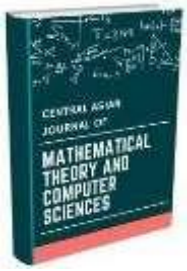

CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS
Volume: 06 Issue: 03 | July 2025 ISSN: 2660-5309

Article

Development of a Hybrid Optimization Algorithm for Efficient Neural Network Training

Mohammed Dawood Salman[1], Ghufran K. Joad[2], Ali Kadhim Yaqoob[3]

1. Institute of Applied Arts, Middle Technical University, Baghdad, Iraq
2. Department of Production and Metallurgical Engineering, University of Technology, Iraq
3. Department of Mathematics, College of Science , University Of Anbar; Ramadi, Iraq

* Correspondence: mohammed_dawood@mtu.edu.iq, ghufran.k.jwad@uotechnology.edu.iq , ali.kazem@uoanbar.edu.iq

Abstract: In this study, a hybrid optimization algorithm in mathematical combining the Hestenes-Stiefel (HS) and Polak-Ribiere (PR) methods was developed using an adaptive approach to optimize the training of Feed-Forward Neural Networks (FFNNs). This approach aims to leverage the global convergence power of the HS method and the ability of PR to avoid local minimum. The hybrid algorithm was tested on three real world datasets (Iris, Glass, and Wine), and compared to the results obtained using the original two methods (HS and PR). The hybrid algorithm showed shorter training times across all datasets and hybrid algorithm significantly reduced the mean square error (MSE) compared to the two separate methods, resulting in faster convergence with training accuracies fir the Iris datasets, Glass dataset, and Wine dataset being 98.50%, 98.39%, and 65.98%, respectively, enhancing efficiency. The algorithm also proved to be able to handle data with high variance or nonlinearity more effectively, making it suitable for training neural networks in machine learning applications.

Keywords: Hestenes-Stiefel, Polak-Ribiere, hybrid algorithm, adaptive hybrid method FFNN

Citation: Salman, M. D. Development of a Hybrid Optimization Algorithm for Efficient Neural Network Training. Central Asian Journal of Mathematical Theory and Computer Sciences 2025, 6(3), 344-360.

Received: 10th Mar 2025
Revised: 21th Mar 2025
Accepted: 04th Apr 2025
Published: 17th Apr 2025

1. Introduction

Conjugate gradient methods are frequently employeing mathematical techniques for addressing Systems of linear equations and optimizing unconstrained problems. These approaches reinforce efficiency over traditional regression techniques by utilizing gradient information from previous steps to accelerate the search for the optimal solution. From a mathematical standpoint, the primary advantage of conjugate regression methods lies in their ability to generate conjugate search directions, which significantly reduces the number of steps required to find the solution [1].

These methods are effective for solving large-scale of problems because they do not require storing entire matrices in memory, making them well-suited for computationally demanding tasks. Conjugate gradient algorithms (C.G.As) find applications across range of a field, as structural optimization in engineering, solving equation in physic, and improving the performance of neural networks particularly in real of artificial intelligence [2]. Despite their efficiency these methods face challenges related the precision and speed of convergence solution, especially in complex scenarios with nonconvex functions or saddle point. To broaden this applicability and enhance their efficiency throughout range of a field of task will find hybrid algorithms [3]. C.G.As also widely applied in fields a portfolio optimization and the resolution of immense systems of equation. Studied

indicates that these methods may produce accurate solutions to complex models while minimizing storage requirements, making them ideal for more dimensional problems especially those involving positive definite and symmetric. But their accuracy may express to danger of numerical errors, particularly when deal with poor conditional matrices which weakens their performance on no quadratic functions. This condition has prompted to development C.G.As to improve both accuracy and convergence [4]. This development preparing more accurate of C.G.As as Hager and Zhang [5], and fast convergence to optimal solution [6-8]. Many studies appeared as hybrid C.G.As in [9], motivation on nonlinear C.G.As [10] and [11], improved on some methods as [12], improving some C.G.As to deal with some nonlinear or fuzzy equations, as [13], [14], [15], and [16]. The unconstrained nonlinear optimization problems issues is addressed as follows:

$$\min\{h(y), y \in \mathbb{R}^n\} \quad (1)$$

Take h: $\mathbb{R}^n \to \mathbb{R}$ is any function with property (continuous and differentiable and bounded below). The modeling of optimization begin by estimate initial y where y in set in $\mathbb{R}$. The backpropagation algorithm (BP) update vector y by steepest descent method. This style update the Wight by depend on negative gradient function h, which minimizes function and improves model performance.

$$y_{k+1} = y_k - \xi_k d_k \quad (2)$$

$$d_k = \nabla h(y_k)$$

$$d_{k+1} = \begin{cases} -g_{k+1} & if\, k = 0 \\ -g_{k+1} + \beta_k d_k & if\ k \geq 1 \end{cases} \quad (3)$$

Where $\xi_k \in (0,1)$ is learning rate, and g_k, β_k are update parameters.

For iteration k the solution achieves global convergence if traditional Wolf criteria are used to evaluate a step size is acceptable [17-18].

$$h(y_k - \xi_k d_k) \leq h(y) + \lambda \xi_k g_k^T d_k \quad (4)$$

$$|d_k^T g(y_k - \xi_k d_k)| \geq \vartheta d_k^T g_k \quad (5)$$

$$0 < \lambda < \vartheta < 1$$

The strong Wolfe conditions by [19-20]

$$f(y_k - \xi_k d_k) \leq f(x) + \lambda \xi_k g_k^T d_k \quad (6)$$

$$|d_k^T g(y_k - \xi_k d_k)| \leq -\vartheta d_k^T g_k \quad (7)$$

Training process for Feed forward neural network (FFNN) is complex that require carful parameter optimization to get high performance. The main problem lies in challenges associated with get high convergence speed and training accuracy particularly, when dealing with nonlinear data or with fluctuating gradient but they face significant bounded as slow convergence or falling to local minima. To overcome these bounded this paper aims to develop a hybrid algorithm that takes advantage of the power of HS method and flexibility for PR method to improve the training efficiency of FFNN, making them better able to handle complex training challenge. The combining two methods enhances the ability to explore optimal solution and dynamical adept to data requirements during the training process. The approved method of hybrid is based on the equation:

$$\beta_{k+1}^{New} = \begin{cases} \beta_{k+1}^{first} & if \quad (\nabla h(y_{k+1}) - \nabla h(y_k))^T \nabla h(y_{k+1}) \geq 0 \\ \beta_{k+1}^{second} & if \quad (\nabla h(y_{k+1}) - \nabla h(y_k))^T \nabla h(y_{k+1}) < 0 \end{cases} \quad (8)$$

The HS and PR were chosen to combined in hybrid algorithm, due to their complementary features. HS method has strong global convergence that improves the ability to reach optimal solutions in general, while PR shows efficiency in handing nonlinear process and avoid falling local minima. The caused to chosen this methods since strong convergence for HS properties and handing nonlinear process and avoid falling local minima for PR method.

This study including 5 sections. section-1 including the introduction, section-2 including derivation the hybrid algorithm, in section-3 introduce information about ratio property linked to hybrid formulation, prove the absolute convergence for hybrid algorithm in section-4, finally in section-5 including the application on real data.

2. Materials and Methods

In this section, the hybrid conjugate gradient algorithm is introduced and derived by combining the PR and HS algorithms using algorithm adaptively switches. The integration of these methods is used to establish a new search direction, leveraging the principle of developing a hybrid optimization algorithm.

$$d_{k+1} = -g_{k+1} + \beta_k^{hybrid} d_k \tag{9}$$

Where β_k^{hybrid} given by:

$$\beta_k^{hybrid} = \begin{cases} \beta^{PR} \ if \quad (\nabla h(y_{k+1}) - \nabla h(y_k))^T \nabla h(y_{k+1}) \geq 0 \\ \beta^{HS}_{k+1} \ if \quad (\nabla h(y_{k+1}) - \nabla h(y_k))^T \nabla h(y_{k+1}) < 0 \end{cases} \tag{10}$$

$$\beta_k^{PR} = \frac{(\nabla h(y_{k+1}) - \nabla h(y_k))^T \nabla h(y_{k+1})}{|\nabla h(y_{k+1})|^2}$$

$$\beta_k^{HS} = \frac{(\nabla h(y_{k+1}) - \nabla h(y_k))^T \nabla h(y_{k+1})}{(\nabla h(y_{k+1}) - \nabla h(y_k))^T d_k}$$

Where β_k^{PR} is search direction of PR method [12-13], and $\beta_k^{HS}$is search direction of HS method [9].

Hybrid Algorithm Steps

Based on new search direction the steps of hybrid algorithm can be butting as follows:

1. Initialization:

- Set an initial estimate for the weights y_0 and compute the initial gradient $g_0 = \nabla h(y_0)$
- Initialize the search direction as $d_0 = -g_0$

2. Iteration Loop:

- For each iteration $k = 0,1, \ldots$

Step 1: Gradient Computation:

- Find the gradient by:$g_k = \nabla h(y_k)$.
- Evaluate $\nabla h(y_{k+1})$ aty_{k+1}, where y_{k+1} is updated weights.

