



Optimization Techniques for Crop Yield Prediction and Air Quality Management: Performance Analysis and Comparative Study

Kuruba Harish,¹ P. Nimmy,^{2,*} K.V. Nagaraja² and T.V. Smitha²

Abstract

Optimization of regression models remains essential to enhance both model precision and level of stability, as well as operational computational efficiency in datasets with large sizes and high noise levels. This paper examines several gradient-based optimization approaches: Batch Gradient Descent, Mini-Batch Gradient Descent, Stochastic Gradient Descent, RMSprop, Momentum, Adam, Nadam, Adagrad and Adadelata. The analysis studies how these methods perform regarding convergence speed and predictive precision, and stability control. Two datasets are used in the study including agricultural crop yield prediction data alongside air quality data for environmental analysis. The experimental outcomes show that adaptive optimization methods including Adam and Nadam achieve the best combination of stability and speed due to their superior effectiveness. The research delivers important data that helps users choose optimization methods according to their dataset features and computational needs to develop better regression modeling for practical uses.

Keywords: Regression optimization; Gradient descent algorithms; Adaptive learning rates; Convergence efficiency; Model stability; Large-scale datasets; Noise reduction.

Received: 22 May 2025; Revised: 21 August 2025; Accepted: 26 September 2025

Article type: Research article.

1. Introduction

Optimization is a part of regression models, and this allows one to break and adjust the parameters of the model to fit the data. This process usually seeks to optimize loss functions such as mean squared error for the distance between the predicted value of output and the actual data.^[1,2] Therefore, optimization is used to model the approximate relationship between the predictors and the target value, mainly to find the set of parameters. This is important to apply when regression techniques are applied in making estimates or predictions,^[3] which could be very crucial in sensitive areas such as health, finance, as well as engineering.^[4-8] The need for optimization in regression arises because of the churned-out real-world data, which may be complex and noisy. For regression tasks, there are usually numerous possible solutions for the model parameters. During training a model, it may fail to find the best

solution and hence perform poorly when such optimization has not been done.^[9,10] To do this, optimization guides a search through this vast space of parameters, to identify the parameters which minimize the loss function and increase the accuracy of the model.^[11] This process also helps in prediction for avoiding complications such as overfitting, underfitting, and other complications surrounding machine learning algorithms.^[12,13]

The importance of using optimization techniques in regression modelling is clearly expressed in its effects on modelling quality, stability, accuracy, and training time. Optimal convergence is the process of making sure that a regression formula gets the best solution possible to give the best possible result.^[14] Furthermore, it helps in dealing with large and higher order datasets to provide an optimal solution to assess the predictive model with low training time and costs.^[15] However, convergence acceleration and stability are important tasks for optimization to improve the applicability of regression models, as well as to increase their accuracy as much as possible in order to achieve an optimal balance with real-life problems when both response time and adaptive accuracy are needed.^[16-17]

The motivation for this work comes from the increased

¹Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru, Karnataka, 560035, India

²Computational Science Lab, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Bengaluru, Karnataka, 560035, India

*Email: p_nimmy@blr.amrita.edu (Nimmy P)

demand for accurate and practical regression models in sectors that face increased responsibilities, such as the health sector, the financial sector, and the engineering discipline, to mention but a few.^[18-19] The real-world data sets are growing large and noisy and, therefore, it becomes important to find optimization methods that enhance model performance. Problems like overfitting and underfitting, and computational issues can hamper the utilization of regression models, so optimization is important to ensure that these models can perform well in the future unseen data and to handle large and higher dimensional data sets.^[20]

This research helps to analyze the different types of optimization techniques for a regression model and analyze their performance in terms of accuracy, specifically, the number of iterations it takes and how stable the result is. It is within this context that this research aims to find out the best optimization technique to improve the accuracy of the given model in order to improve its ability to generalize as well as its flexibility in dealing with other different and perhaps larger datasets. The innovation of this work is the systematic study of different optimization techniques in terms of the ability to efficiently process high-dimensional noisy data and solve problems of overfitting, underfitting, and computational complexity. Thus, in turn, this research helps increase the efficiency of the chosen optimization methods and enhances the general practicability of regression models in crucial areas such as healthcare, finance, and engineering.

2. Related work

A complete analysis discusses how built-in gradient descent has some issues, including fixed learning rates, and presents adaptive methods like momentum techniques and gradient clipping to increase the convergence and standardization of the approach.^[21] There is evidence of how AdaGrad performs better than Adam under certain circumstances, like while dealing with through-silicon-via images, it provides faster convergence and fewer generalization errors.^[22]

In the domain of cellular networks, stochastic gradient descent (SGD)-based methods were proposed to optimize network coverage. These methods demonstrate computational efficiency and accuracy improvements compared to gradient-free approaches.^[23] Kida *et al.*^[24] investigated online fuzzy co-clustering, revealing that AdaGrad and Adam outperform traditional SGD in learning speed and stability, especially when paired with normalization techniques to handle large co-occurrence datasets. The work studying nonlinearity optimization looks at decay rates as being adaptive and the methods of initialization as being superior to Adadelta and Adam.^[25] When it comes to deep learning, researchers^[26] discovered that LSTM networks respond best to the Adam optimizer, which improves convergence compared to other algorithms like RMSprop, and efficiency in noisy time series data sets.

