θ	Threshold	0.5 or quantile-based
β_1, β_2	Momentum parameters	0.9 / 0.999
ε	Perturbation noise magnitude in AGP	10^{-8}
K	K-fold cross validation	5
Δx	Range size for NRSR	Randomly selected (adaptive)
rand_1, rand_n	Random number generators are from the distribution (0,1) or N(0,1)	U(0,1) / N(0,1)

References

Abbasi Mesrabadi, H., Faez, K., & Pirgazi, J. (2023). Drug–target interaction prediction based on protein features, using wrapper feature selection. *Scientific Reports*, *13*(1), 3594.

Abualigah, L., Shehab, M., Alshinwan, M., & Alabool, H. (2020). Salp swarm algorithm: A comprehensive survey. *Neural Computing and Applications*, *32*(15), 11195–11215.

Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019). *IEEE Access*, *9*, 26766–26791.

Akinola, O. O., Ezugwu, A. E., Agushaka, J. O., Zitar, R. A., & Abualigah, L. (2022). Multiclass feature selection with metaheuristic optimization algorithms: A review. *Neural Computing and Applications*, *34*(22), 19751–19790.

Alhenawi, E., Al-Sayyed, R., Hudaib, A., & Mirjalili, S. (2022). Feature selection methods on gene expression microarray data for cancer classification: A systematic review. *Computers in Biology and Medicine*, *140*, 105051.

Ali, A. M., & Abdelhafeez, A. (2022). DeepHAR-net: A novel machine intelligence approach for human activity recognition from inertial sensors. *Sustainable Machine Intelligence Journal*, *1*, 1.

Amrein, M., & Wihler, T. P. (2014). An adaptive newton-method based on a dynamical systems approach. *Communications in Nonlinear Science and Numerical Simulation*, *19*(9), 2958–2973.

Barrera-García, J., Cisternas-Caneo, F., Crawford, B., Gómez Sánchez, M., & Soto, R. (2023). Feature selection problem and metaheuristics: A systematic literature review about its formulation, evaluation and applications. *Biomimetics*, *9*(1), 9.

Boothroyd, S., Kerridge, A., Broo, A., Buttar, D., & Anwar, J. (2018). Solubility prediction from first principles: A density of states approach. *Physical Chemistry Chemical Physics*, *20*(32), 20981–20987.

Chen, L., Wu, R., Zhou, F., Zhang, H., & Liu, J. K. (2023). HybridGCN for protein solubility prediction with adaptive weighting of multiple features. *Journal of Cheminformatics*, *15*(1), 118.

Choi, S., Kim, E., & Oh, S. (2013). Human behavior prediction for smart homes using deep learning. In *2013 IEEE RO-MAN* (pp. 173–179). IEEE.

Darmawahyuni, A., Nurmaini, S., Tutuko, B., Rachmatullah, M. N., Firdaus, F., Sapitri, A. I., Islami, A., Marcelino, J., Isdwanta, R., & Karim, M. I. (2024). Health-related data analysis using metaheuristic optimization and machine learning. *IEEE Access*, *12*, 55342–55356.

Davis, G. D., Elisee, C., Newham, D. M., & Harrison, R. G. (1999). New fusion protein systems designed to give soluble expression in escherichia coli. *Biotechnology and Bioengineering*, *65*(4), 382–388.

De Santis, E., Martino, A., Rizzi, A., & Mascioli, F. M. F. (2018). Dissimilarity space representations and automatic feature selection for protein function prediction. In *2018 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Ghaderzadeh, M., Shalchian, A., Irajian, G., Sadeghsalehi, H., Sabet, B. et al. (2024). Artificial intelligence in drug discovery and development against antimicrobial resistance: A narrative review. *Iranian Journal of Medical Microbiology*, *18*(3), 135–147.

Gulande, P., & Awale, R. (2025). A hybrid mRMR-RSA feature selection approach for lung cancer diagnosis using gene expression data. *Biomedical and Pharmacology Journal*, *18*(March Spl Edition), 257–270.

Habibi, N., Mohd Hashim, S. Z., Norouzi, A., & Samian, M. R. (2014). A review of machine learning methods to predict the solubility of overexpressed recombinant proteins in escherichia coli. *BMC Bioinformatics*, *15*, 1–16.

Hamidi, F., Gilani, N., Arabi Belaghi, R., Yaghoobi, H., Babaei, E., Sarbakhsh, P., & Malakouti, J. (2023). Identifying potential circulating miRNA biomarkers for the diagnosis and prediction of ovarian cancer using machine-learning approach: Application of boruta. *Frontiers in Digital Health*, *5*, 1187578.

Hammerla, N. Y., Halloran, S., & Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*.

Idicula-Thomas, S., & Balaji, P. V. (2005). Understanding the relationship between the primary structure of proteins and its propensity to be soluble on overexpression in escherichia coli. *Protein Science*, *14*(3), 582–592.

Kwon, H., Du, Z., & Li, Y. (2024). Alphafold 2-based stacking model for protein solubility prediction and its transferability on seed storage proteins. *International Journal of Biological Macromolecules*, *278*, 134601.

Li, B., & Ming, D. (2024). Gatsol, an enhanced predictor of protein solubility through the synergy of 3d structure graph and large language modeling. *BMC Bioinformatics*, *25*(1), 204.

Madgwick, S. O. H., Harrison, A. J. L., & Vaidyanathan, R. (2011). Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE international conference on rehabilitation robotics* (pp. 1–7). IEEE.