Step 2: calculate the adaptive hybrid coefficient β_k^{hybrid} using equation (10).

Step 3: Update the search direction using the hybrid coefficient β_k^{hybrid} using equation (9).

Step 4: Perform a line search to find the optimal step size ξ_k that satisfies the strong Wolfe conditions. This ensures sufficient decrease and curvature conditions.

Step 5: Weight Update using equation (2).

Step 6: Convergence Check:

- Check the norm of the gradient$\|g_k\|$, if $\|g_k\| < \epsilon$, (where ϵ is a small predefined tolerance), the algorithm has converged, and the loop is terminated.
- If the stopping criterion is not met, increment k and return to **Step 1** for the next iteration.

Step 7: Update the neural network parameters

This hybrid method ensures that the algorithm adaptively chooses between the Polak-Ribiere and Hestenes-Stiefel methods based on the gradient behavior, leading to faster convergence and better handling of complex, non-linear optimization problems.

The hybrid conjugate gradient algorithm ensures that the search direction satisfies the descent property at each iteration, which is essential for convergence. The descent property can be stated as follows:

$$g_{k+1}^T d_{k+1}^{hybrid} \leq -\delta \|g_{k+1}\|^2 \tag{11}$$

When $k \geq 0$, and $\delta > 0$, and d_{k+1}^{hybrid} is the new search direction.

To prove that the hybrid method maintains this descent property, consider the update of the search direction, and by using equation (9). By multiplying both sides by g_{k+1}^{T}, we get:

$$d_{k+1} g_{k+1}^{T} = -g_{k+1} g_{k+1}^{T} + \beta_{k}^{hybrid} d_{k} g_{k+1}^{T} \quad (12)$$

For descent to hold, we need $g_{k+1}^{T} d_{k+1} \leq 0$. The adaptive hybrid coefficient β_{k}^{hybrid} is designed to ensure this condition is met. Specifically, it adjusts based on the gradient behavior:

- If $\nabla h(y_{k+1}) - \nabla h(y_k)$ aligns positively with $\nabla h(y_{k+1})$, is β_k^{PR} used, leveraging Polak-Ribiere's robustness in maintaining descent properties.
- If there is a negative alignment, β_k^{PR} is selected, ensuring global convergence by Hestenes-Stiefel's formula.

Thus, the hybrid algorithm maintains sufficient descent at each iteration.

1. The Hybrid Method and Improved Algorithm is Converging

The proposed hybrid algorithm combines the strengths of the Hestenes-Stiefel (HS) and Polak-Ribiere (PR) methods, ensuring convergence under standard assumptions used for conjugate gradient methods. The convergence of the algorithm is guaranteed by the following assumptions:

1. Lipschitz Continuity of the Gradient: The gradient of the objective function $h(y)$ is assumed to be Lipschitz continuous, meaning there exists a constant $\tau > 0$:

$$\|\nabla h(x) - \nabla h(y)\| \leq \tau \|x - y\|, \quad \forall x, y$$

2. Bounded Level Set: The level set $\{y \in \mathbb{R}^n : h(y) \leq h(y_0)\}$ is bounded, ensuring that the iterates remain within a finite region.

3. Sufficient Descent Condition: The hybrid algorithm satisfies a sufficient descent condition, ensuring that the function value decreases by at least:

$$h(y_{k+1}) \leq h(y_{k+1}) - \alpha \|g_k\|^2 \quad (12)$$

For some $\alpha > 0$.

4. The codomain (target) function h is a uniformly convex function, meaning that there exists a constant μ that achieves variance.

$$(\nabla f(x) - \nabla f(y))^{T} (x - y) \geq \mu \|x - y\|^2, x, y \in \{y \in \mathbb{R}^n : h(y) \leq h(y_0)\}$$

Alternatively, assuming, there exists an upward constant θ such that:

Alternatively, assuming, there exists an upward constant θ such that:

$$\|x\| \leq \theta, \forall x \in \{y \in \mathbb{R}^n : h(y) \leq h(y_0)\}$$
$$\underline{\epsilon} \leq \|g(x)\| \leq \epsilon, \forall x \in \{y \in \mathbb{R}^n : h(y) \leq h(y_0)\} \quad (13)$$

Lemma: We introduce assumptions and (13) that are fulfilled, and we refer to (10) for the conjugate gradient method, where β_k represents the direction of the search with a slope, and the step length ξ_k is determined using the strong line search for Wolfe conditions . Assuming that:

$$\sum_{k>1} \frac{1}{\|d_k^{hybrid}\|^2} = \infty$$

Then we get:

$$\lim(\inf \|g_k\|) = 0, as\ k \to \infty$$

Theorem: Let h be a continuously differentiable function with a Lipschitz continuous gradient and bounded level set. The hybrid conjugate gradient method, with appropriately chosen step sizes ξ_k satisfying the Wolfe conditions, generates a sequence $\{y_k\}$ such that:

$$\lim\{\nabla h(y_k)\} = 0 \quad (13)$$

This implies that the algorithm converges to a stationary point of the objective some function.

Proof: The proof follows from standard results in conjugate gradient methods, leveraging the sufficient descent condition, the boundedness of the level set, and the fact that the hybrid algorithm ensures a decreasing sequence of function values.

1. The hybrid method alternates between the Hestenes-Stiefel and Polak-Ribiere methods, both of which are known to ensure global convergence under standard assumptions.
2. By ensuring that β_k^{PR}, and β_k^{HS}always maintains sufficient descent, and using line search methods (satisfying the Wolfe conditions), the algorithm guarantees that the iterates y_k remain bounded.
3. The decreasing sequence of function values, combined with the boundedness of the gradient, ensures that the gradient norm converges to zero.

Thus, the proposed hybrid method is globally convergent, combining the rapid convergence properties of the Hestenes-Stiefel method and the robustness of the Polak-Ribiere method. This makes it highly effective for training feed-forward neural networks and other large-scale optimization problems.

2. Performance of the Hybrid Algorithm

A method called the CGA was evolved to train feedforward neural networks. In order to classify data from various classification problems, including the Iris [21], Glass [22], and Wine datasets [23], this approach was compared against the PR, HS and hybrid algorithms. The simulations were conducted using MATLAB R2022a.

Table 1: Problems in Real-data Sorting

	Sorting data set		
	Iris	Glass	Wine
Data size	150	214	178
No. of training observations	135	190	150
No. of testing observations	15	24	28
No. of Hidden	5	5	5

Table 1 presents real-world data classification problems: This table shows details about the three datasets used (Iris, Glass, and Wine), including the data size, number of training samples, number of test samples, and number of nodes in the hidden layer. The table reflects the diversity and size of the data to enhance the reliability of the hybrid algorithm performance evaluation.

Table 2: Average Performance Comparison for Sorting Problems for hybrid algorithm

Dataset	Algorithms	No. of training iteration	Average training time	Average training accuracy	Average test accuracy	Average training MSE	Average test MSE
Iris	**Hybrid**	**80**	**98.50%**	**98.67%**	**1.41**	**6.665461e-01**	**1.500000e-02**
	PR	80	98.03%	98.58%	1.13	6.665882e-01	1.972222e-02
	HS	80	98.31%	98.60%	1.14	6.666112e-01	1.694444e-02
Glass	**Hybrid**	**80**	**98.39%**	**94.04%**	**1.10**	**4.999887e-01**	**1.614583e-02**
	PR	80	96.12%	93.80%	1.06	4.999864e-01	3.880208e-02
	HS	80	96.56%	94.50%	1.07	4.999950e-01	3.437500e-02
Wine	**Hybrid**	**80**	**65.98%**	**92.89%**	**1.10**	**6.666665e-01**	**3.401786e-01**
	PR	80	65.60%	92.81%	1.04	6.666666e-01	3.440476e-01
	HS	80	64.42%	92.66%	1.11	6.665562e-01	3.558036e-01

Table 2 shows the average performance of the hybrid algorithm compared to the PR and HS methods. It includes the number of training iterations, average training time, training and testing accuracy, and mean square error (MSE) for each of the three datasets. The table shows that the hybrid algorithm outperforms the individual methods in terms of training and testing accuracy, as well as improving MSE.

Table 3: Value the training for Iris

Unit	Initial value	Stopped value	Target value
Epoch	0	20	1000
Elapsed Time	-	00:00:00	-
Performance	0.873	0.00881	0
Gradient	0.99	0.00773	1e-07
Mu	0.001	1e-06	1e+10
Validation Checks	0	6	6

Table 3 shows the improvement in performance across a number of stages, including number of iterations, performance (MSE), gradient, and μ measure during the training periods. It shows that performance reaches its best level after 20 iterations.