Mini-Batch Gradient is suitable for large-scale learning models, while L-BFGS provides good results as well. Mini-

Batch Gradient is in between computational complexity and convergence speed, while L-BFGS is best in solving problems of high dimensionality.^[27] Moreover, it was found that the most accurate learning method among the Gradient Descent and RMSprop in CNN-based disease detection model is Adam applied to apple leaf diseases detection in precision agriculture.^[28]

There are stand-offs to helping landmine detection in that they are inefficient and not very accurate. A new regression model used to analyze hyperspectral data promoted the increment of automated detection.^[29] Design parameters such as load have a vital impact on wear behaviour in tools, and sometimes it causes tool failure depending on the velocity. These parameters were systematically controlled in the study for TiAlN coatings using modern approaches to bring better wear protection and enhanced surface toughness under severe operating environments.^[30] Muthukani and Paramanathan^[31] addressed some weaknesses resulting from the scalarization method in solving the convex multi-objective problem and suggested an efficient method called the ellipsoid-based method for Pareto front coverage. Bhavani and Paramanathan^[32] optimized wear parameters for Duplex-TiAlN coated tool steel using fuzzy techniques to balance conflicting objectives effectively.

As optimization techniques are developed, there still exists a research gap in systematically comparing the efficiency of these methodologies within different and more complex real-world regression model scenarios.^[33] This includes: dimensions of noise, the ability to handle overfitting as well as underfitting, working in the domain sector, and industry. While there are many theoretical algorithms among these methods, there exists no systematic approach that can objectively qualify them according to their speed of convergence, computational cost, or generality of application to different domains such as medicine, economics, and engineering.

The following sections will detail the methodology used to evaluate several optimization techniques, such as Batch Gradient Descent, Mini-batch Gradient Descent, RMSprop, Momentum, Adam, Nadam, Adagrad, and Adadelta. These will be examined for training regression models on a given dataset, and the steps are discussed. After that, the results and discussion section will describe the evaluation of the above technique's performance considering model concerns, such as prediction accuracy, convergence, and stability. The results and trends will be discussed for each technique with respect to its performance and its applicability to regression problems. The pros and cons of each approach will also be discussed, paying particular attention to their application. Lastly, the conclusion section shall be a consolidation of the findings deduced from the research and shall explain the best trends in optimization procedures and how applicable the research findings are with regard to real-life regression problems.

3. Methodology

The methodology is based on linear regression that defines the connection between the inputs and outputs as the weighted sum of the input and a bias term. To improve the accuracy of the modeling process, Mean Squared Error (MSE) has been used as a measure of quantifying the cumulative error of the model, hence serving as the loss function. To provide a proof for the concavity of MSE, its second-order derivatives and the Hessian matrix are examined. This assure that there exists a unique global minimum and hence optimization algorithms such as the gradient descent converge appropriately towards the global minimum.

This research considers different gradient-based optimization methods employed in order to minimize the MSE and optimize model coefficients. These include basic methods like: Batch Gradient Descent, Mini-Batch Gradient Descent, Stochastic Gradient Descent and further development on them like, RMSProp, Momentum, Adam, Nadam, Adagrad and Adadelta. Definitions of all the techniques and update formulas are provided together with their description of how they handle learning rate adjustments, computational intensity and stability during training. Using these methods, the objectives which include getting to the best weights and bias and above all, getting the best convergence and stability in the test.

A. Model representation

The relationship between inputs (x) and output ($\hat{y}$) can be modeled as by Eq. (1):

$$\hat{y} = w_1x_1 + w_2x_2 + \dots + w_nx_n + b \quad (1)$$

where $x_1, x_2, \dots, x_n$ are the input features, $w_1, w_2, \dots, w_n$ are the weights corresponding to each feature and b is the bias term.

B. Loss function (Mean squared error)

The loss function is given as in Eq. (2):

$$L(w, b) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (2)$$

where n is the number of data points, $\hat{y}_i$ is the predicted output for the i -th sample, and y_i is the true output for the i -th sample.

C. Checking for convexity of the loss function

Curvature is beneficial because it suggests that the algorithm for optimization is going towards one global extremum, and will not get stuck at local optimizers and Saddle points. Analyzing the curvature of the Mean Squared Error (MSE) loss function, through its Hessian matrix, helps us confirm that such curvature is positive. The Hessian matrix is derived from the second-order partial derivatives of the loss function with the main parameters of the model, being the weight vector w and bias vector b .

The Eq. (3) represents the second derivative with respect to w and w (i.e., $\frac{\partial^2 L(w, b)}{\partial w^2}$):

$$\frac{\partial^2 L(w, b)}{\partial w^2} = \frac{2}{n} \sum_{i=1}^n x_i x_i^T = \frac{2}{n} X^T X \quad (3)$$

where X is the matrix of input features, and $X^T X$ is a positive semi-definite matrix because for any non-zero vector v , the condition is mentioned in Eq. (4):

$$v^T X^T X v = \|Xv\|^2 \geq 0. \quad (4)$$

The second derivative with respect to b and b is given in Eq. (5):

$$\frac{\partial^2 L(w, b)}{\partial b^2} = \frac{2}{n} \sum_{i=1}^n 1 = 2 \quad (5)$$

The Eq. (6) describes about the mixed second derivatives with respect to w and b (i.e., $\frac{\partial^2 L(w, b)}{\partial w \partial b}$ or $\frac{\partial^2 L(w, b)}{\partial b \partial w}$)

$$\frac{\partial^2 L(w, b)}{\partial w \partial b} = \frac{2}{n} \sum_{i=1}^n x_i = \frac{2}{n} X^T \mathbf{1} \quad (6)$$

where $\mathbf{1}$ is the vector of ones (corresponding to the bias term). The Hessian matrix $H(w, b)$ for the MSE loss function is given in Eq. (7):

$$H(w, b) = \begin{bmatrix} \frac{\partial^2 L(w, b)}{\partial w^2} & \frac{\partial^2 L(w, b)}{\partial w \partial b} \\ \frac{\partial^2 L(w, b)}{\partial b \partial w} & \frac{\partial^2 L(w, b)}{\partial b^2} \end{bmatrix} \quad (7)$$

Substituting the calculated second derivatives from Eq. (3), (5) and (6) in Eq. (7), $H(w, b)$ is obtained as follows in Eq. (8):

$$H(w, b) = \begin{bmatrix} \frac{2}{n} X^T X & \frac{2}{n} X^T \mathbf{1} \\ \frac{2}{n} \mathbf{1}^T X & 2 \end{bmatrix} \quad (8)$$

where X is the matrix of input features, and $\mathbf{1}$ is the vector of ones corresponding to the bias term.