Manzoor, U., Halim, Z. et al. (2023). Protein encoder: An autoencoder-based ensemble feature selection scheme to predict protein secondary structure. *Expert Systems with Applications*, *213*, 119081.

Minervini, P., Franceschi, L., & Niepert, M. (2023). Adaptive perturbation-based gradient estimation for discrete latent variable models. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 9200–9208). (*vol. 37*).

Mirjalili, S. (2015a). The ant lion optimizer. *Advances in Engineering Software*, *83*, 80–98.

Mirjalili, S. (2015b). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, *89*, 228–249.

Mirjalili, S. (2016). Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, *27*, 1053–1073.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications*, *27*, 495–513.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61.

Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018). Grasshopper optimization algorithm for multi-objective optimization problems. *Applied Intelligence*, *48*, 805–820.

Moré, J. J. (2006). The levenberg-marquardt algorithm: Implementation and theory. In *Numerical analysis: Proceedings of the biennial conference held at Dundee, june 28–july 1, 1977* (pp. 105–116). Springer.

Nemati, S., Basiri, M. E., Ghasem-Aghaee, N., & Aghdam, M. H. (2009). A novel ACO–GA hybrid algorithm for feature selection in protein function prediction. *Expert Systems with Applications*, *36*(10), 12086–12094.

Niwa, T., Ying, B.-W., Saito, K., Jin, W., Takada, S., Ueda, T., & Taguchi, H. (2009). Bimodal protein solubility distribution revealed by an aggregation analysis of the entire ensemble of escherichia coli proteins. *Proceedings of the National Academy of Sciences*, *106*(11), 4201–4206.

Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, *16*(1), 115.

Osama, S., Shaban, H., & Ali, A. A. (2023). Gene reduction and machine learning algorithms for cancer classification based on microarray gene expression data: A comprehensive review. *Expert Systems with Applications*, *213*, 118946.

Osorio, D., Rondón-Villarreal, P., & Torres, R. (2015). Peptides: A package for data mining of antimicrobial peptides. *Small*, *12*, 44–444.

Pellizza, L., Smal, C., Rodrigo, G., & Arán, M. (2018). Codon usage clusters correlation: Towards protein solubility prediction in heterologous expression systems in e. coli. *Scientific Reports*, *8*(1), 10618.

Phan, B.-C., Ma, T., Nguyen, H.-H., & Do, T.-N. (2025). Bolimes: Boruta and lime optimized feature selection for gene expression classification. *arXiv preprint arXiv:2502.13080*.

Qian, L., Wen, Y., & Han, G. (2020). Identification of cancerlectins using support vector machines with fusion of g-gap dipeptide. *Frontiers in Genetics*, *11*, 275.

Rezaee, K., Jeon, G., Khosravi, M. R., Attar, H. H., & Sabzevari, A. (2022). Deep learning-based microarray cancer classification and ensemble gene selection approach. *IET Systems Biology*, *16*(3–4), 120–131.

Sazzed, S. (2021). Anova-src-bpso: A hybrid filter and swarm optimization-based method for gene selection and cancer classification using gene expression profiles. In *Canadian AI*. Springer.

Shimizu, Y., Kanamori, T., & Ueda, T. (2005). Protein synthesis by pure translation systems. *Methods*, *36*(3), 299–304.

Singh, S., Gupta, H., Sharma, P., & Sahi, S. (2024). Advances in artificial intelligence (AI)-assisted approaches in drug screening. *Artificial Intelligence Chemistry*, *2*(1), 100039.

Sowmya, R., Premkumar, M., & Jangir, P. (2024). Newton-raphson-based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems. *Engineering Applications of Artificial Intelligence*, *128*, 107532.

Tang, H., Cao, R.-Z., Wang, W., Liu, T.-S., Wang, L.-M., & He, C.-M. (2017). A two-step discriminated method to identify thermophilic proteins. *International Journal of Biomathematics*, *10*(04), 1750050.

Tavasoli, N., Rezaee, K., Momenzadeh, M., & Sehhati, M. (2021). An ensemble soft weighted gene selection-based approach and cancer classification using modified metaheuristic learning. *Journal of Computational Design and Engineering*, *8*(4), 1172–1189.

Wang, X., Liu, Y., Du, Z., Zhu, M., Kaushik, A. C., Jiang, X., & Wei, D. (2021). Prediction of protein solubility based on sequence feature fusion and DDcCNN. *Interdisciplinary Sciences: Computational Life Sciences*, *13*(4), 703–716.

Weerakoon, S., & Fernando, T. (2000). A variant of newton's method with accelerated third-order convergence. *Applied Mathematics Letters*, *13*(8), 87–93.

Xiaohui, N., Feng, S., Xuehai, H., Jingbo, X., & Nana, L. (2014). Predicting the protein solubility by integrating chaos games representation and entropy in information theory. *Expert Systems with Applications*, *41*(4), 1672–1679.

Yao, S., Hu, S., Zhao, Y., Zhang, A., & Abdelzaher, T. (2017). Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th international conference on world wide web* (pp. 351–360).

Yugandhar, K., Gupta, S., & Yu, H. (2019). Inferring protein-protein interaction networks from mass spectrometry-based proteomic approaches: A mini-review. *Computational and Structural Biotechnology Journal*, *17*, 805–811.

Zhang, S., Zhang, T., & Liu, C. (2019). Prediction of apoptosis protein subcellular localization via heterogeneous features and hierarchical extreme learning machine. *SAR and QSAR in Environmental Research*, *30*(3), 209–228.