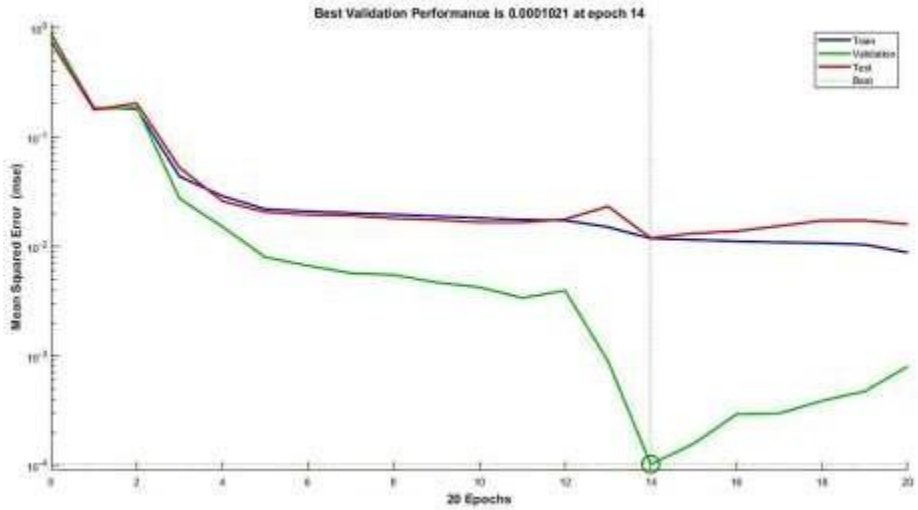

Figure 1. Optimal validation performance for Iris dataset at epoch 7

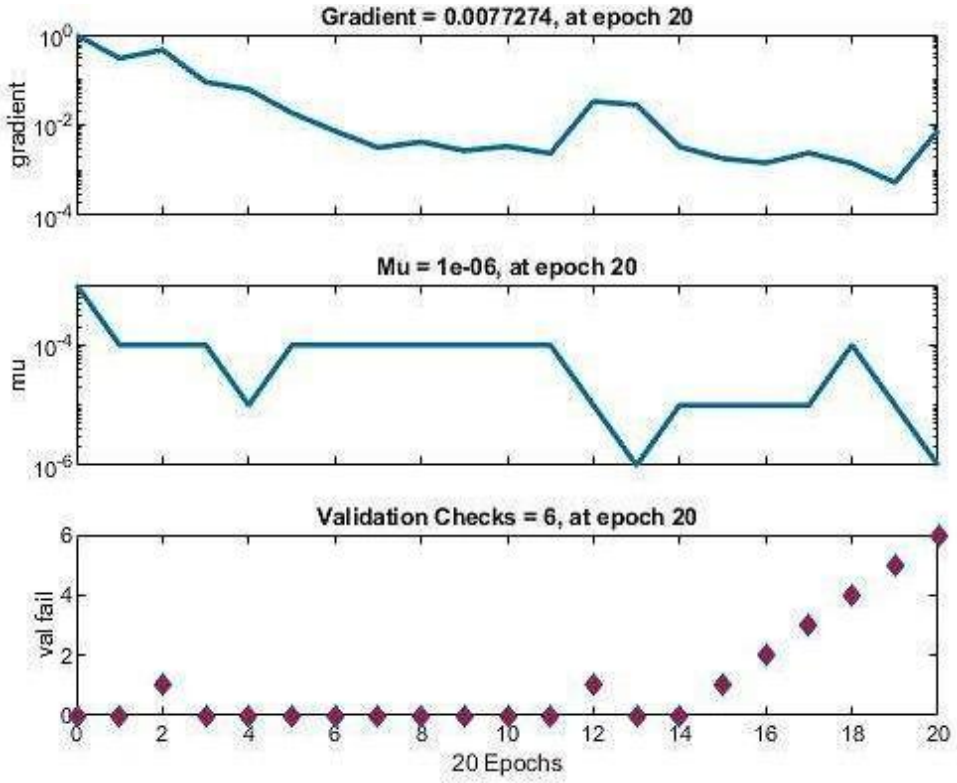

Figure 2. Evolution of training performance of the Iris dataset

Figure 1 shows that the best performance was achieved in the seventh iteration, indicating rapid convergence of the algorithm. Figure 2 shows the evolution of performance over time, reflecting the continuous progress of the training process. This is consistent with the results obtained in Table 3.