The matrix $H(w, b)$ is block diagonal for a large part, and the term $X^T X$ is able to govern the convexity behavior in the case of w . The second derivative of the Hessian matrix $H(w, b)$ indicates that $H(w, b)$ is positive semi-definite. Given this observation, we conclude that the MSE loss function is convex, exhibiting positive curvature.

The MSE loss function, used for regression models, is a convex function and therefore also defines one global minimum. This property ensures that gradient-based optimization algorithms will reach the global minimum as long as one selects the correct hyperparameters, such as the learning rate. Hence, as a method of reducing the error and fine-tuning model parameters, the following gradient descent optimization methods will be used throughout the study: RMSProp, Adam, Nadam, Adagrad, Adadelta, Momentum, and the batch and mini-batch gradient descent. Each optimizer will then be discussed based on the convergence rate, stability

as the training progresses, and at different learning rates. The accuracy of the constructed model will be estimated by observing the value of MSE after each epoch, and the values of weights and biases corresponding to the epoch where the least value of MSE of the learned parameters will be chosen.

These optimal parameters among themselves will then be again tested on test data to know how much the model generalizes. The local concavity of the loss function as defined by its convexity is peaceful to accept as true that all types of

optimizer will subsequently arrive at the global minimum. However, the relative discrepancies such as efficiency, stability, and convergence character will offer important information on the efficiency of each of the above-mentioned methods giving minimal MSE, and maintaining stability in the training phase. This analysis aims to highlight the best optimization method for minimizing the error rate for the generalization of predictive performance. The outline of the gradient descent optimizers are depicted in Fig. 1.

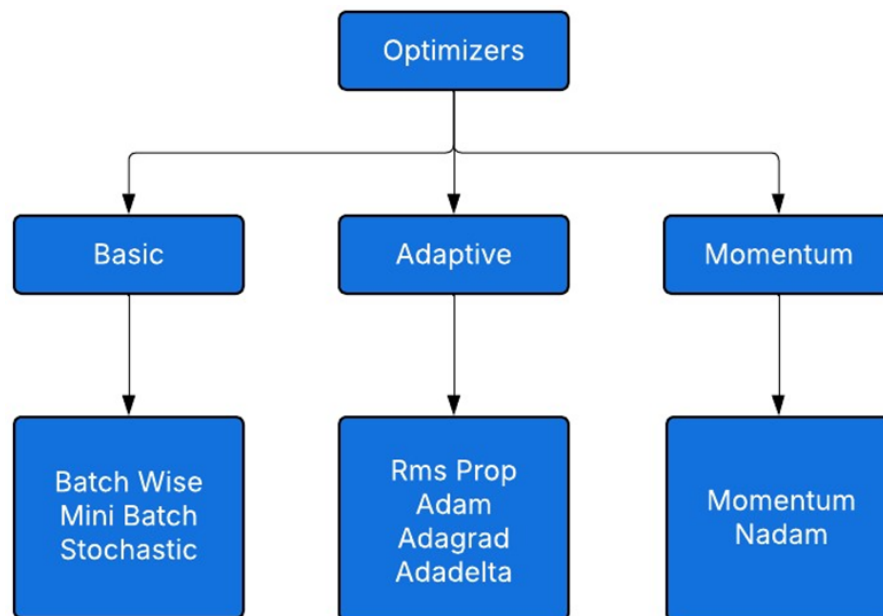


Fig. 1: Outline of Gradient Descent.

D. Gradient optimization techniques: definitions and formulations

1) Batch gradient descent (BGD)

Batch Gradient Descent calculates gradients over all the data points which makes it slow, but guarantees slow and stable convergence even with large datasets. It is less vulnerable to noise in the gradient due to using the full data, however, it is expensive to calculate even with large data.

The weights (w) and bias (b) update formulas are represented as follows in Eq. (9) and Eq. (10):

$$w = w - \eta \cdot \frac{1}{n} \sum_{i=1}^n \frac{\partial L(w, b; x_i)}{\partial w} \quad (9)$$

$$b = b - \eta \cdot \frac{1}{n} \sum_{i=1}^n \frac{\partial L(w, b; x_i)}{\partial b} \quad (10)$$

where n is the total number of training examples, η represents the learning rate, which controls the step size for updates, $L(w, b; x_i)$ demonstrates the loss function evaluated for the i -th data point given in Eq. (11):

$$\frac{\partial L(w, b; x_i)}{\partial w} \quad (11)$$

and Eq. (12) tells how much the loss will change with respect to each weight and

$$\frac{\partial L(w, b; x_i)}{\partial b} \quad (12)$$

conveys the change with respect to the bias term.

2) Mini-batch gradient descent

Mini-Batch Gradient Descent (MBGD) combines the efficiency of SGD and the stability of BGD; by using small batches of data when updating the model parameters it achieves quick and stable convergence when dealing with large data-sets, where the selection of the batch size is critical to reach the best performance. Here, the weights(w) and bias (b) update formulas are given as in Eq. (13) and Eq. (14):

$$w = w - \eta \cdot \frac{1}{m} \sum_{i=1}^m \frac{\partial L(w, b; x_i)}{\partial w} \quad (13)$$

$$b = b - \eta \cdot \frac{1}{m} \sum_{i=1}^m \frac{\partial L(w, b; x_i)}{\partial b} \quad (14)$$

where m is the size of the mini-batch, η is the learning rate, and $L(w, b; x_i)$ is the loss function with respect to the parameters w and b for the input x_i .