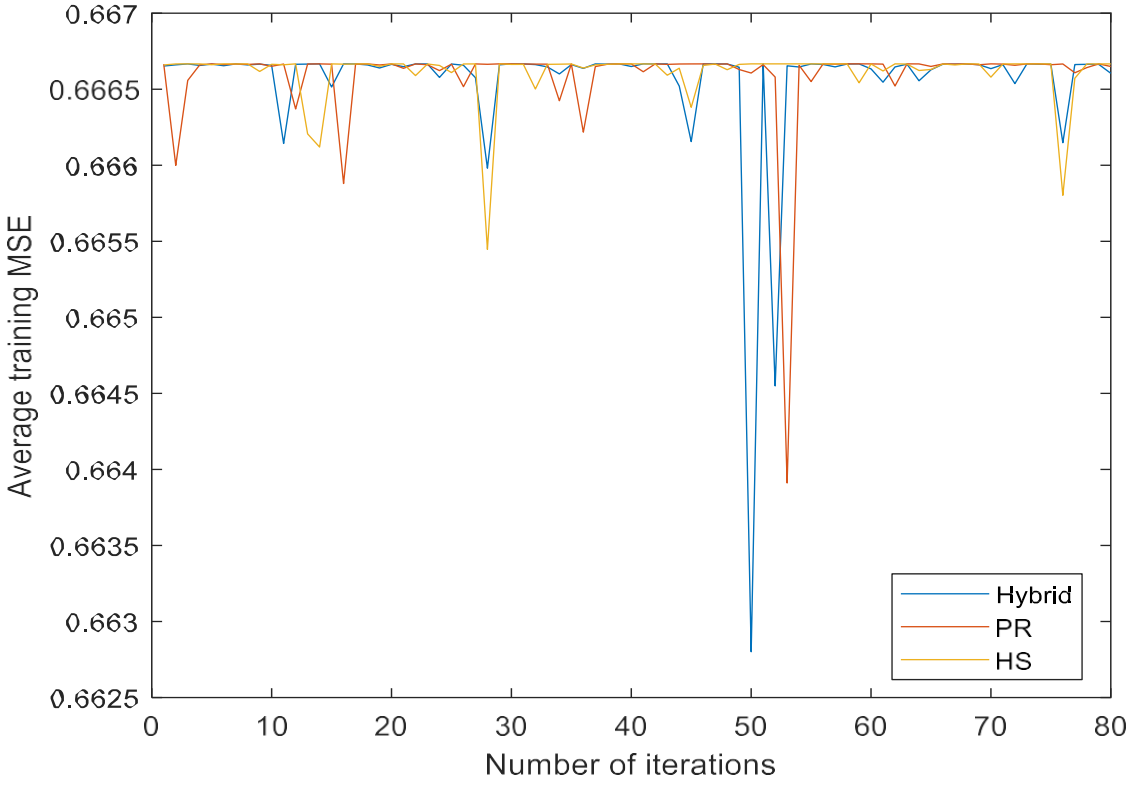

Figure 3 Mean Squared Error (MSE) during training for the Iris dataset

Figure 4. Time taken to train on Iris dataset

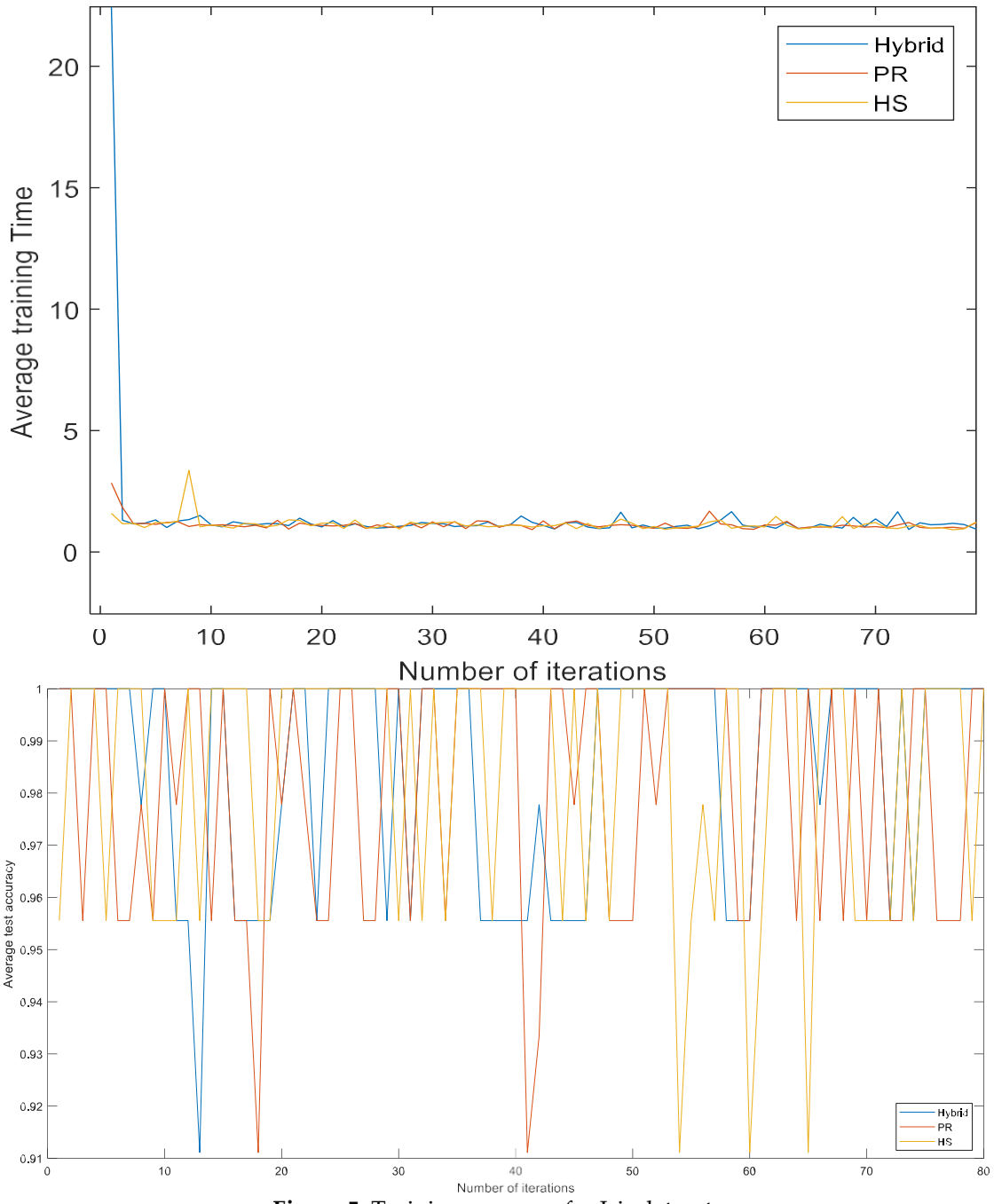

Figure 5. Training accuracy for Iris dataset

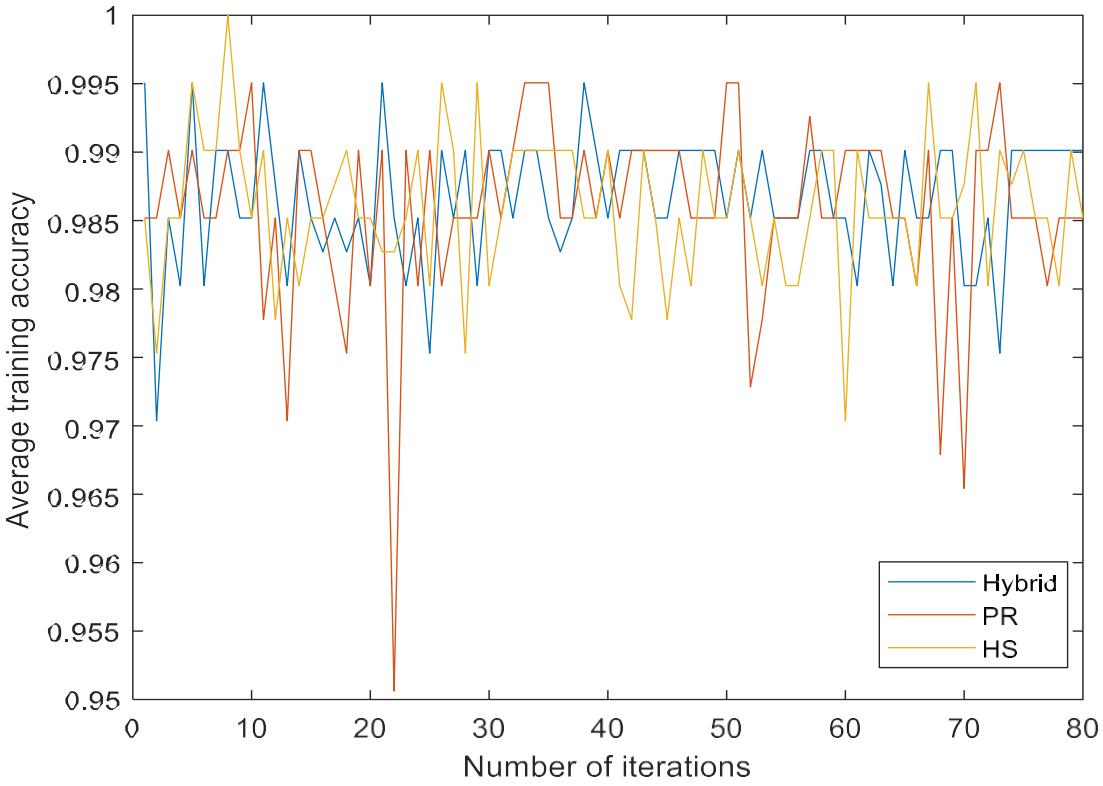

Figure 6. Final results of training errors for the Iris dataset

Figures 3-6 show the results of the first dataset. Figure 3 shows a decrease in MSE with increasing iterations, indicating that the model is gradually improving. Figure 4 shows how the average training time gradually decreases thanks to the efficiency of the hybrid algorithm. Figure 5 shows a gradual improvement in model accuracy, indicating that the algorithm is learning effectively. Figure 6 shows the shape of the training error decline with increasing iterations. From the results obtained for the first dataset, the hybrid algorithm achieved high accuracy in training (98.50%) and testing (98.67%), demonstrating its ability to handle data classification effectively. It also showed a significant decrease in MSE during training, reflecting strong performance in reducing errors.

Table 4: Value the training for Glass

Unit	Initial value	Stopped value	Target value
Epoch	0	14	1000
Elapsed Time	-	00:00:01	-
Performance	0.722	0.0275	0
Gradient	1.07	0.00328	1e-07
Mu	0.001	0.0001	1e+10
Validation Checks	0	6	6