3) Stochastic gradient descent (SGD)

Stochastic Gradient Descent works on only one data point at a time, making it easy and fast to work with large datasets. Its ease of escape local minima is secondary to the fact that it can achieve very erratic convergence paths. The Eq. (15) and Eq. (16) represent the weights (w) and bias (b) updating the formulas:

$$w = w - \eta \cdot \frac{\partial L(w, b; x_i)}{\partial w} \quad (15)$$

$$b = b - \eta \cdot \frac{\partial L(w, b; x_i)}{\partial b} \quad (16)$$

where x_i is the individual data point. The gradient is computed with respect to the single data point x_i .

4) Root mean square propagation (RMSProp)

RMSProp normalises the learning rate of each parameter according to a moving average of squared gradients and so is suited to non-stationary and noisy figures. The learning rate decreases with large gradients and increases with small ones, stabilizing the training and averting exploding or diminishing gradients. The weights (w) and bias update formulas are represented in Eq. (17), (18), (19), and (20):

$$S_w = \beta S_w + (1 - \beta) \left(\frac{\partial L(w, b)}{\partial w} \right)^2 \quad (17)$$

$$S_b = \beta S_b + (1 - \beta) \left(\frac{\partial L(w, b)}{\partial b} \right)^2 \quad (18)$$

$$w = w - \frac{\eta}{\sqrt{S_w} + \epsilon} \cdot \frac{\partial L(w, b)}{\partial w} \quad (19)$$

$$b = b - \frac{\eta}{\sqrt{S_b} + \epsilon} \cdot \frac{\partial L(w, b)}{\partial b} \quad (20)$$

where the running averages S_w and S_b accumulate the squared gradients to adaptively adjust the learning rate, decay factor β controls the smoothness of the running average, typically $\beta = 0.9$ and ϵ is a small value (e.g., 10^{-8}) added for the numerical stability.

5) Momentum optimizer

Momentum Optimizer can be thought of as an augmentation of gradient descent in that we use previous update directions to our advantage. Momentum optimizer can be used to speed up further over flat areas and mitigate overfitting in low angle portions of the loss landscape. Here the weights (w) and bias (b) update formulas are given in Eq. (21), (22), (23), (24):

$$v_w = \gamma v_w + \eta \cdot \frac{\partial L(w, b)}{\partial w} \quad (21)$$

$$v_b = \gamma v_b + \eta \cdot \frac{\partial L(w, b)}{\partial b} \quad (22)$$

$$w = w - v_w \quad (23)$$

$$b = b - v_b \quad (24)$$

where γ (Momentum) is a fraction (usually 0.9) of the previous velocity, helping to carry forward momentum in parameter updates and v_w, v_b are the velocities which keeps track of the smoothed direction of parameter updates.

6) Adam (Adaptive moment estimation)

Adam combines Momentum and RMSProp by using both the mean and variance of gradients to adapt learning rates for each parameter. It is computationally efficient, handles large datasets well, and performs reliably with sparse gradients, making it widely preferred. The weights (w) and bias (b) update formulas are as follows:

First Moment Update (Eq. (25)):

$$m_w = \beta_1 m_w + (1 - \beta_1) \frac{\partial L(w, b)}{\partial w} \quad (25)$$

Second Moment Update (Eq. (26)):

$$v_w = \beta_2 v_w + (1 - \beta_2) \left(\frac{\partial L(w, b)}{\partial w} \right)^2 \quad (26)$$

Bias-Corrected Moments (Eq. (27)):

$$\hat{m}_w = \frac{m_w}{1 - \beta_1^t}, \quad \hat{v}_w = \frac{v_w}{1 - \beta_2^t} \quad (27)$$

Parameter Update (Eq. (28)):

$$w = w - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w} + \epsilon} \quad (28)$$

(Similar for b),

where the first moment (m_w, m_b) represents the running average of gradients, the second moment (v_w, v_b) gives the running average of squared gradients, Bias correction corrects initialization bias during early iterations, ϵ is a small constant added for numerical stability and β_1 and β_2 are the decay rates for the moments.

7) Nesterov-accelerated adaptive moment estimation (Nadam)

Nadam extends Adam by incorporating Nesterov momentum, which looks ahead at the gradient after a momentum step. This enhances convergence speed and can yield better optimization paths, especially in deep networks with complex loss landscapes. The weights (w) and bias (b) update formulas are stated in Eq. (29), (30), and (31):

$$m_w = \beta_1 m_w + (1 - \beta_1) \frac{\partial L(w, b)}{\partial w} \quad (29)$$

$$v_w = \beta_2 v_w + (1 - \beta_2) \left(\frac{\partial L(w, b)}{\partial w} \right)^2 \quad (30)$$

$$w = w - \eta v_w + \epsilon \left(\beta_1 m_w + (1 - \beta_1) v_w + \epsilon \cdot \frac{\partial L(w, b)}{\partial w} \right) \quad (31)$$

8) Adaptive gradient algorithm (Adagrad)

Adagrad adapts the learning rate for each parameter based on the inverse square root of the accumulated squared gradients, benefiting rare features. However, its learning rate can shrink excessively over time, potentially halting further learning. The weights (w) and bias (b) update formulas are mentioned in Eq. (32), (33), (34), and (35):

$$G_w = G_w + \left(\frac{\partial L(w, b)}{\partial w} \right)^2 \quad (32)$$

$$G_b = G_b + \left(\frac{\partial L(w, b)}{\partial b} \right)^2 \quad (33)$$

$$w = w - \eta \cdot G_w + \epsilon \cdot \frac{\partial L(w, b)}{\partial w} \quad (34)$$

$$b = b - \eta \cdot G_b + \epsilon \cdot \frac{\partial L(w, b)}{\partial b} \quad (35)$$

where cumulative gradient G_w, G_b gives the sum of squared gradients used to normalize the learning rate.