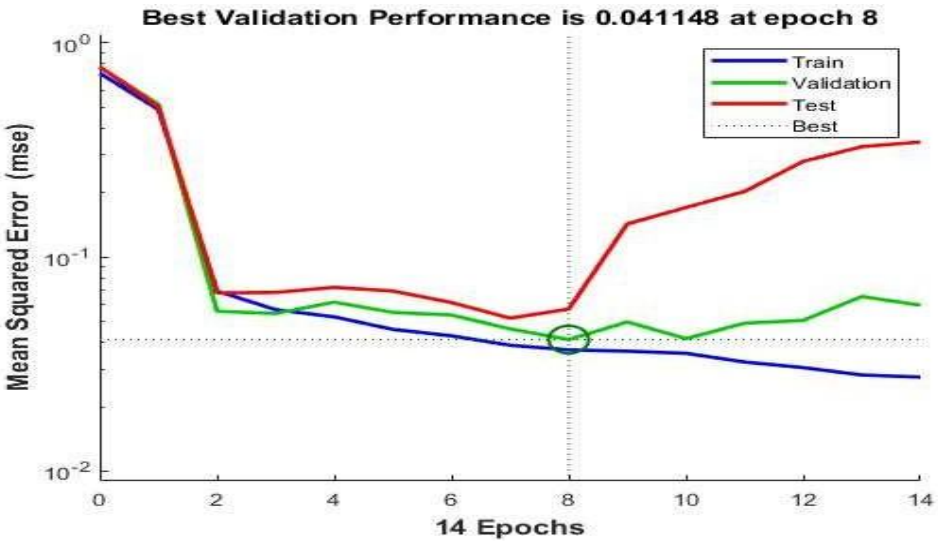

Figure 7. Optimal validation performance for Glass dataset at epoch 14

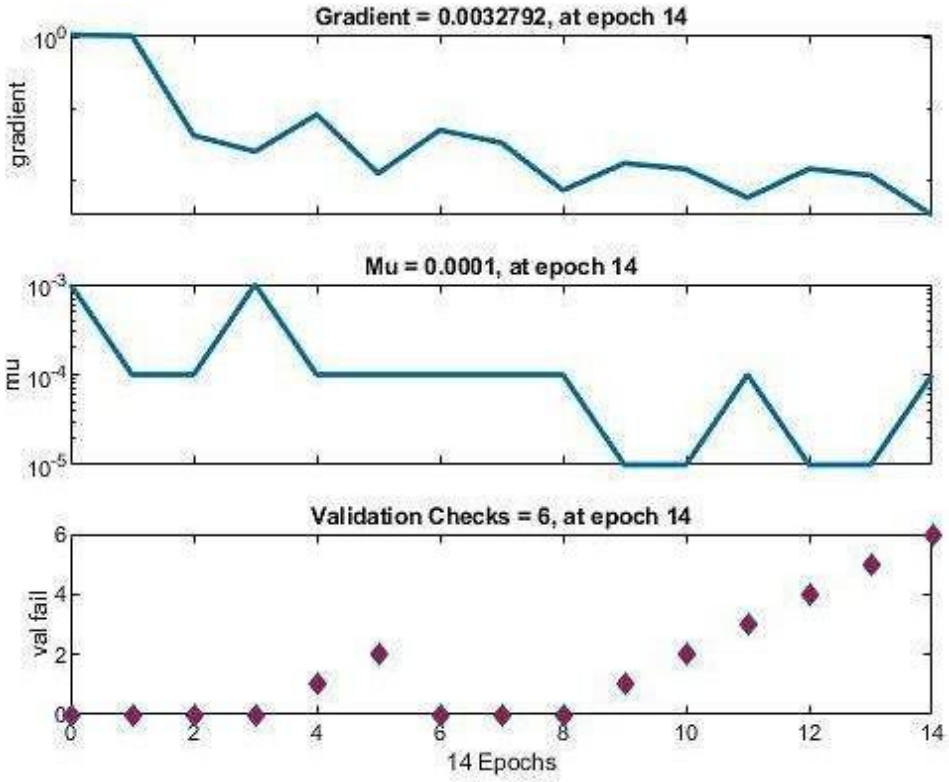

Figure 8. Training state for Glass dataset

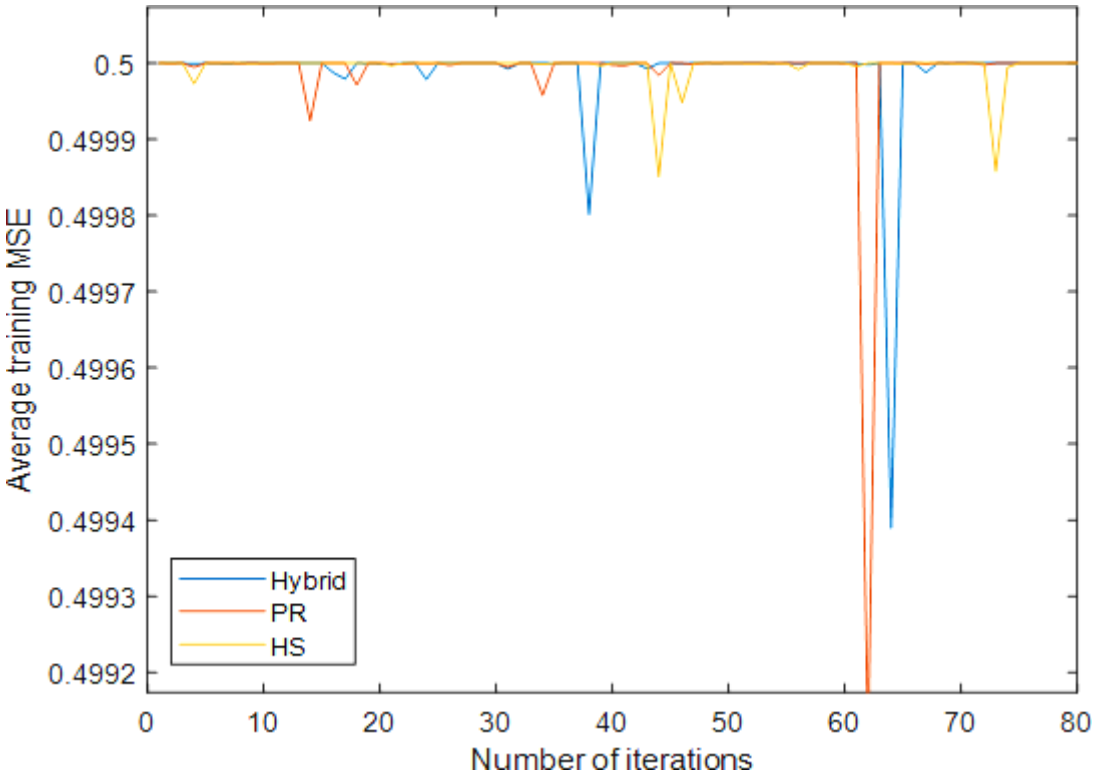

Figure 9. Mean squared error (MSE) during training for the Glass dataset

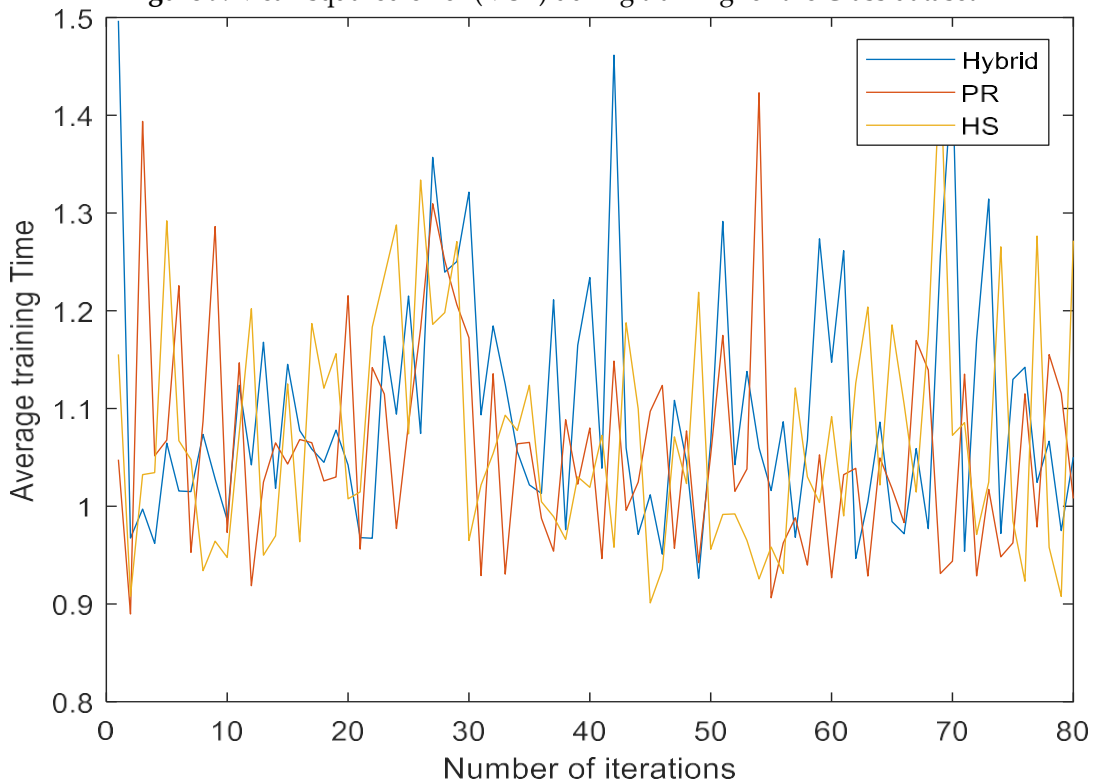

Figure 10. Time taken to train on Glass dataset

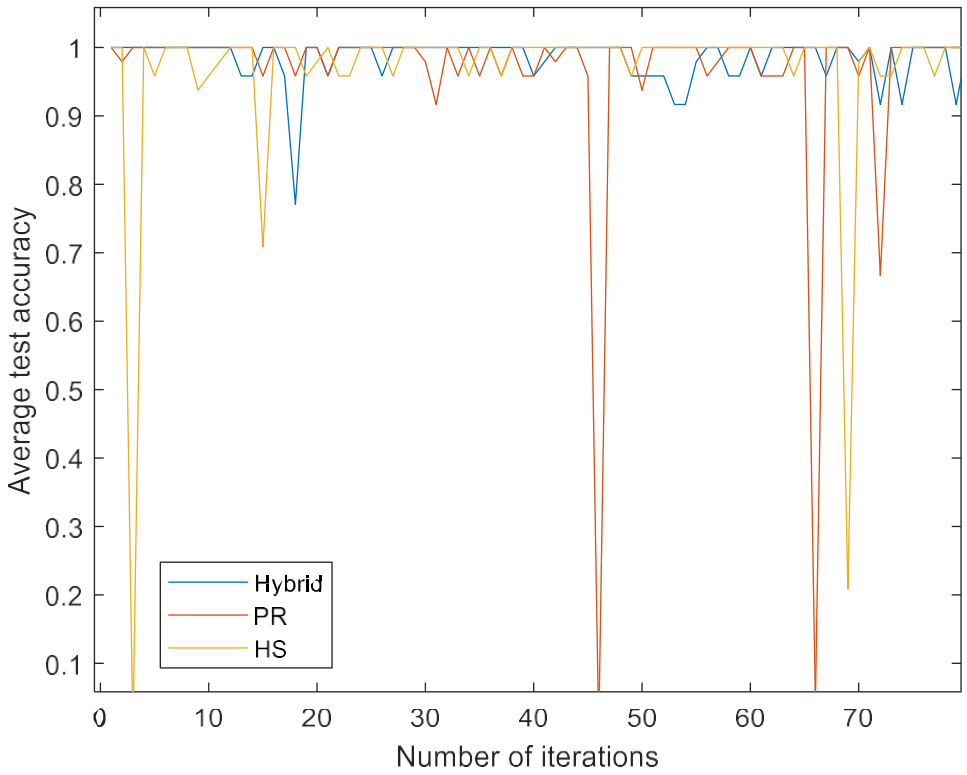

Figure 11. Training accuracy for Glass dataset

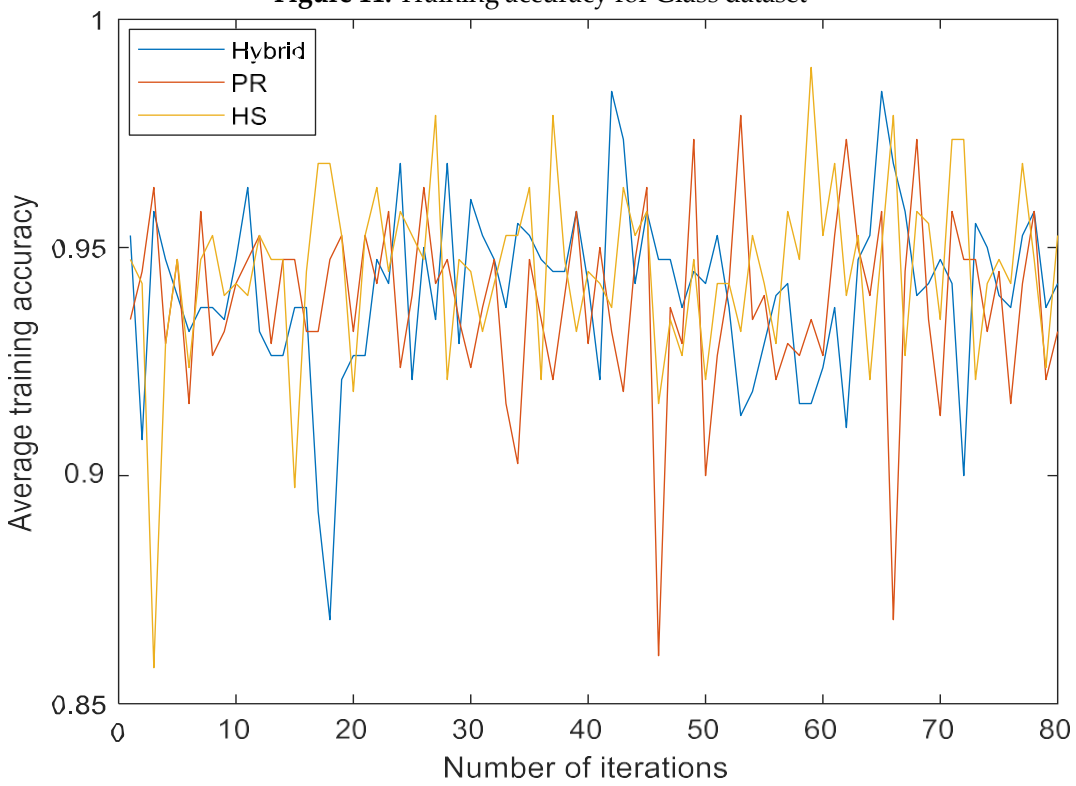

Figure 12. Final results of training errors for the Glass dataset

Table 4 shows similar improvements to Table 3 but for the glass data, where performance reaches its best level after 14 iterations. Figures 9-12 are similar to those shown in Figures 3-6. The results for the second set showed that the algorithm showed a training accuracy of 98.39% and a test accuracy of 94.04%, with continuous improvements in performance when compared to the individual methods. There was faster convergence and a continuous improvement in MSE compared to the traditional methods.

Table 5: Value the training for Wine

Unit	Initial value	Stopped value	Target value
Epoch	0	13	1000
Elapsed Time	-	00:00:01	-
Performance	0.621	0.0109	0
Gradient	0.735	0.0218	1e-07
Mu	0.001	1e-06	1e+10
Validation Checks	0	6	6