9) Adadelta

Adadelta improves Adagrad by using a moving average of squared gradients, preventing the rapid decay of the learning rate. It ensures self-tuning updates and maintains consistent learning progress, making it well-suited for machine learning tasks. The weights (w) and bias (b) update formulas are given in Eq. (36), (37), (38), and (39):

$$S_w = \beta S_w + (1 - \beta) \left(\frac{\partial L(w, b)}{\partial w} \right)^2 \quad (36)$$

$$S_b = \beta S_b + (1 - \beta) \left(\frac{\partial L(w, b)}{\partial b} \right)^2 \quad (37)$$

$$w = w - \eta S_w + \epsilon \cdot \frac{\partial L(w, b)}{\partial w} \quad (38)$$

$$b = b - \eta S_b + \epsilon \cdot \frac{\partial L(w, b)}{\partial b} \quad (39)$$

where the moving average S_w and S_b track the gradient magnitude over recent updates, S_w is the moving average for the weight updates, while S_b is for the bias updates.

4. Experimental setup

The study therefore aims at comparing the different gradient-based optimization techniques while using different datasets. The emphasis is to assess the efficiency of the optimizers regarding convergence rate, stability, and accuracy of the forecast. When comparing the various datasets, the goal is to determine the best optimizer for the specific situation, while putting special focus on the MSE values. This is achieved in the hope of discovering which of the optimization strategies provides the best results in different data characteristics. The flow chart of the study is given in Fig. 2 and is described below:

A. Datasets

The evaluation of gradient-based optimization techniques

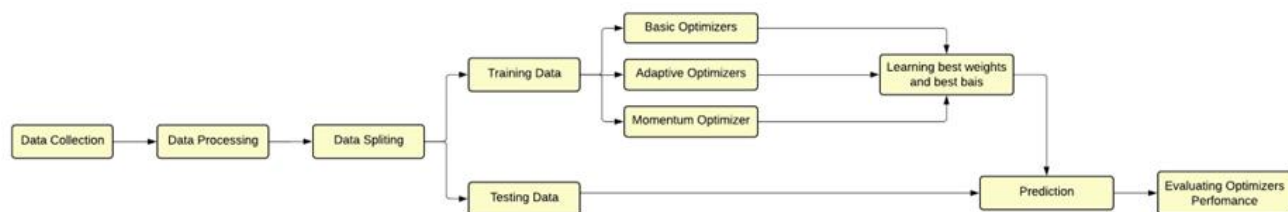


Fig. 2: Flow Chart of Research.

occurred using two datasets, which showed different sample characteristics:

- **Agriculture Dataset:** The dataset covers agricultural data that includes state-wise information with records about crop names alongside soil nutrients (N, P, K) and pH levels, as well as rainfall and temperature statistics and farming area data and production counts, and yield figures. Using the dataset enables predictions of crop yields and provides fertilizer recommendations that produce better efficiency in farming operations at the regional level. Data accuracy together with seasonal adjustments and handling of regional variations are essential factors to achieve an accurate understanding.^[34]
- **quantitative measurements about air quality as well as connected environmental variables.** This dataset presents information about pollutants and measures particulate matter concentrations along with weather parameters for analyzing

environmental health conditions. The columns are aqi (air quality index), pollutant, status, so2(sulphur dioxide), co (carbon monoxide), o3(ozone, o3 8hr, pm10(pollutant), pm2.5, no2 (nitrogen dioxide), nox, no (nitrogen monoxide), windspeed, winddirec, co 8hr, pm2.5 avg, pm10 avg, so2 avg.^[35]

Both datasets required preprocessing techniques to scale variables before the IQR method removed outliers, and feature selection completed the set for regression purposes.

B. Optimization techniques

- **Basic Techniques:** Gradient Descent (GD), Stochastic Gradient Descent (SGD), and Mini-Batch Gradient Descent.
- **Adaptive Methods:** Adam, RMSProp, Adagrad, and Adadelta.
- **Advanced Techniques:** Nadam, Momentum.

C. Evaluation metrics

The optimizers were assessed using:

- **Convergence Rate:** The iterative running time needed to meet this loss threshold criterion. In this case, when loss is no longer decreasing, that is the threshold condition.
- **Stability:** Variability in convergence path for each optimiser
- **Accuracy (R^2 Score):** It is a measure of model performance and is defined as given in Eq. (40).

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (40)$$

where y_i is the actual values $\hat{y}_i$ is the Predicted values, y is the mean of actual values, and n is the number of data points. Models with better predictive accuracy maintain an R^2 score approaching 1, whereas scores closer to 0 indicate poor data explanation.

D. Experimental workflow

- **Model Initialization:** Models were initialized with random weights for each optimizer.

- **Training:** The optimization methods were operated for a predetermined number of training cycles across all models. Each data set received specific learning rate and batch size adjustments.
- **Evaluation:** After training, the model was evaluated using pre-defined evaluation metrics on test data.

E. Comparison approach

The focus was to evaluate whether each data set requires identification of its optimizer that results in the minimum MSE value, and to evaluate the relationship between fast theoretical system convergence and steady operations.

5. Results and discussion

The performance of different optimization techniques, including RMSProp, Momentum, Adam, Nadam, Adagrad, Adadelta, Batch Gradient Descent, and Mini-Batch Gradient Descent. The parameters used were accuracy, stability, and convergence rate.

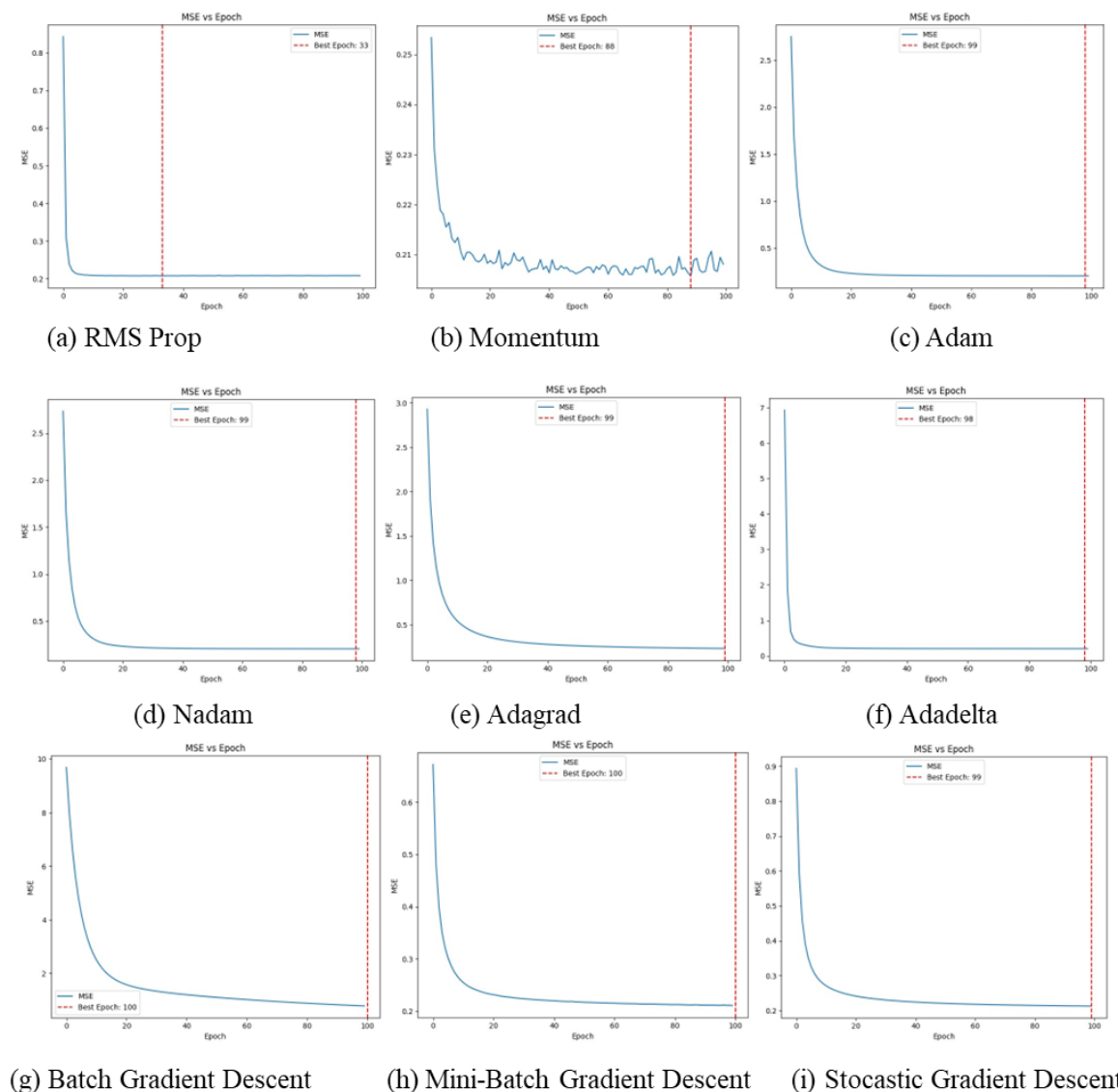


Fig. 3: Comparison of Different Optimization Techniques for Crop Yield Production Data.

Table 1: Comparison of Optimization Methods for Crop Yield Prediction.

Method	Accuracy (R ²)	R ² (Mean ± Std)	95% CI (R ²)	Best Epoch	Total Epochs	Stability	Convergence Speed
RMSProp	0.79906	0.7988±0.0002	0.7987- 0.7989	33	100	Stable	Fast
Momentum	0.80019	0.8002±0.0002	0.8001- 0.8003	88	100	Less stable	Slow
Adam	0.800482	0.8007±0.0002	0.8006-0.8008	99	100	Stable	Fast
Nadam	0.800481	0.8007±0.0002	0.8006- 0.8008	99	100	Stable	Fast
Adagrad	0.772135	0.7832±0.0136	0.7781- 0.7883	99	100	Stable	Fast
Adadelata	0.79827	0.7981±0.0012	0.7977-0.7985	98	100	Stable	Medium
Batch Gradient Descent	0.205469	0.22810±0.1171	0.2373-0.3248	100	100	Very stable	Very Slow
Mini-Batch Gradient Descent	0.7948	0.7960±0.0013	0.7955-0.7965	100	100	stable	Fast
Stochastic Gradient Descent	0.792899	0.7929±0.0019	0.7922-0.7935	99	100	Very stable	Fast

The optimization methods used for crop yield production data, perform distinctively regarding their convergence speed and accuracy, as well as stability levels based on Table 1 and Fig. 3. From Table 1, it is clear that Adam and Nadam optimizers showcases the highest R^2 value of about 0.8005, and also RMSprop and momentum exhibits a very close value of about 0.8. But compared with the other optimisers, batch gradient descent performs very poorly having a R^2 value around 0.206. The stability of the RMSprop, Adam, Nadam, Adagrad, Adadelata and Mini-Batch and stochastic gradient descent possess high stability which makes them converge consistently. From Fig. 3, it is evident that the momentum oscillates too much before converging, which showcase its less stability. As out of all the optimization techniques, Adam and Nadam are outperforming others with a high accuracy of R^2 value as 0.8005 and also because of their high stability and fast convergence, we can conclude that Adam and Nadam is

the best choice for the predicting the crop yield. On the other hand, since Batch Gradient Descent is having very slow convergence speed with a less accuracy of 0.255, it can be concluded that Batch Gradient Descent is not at all suitable. To mention about the statistical significance, we have run the optimiser 30 times with numerous random seeds and checked the value of R^2 for each case. Along with this, we have calculated the 95% confidence interval (CI) to identify the range of R^2 . From the results, it is clear the mean R^2 for RMSprop is 0.7988 with a standard deviation of 0.0002. The low value 0.0002 indicates that the optimiser gives almost the same R^2 values. And the 95% CI represents that the R^2 value will lie in the range of 0.7987- 0.7989. For optimizers like Adam or Nadam, we can see that R^2 is lying outside 95% CI. But that does not mean the result is wrong, it indicates that some individual runs can give R^2 value within that range.