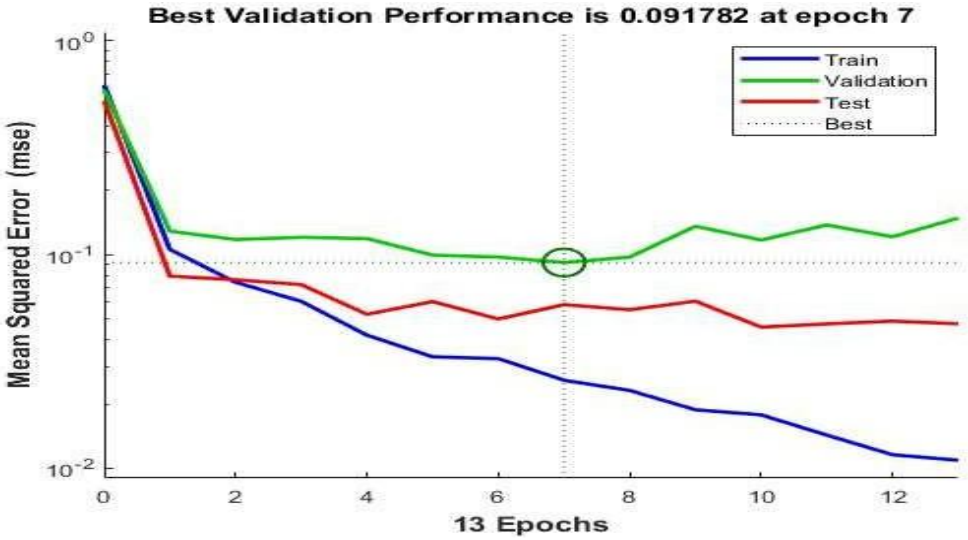

Figure 13. Optimal validation performance for Wine dataset at epoch 13

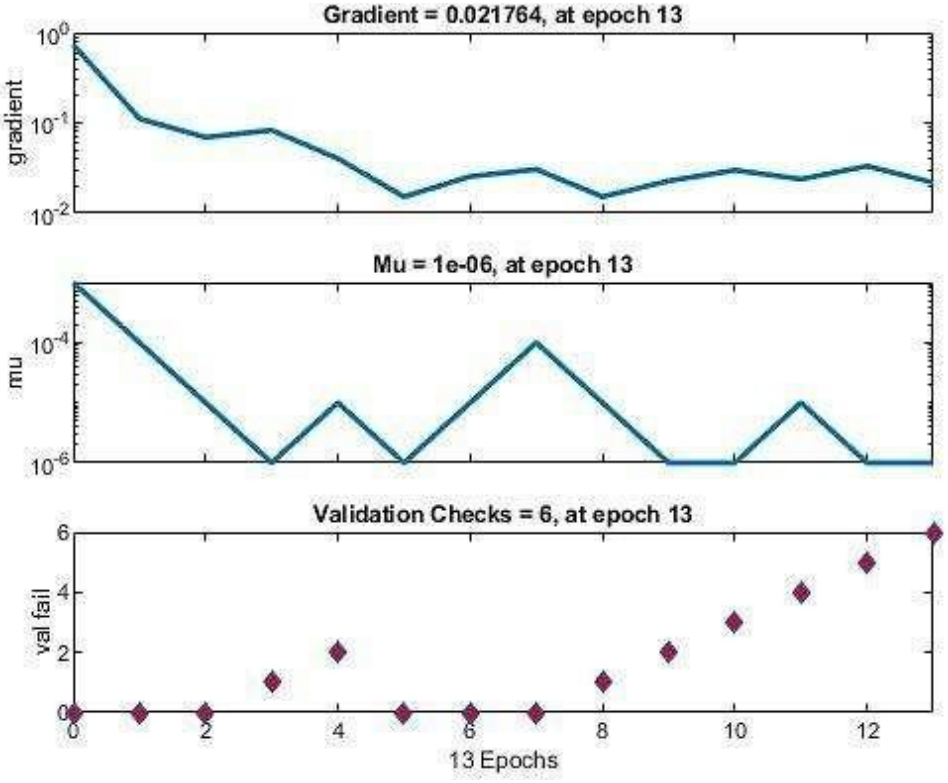

Figure 14. Training state for Wine dataset

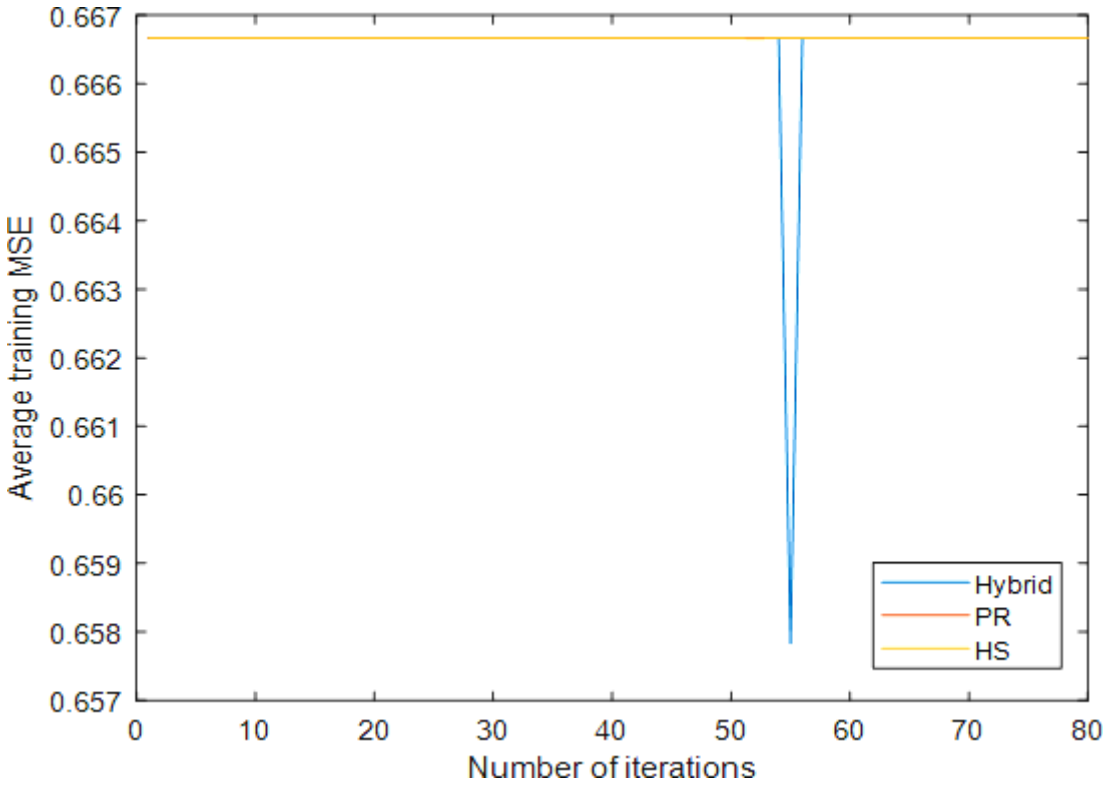

Figure 15. Mean Squared Error (MSE) during training for the Wine dataset

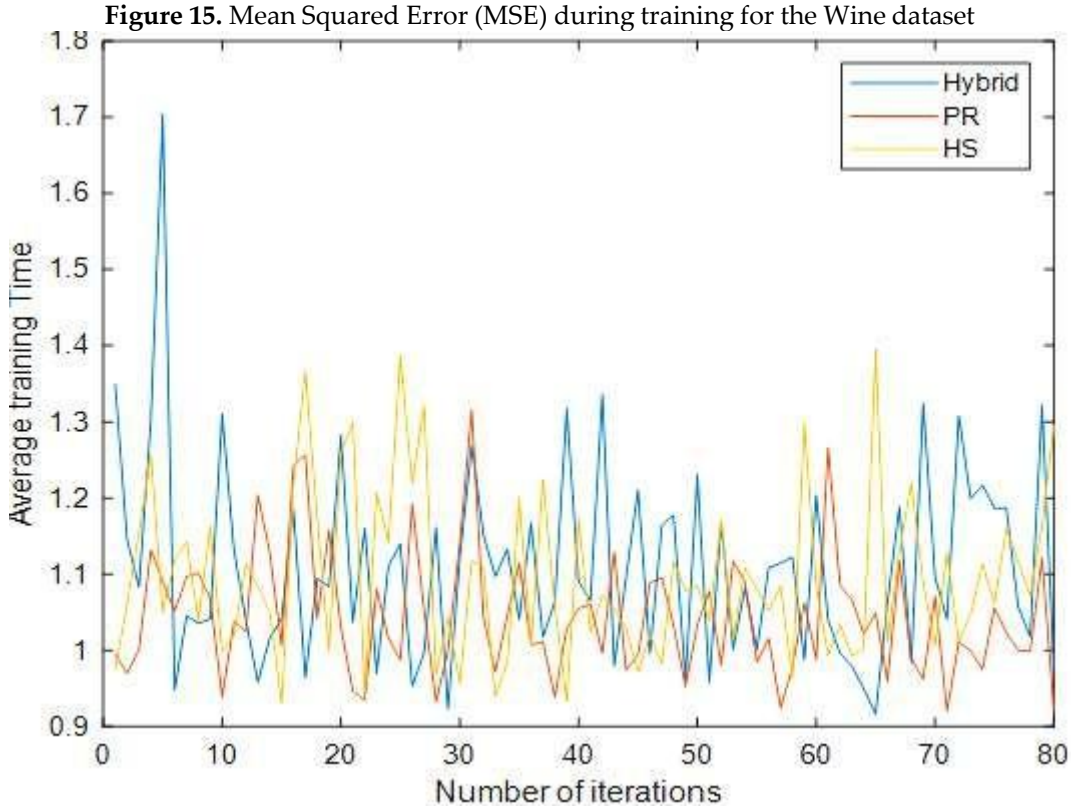

Figure 16. Time taken to train on Wine dataset

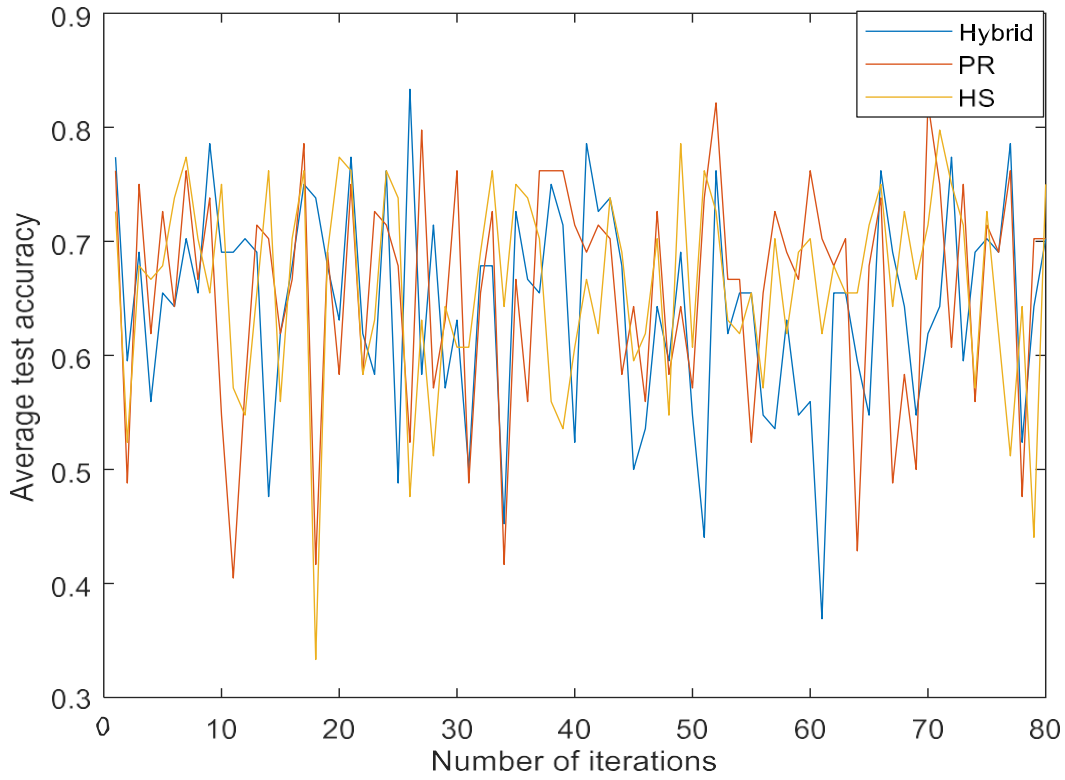

Figure 17. Training accuracy for Wine dataset

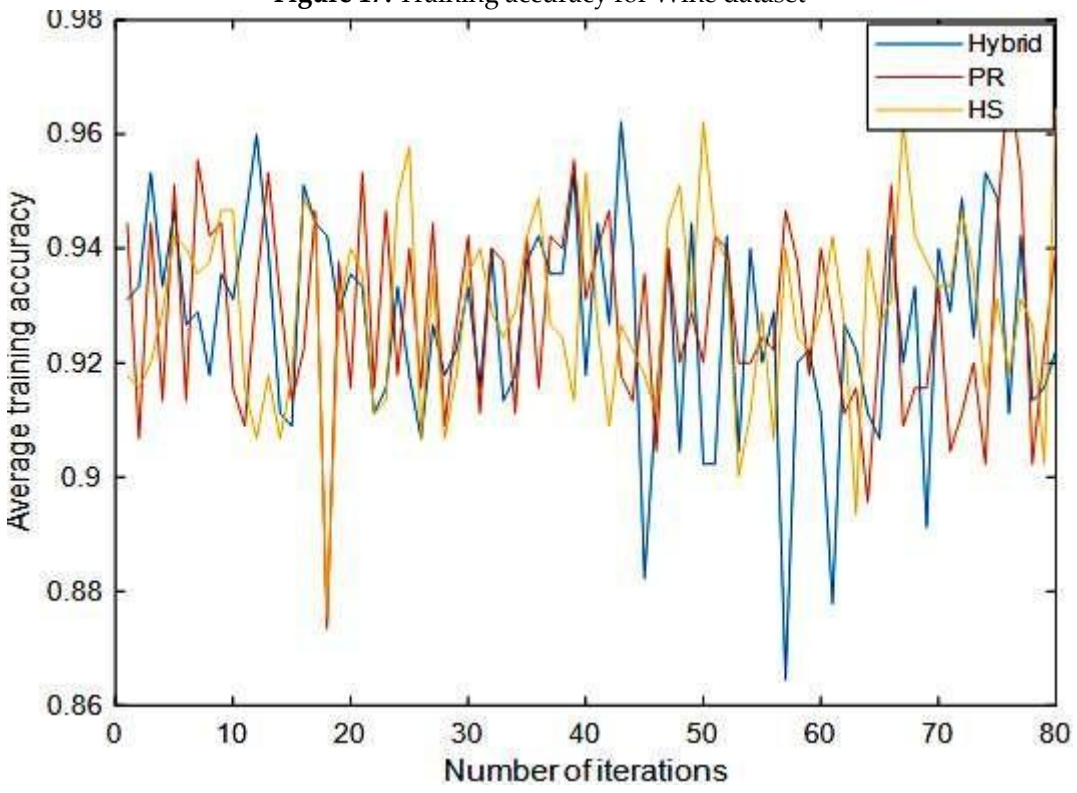

Figure 18. Final results of training errors for the Wine dataset

Similarly, Table 5 shows similar results to Tables 4 and 5, with the best performance achieved after 13 iterations. While the results for the third dataset showed that the algorithm achieved a test accuracy of 92.89%, with a decrease in MSE during training, indicating that the hybrid algorithm excels at handling more complex data.

.

3. Results and Discussion

The performance of hybrid algorithm evaluated using three different data set and compared with HS and PR, the results showed the speed of convergence and accuracy of the hybrid algorithm was better performance in terms of number of iterations and training time. The training accuracy to Iris was 98.5% and testing 98.67%, while the glass has 98.39% and testing 94.04%, and Wine was 65.98% and testing 92.89%. while the mean square error (MSE), which was the hybrid algorithm showed lowest MSE which was 0.665 and was better than HS and PR. hybrid algorithm showed significant reduction in training time across all data sets, making it more efficient in practice.

The hybrid algorithm showed high efficiency with relatively stable Iris and Glass, and with complex data Wine. The significant decrease in MSE across all sets of data reflects a hybrid algorithm ability to improve the FFNN model by taking advantage of dynamical adaptation. The algorithm demonstrated remarkable efficiency in handling data with high reflection or high volatility, this volatility its ability to meet challenges of nonlinear problems.

The hybrid algorithm can used to train more complex neural networks, as deep network also used to image classification, speech recognition, and language model. The hybrid algorithm can applied in adaptive control systems and structural engendering.

4. Conclusion

This study demonstrated the effectiveness of a developed a hybrid algorithm, which combine HS and PR methods using adaptive approach, in improving the training process of FFNN. The main results including improved convergence and accuracy performance of the hybrid algorithm, efficient mean MSE reduction, suitability for nonlinear data, and broad practical applicability. It's also recommended to test algorithm on different size of data sets including industrial and scientific applications also to combine the algorithm with techniques as genetic algorithms or evolutionary optimization to improve performance in more complex scenarios.

REFERENCES

[1] M. Benzi, "Solving large-scale linear systems: Methods and applications," *Journal of Computational and Applied Mathematics*, vol. 380, 2020.

[2] H. Zhu, "Structural optimization using conjugate gradient method with applications in engineering," *Engineering Reorts*, vol. 2, no. 3, 2020.

[3] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross-validation, regularization, bagging, and boosting: Tutorial," *Machine Learning: Science and Technology*, vol. 2, 2020.

[4] Fletcher, R. (2013). Practical methods of optimization. *John Wiley & Sons*.