The comparison of optimization methods for Air Quality

Table 2: Comparison of optimization methods for air quality data.

Method	Accuracy (R ²)	Best Epoch	Total Epochs	Stability	Convergence Speed
RMSprop	0.9440	58	100	Stable	Fast
Momentum	0.9439	97	100	Stable	Moderate
Adam	0.9442	82	100	Stable	Fast
Nadam	0.9442	95	100	Stable	Moderate
Adagrad	0.9441	199	200	Stable	Slow
Adadelata	0.9441	26	100	Less Stable	Fast
Batch Wise	0.9393	1000	1000	Less Stable	Very Slow
Mini Batch	0.9429	100	100	Stable	Moderate
Stochastic	0.9442	99	100	Stable	Slow

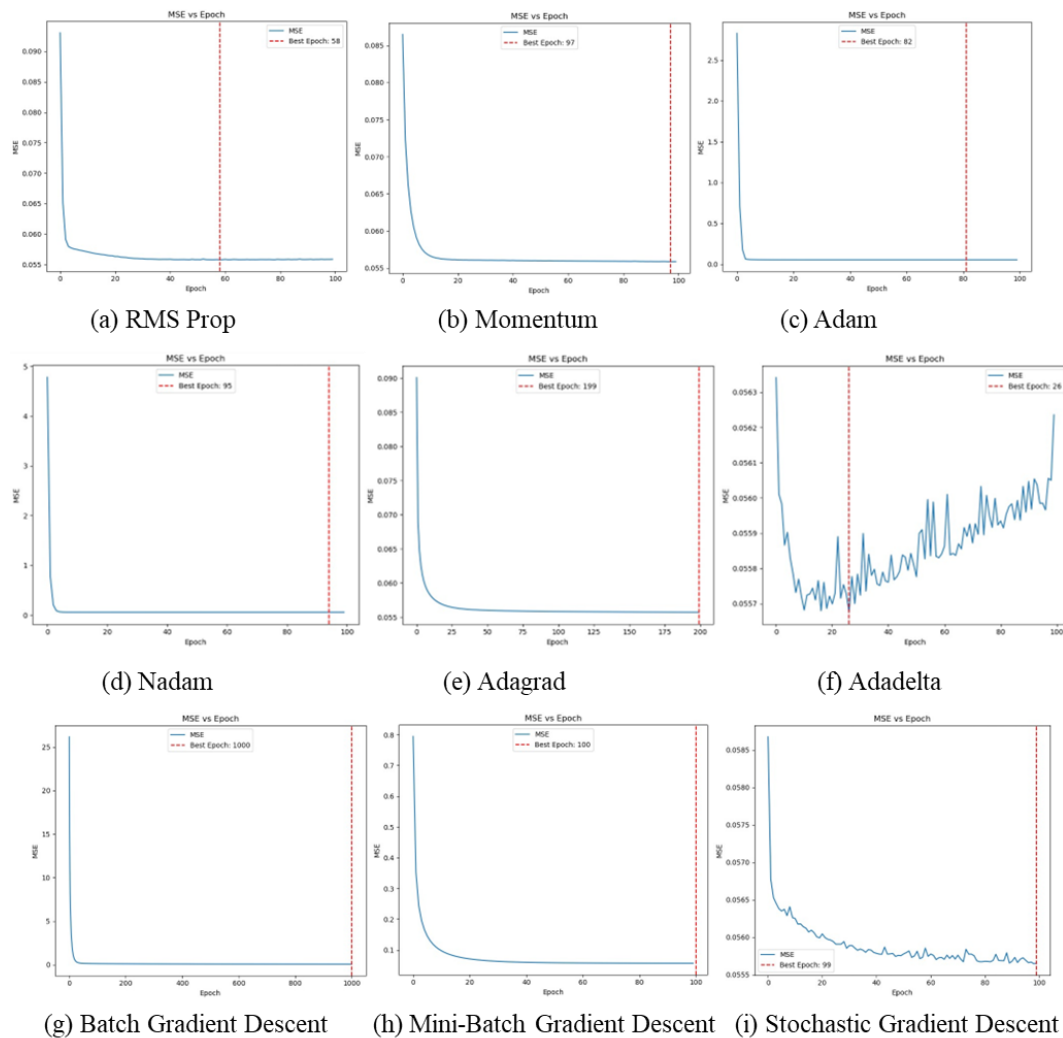


Fig. 4: Comparison of Different Optimization Techniques for Air Quality Data.

Data from Table 2 and Fig. 4 reveals that Adam and Nadam prove to be the optimal optimization algorithms given their strong accuracy ($R^2 = 0.9442$) and stable convergence together with a reasonable training time. RMSprop and Momentum demonstrate reliable options because they perform stably, even though their accuracy level is slightly below other methods. The fastest convergences belong to Adadelata at epoch 26, but the MSE behaves unstably after reaching optimal parameters. The computational performance of Batch Wise is the worst because it needs 1000 training epochs to complete with an accuracy rate of 0.9393. Adagrad shows stability throughout its learning process, yet requires a large number of 199 epochs for convergence. On its path to stability, SGD maintains high accuracy during the 99 epochs. Adam (82 epochs) showcases a desirable combination of efficiency with high accuracy, together with RMSprop (58 epochs) to achieve speedy convergence. Adam, along with Nadam, demonstrates superior performance by delivering superior convergence, stability, and accuracy at the same time.