[5] H. Zhu and X. Pan, "Structural optimization using conjugate gradient method with applications in engineering," *Engineering Reports*, vol. 2, no. 3, 2020.

[6] J. Liu, H. Zhang, and Y. Zhou, "A parallel algorithm for solving large linear systems in optimization," *Optimization Methods and Software*, vol. 35, no. 6, pp. 1133–1155, 2020.

[7] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross-validation, regularization, bagging, and boosting: A tutorial," *Machine Learning: Science and Technology*, vol. 2, no. 1, 2020.

[8] M. S. Jameel and Z. M. Abdullah, "A new shifted conjugate gradient method based on shifted quasi-Newton condition," *Journal of Physics: Conference Series*, vol. 1818, no. 1, 2021.

[9] L. Zhang and W. Zhou, "Two descent hybrid conjugate gradient methods for optimization," *J. Comput. Appl. Math.*, vol. 216, no. 1, pp. 251–264, 2008.

[10] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific Journal of Optimization*, vol. 2, no. 1, pp. 35–58, 2020.

[11] M. Al-Baali, "Improved convergence properties of the Fletcher-Reeves conjugate gradient method," *IMA Journal of Numerical Analysis*, vol. 5, no. 1, pp. 121–124, 2021.

[12] A. Al-Saidi, "Improved Fletcher-Reeves methods based on new scaling techniques," *Sultan Qaboos Univ. J. Sci. [SQUJS]*, vol. 26, no. 2, pp. 141–151, 2021.
[13] Z. M. Abdullah, H. M. Khudhur, and A. Khairulla Ahmed, "Modification of the new conjugate gradient algorithm to solve nonlinear fuzzy equations," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 27, no. 3, p. 1525, 2022.
[14] M. Azzam and N. Hawraz, "Four-Term Conjugate Gradient (CG) Method Based on Pure Conjugacy Condition for Unconstrained Optimization," *Kirkuk Journal of Science*, vol. 13, pp. 101–113, 2018.
[15] H. Mohammed and K. K. Abbo, "New hybrid of Conjugate Gradient Technique for Solving Fuzzy Nonlinear Equations," *Journal of Soft Computing and Artificial Intelligence*, vol. 2, no. 1, pp. 1–8, 2021.
[16] K. K. Abbo and H. M. Khudhur, "New A hybrid conjugate gradient Fletcher-Reeves and Polak-Ribiere algorithm for unconstrained optimization," *Tikrit J. Pure Sci.*, vol. 21, no. 1, pp. 124–129, 2023.
[17] H. M. Khudhur and K. K. Abbo, "A new type of Conjugate Gradient Technique for solving fuzzy nonlinear algebraic equations," *J. Phys. Conf. Ser.*, vol. 1879, no. 2, p. 022111, 2021.
[18] H. N. Jabbar, K. Khalil, and H. M. Abbo, "Four--term conjugate gradient (CG) method based on pure conjugacy condition for unconstrained optimization," *Univ. J. Sci. Stud*, vol. 13, pp. 101–113, 2018.
[19] Y. A. Laylani, K. Khalil, and H. M. Abbo, "Training feed forward neural network with modified Fletcher-Reeves method," *Journal of Multidisciplinary Modeling and Optimization*, vol. 1, pp. 14–22, 2018.
[20] K. K. Abbo and F. H. Mohammed, "Spectral Fletcher-Reeves Algorithm for Solving Non-Linear Unconstrained Optimization Problems," *Iraqi Journal of Statistical Sciences*, vol. 19, pp. 21–38, 2011.
[21] T. Gao, Z. Zhang, Q. Chang, X. Xie, P. Ren, and J. Wang, "Conjugate gradient-based Takagi-Sugeno fuzzy neural network parameter identification and its convergence analysis," *Neurocomputing*, vol. 364, pp. 168–181, 2019.
[22] F. Venturi, "How to choose a glass: the influence of glass parameters on the profile of different wines (full bodied red and rosé) during tasting," in *ISPROF 2015 Book of Abstracts 2nd International Symposium on Profiling*, Proteomass, 2015.
[23] L. Omelina, J. Goga, J. Pavlovicova, M. Oravec, and B. Jansen, "A survey of iris datasets," *Image Vis. Comput.*, vol. 108, no. 104109, p. 104109, 2021.