6. Conclusion

Different optimization techniques display special

characteristics when used to process crop yield production data and manage air quality data. The crop yield production data achieves performance excellence through RMSProp, Momentum, and Mini Batch Gradient Descent because the combination delivers good speed, stability, and accuracy, while Mini-Batch Gradient Descent and Momentum demonstrate quick convergence and stable results. The convergence speed of Batch Gradient Descent remains slow, even though this method provides stability. Adam and Nadam provide the most effective solution for the Air Quality Data by showing high accuracy, together with stable convergence and efficient training times. When implemented, RMSProp with Momentum demonstrates consistent operation, but Adam and Nadam achieve better results in terms of accuracy. The convergence speed of Adadelata is fast, yet instability occurs when it reaches the optimal parameter range, while Batch Gradient Descent and Adagrad demonstrate slower convergence rates, leading to lower accuracy levels in the case of air quality data. The optimization methods demonstrate different compromise factors between the two datasets, yet Adam and Nadam stand out consistently as superior choices because they provide fast processing and steady system

operation alongside high accuracy rates. RMSProp, together with Momentum and Mini-Batch Gradient Descent establish suitable replacements which maintain both performance quality and operational speed, yet fail to equal the total efficiency of Adam and Nadam. The combination of slower convergence speed with reduced accuracy makes Batch Gradient Descent and Adagrad less effective than other methods for reaching optimal results in both datasets.

Acknowledgments

The authors express their sincere gratitude to the esteemed referees for their insightful comments and suggestions, which have significantly improved the quality of this paper. The authors gratefully acknowledge Sri Mata Amritanandamayi Devi (Amma), Chancellor, Amrita Vishwa Vidyapeetham, for her inspiration and for providing financial support for the Article Processing Charges (APC) of this publication.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

CRediT Statement

Kuruba Harish: Conceptualization, Methodology, Validation, Visualization, Writing - Original draft. **P. Nimmy:** Writing - original draft, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis and Data curation, Conceptualization. **K. V. Nagaraja:** Supervision, Conceptualization, Methodology, Visualization, Review & editing. **T. V. Smitha:** Investigation, Formal analysis, Writing - Review & editing.

References

- [1] Y. Han, S.-J. Li, Stability of the approximate solution sets for set optimization problems with the perturbations of feasible set and objective mapping, *Optimization*, 2024, **73**, 3755-3785, doi: 10.1080/02331934.2023.2231492.
- [2] T. O. Hodson, T. M. Over, S. S. Foks, Mean squared error, deconstructed, *Journal of Advances in Modeling Earth Systems*, 2021, **13**, e2021MS002681, doi: 10.1029/2021MS002681.
- [3] A. Patil, P. Mahalle, Machine learning for sustainable battery optimization: a data driven approach, *Engineered Science*, 2024, **32**, 1317, doi: 10.30919/es1317.
- [4] U. Sadhana, T. Singh, B. B. M, Swarm-optimized deep convolutional networks for precise detection and grading of diabetic retinopathy, *4th International Conference on Advancement in Electronics & Communication Engineering (AECE)*, November 22-23, 2024, Ghaziabad, India, 2024, 663-667, doi: 10.1109/AECE62803.2024.10911244.
- [5] L. Q. Anh, V. T. Tai, T. N. Tam, Convergence of solutions to nonlinear nonconvex optimal control problems, *Optimization*, 2024, **73**, 3859-3897, doi: 10.1080/02331934.2023.2239831.
- [6] A. Das, N. Neelima, K. Deepa, T. Özer, Gene selection based cancer classification with adaptive optimization using deep learning architecture, *IEEE Access*, 2024, **12**, 62234-62255.
- [7] R. Michael, J. Kæstel-Hansen, P. Mørch Groth, S. Bartels, J. Salomon, P. Tian, N. S. Hatzakis, W. Boomsma, A systematic analysis of regression models for protein engineering, *PLoS Computational Biology*, 2024, **20**, e1012061, doi: 10.1371/journal.pcbi.1012061.
- [8] S. Varun.B.p, I. Jonnalagadda, V. V. Ch, V. C S, A. Sahay, The impact of optimization techniques on cancer classification with gene expression data, *11th International Conference on Advances in Computing and Communications (ICACC)*, Kochi, India, November 6-8, 2024, 1-6, doi: 10.1109/ICACC63692.2024.10845662.
- [9] M. Ilchi Ghazaan, A. Salmani Oshnari, A. Salmani Oshnari, A novel adaptive optimization scheme for advancing metaheuristics and global optimization, *Swarm and Evolutionary Computation*, 2024, **91**, 101779, doi: 10.1016/j.swevo.2024.101779.
- [10] P. Echtenbruck, M. Echtenbruck, J. Batenburg, T. Bäck, B. Naujoks, M. Emmerich, Optimally weighted ensembles of regression models: Exact weight optimization and applications, 2022, doi: 10.48550/arXiv.2206.11263.
- [11] E. Carrizosa, C. Molero-Río, D. Romero Morales, Mathematical optimization in classification and regression trees, *TOP*, 2021, **29**, 5-33, doi: 10.1007/s11750-021-00594-1.
- [12] S. Saleem, M. Gallagher, Using regression models for characterizing and comparing black box optimization problems, *Swarm and Evolutionary Computation*, 2022, **68**, 100981, doi: 10.1016/j.swevo.2021.100981.
- [13] M. Salama, Optimization of regression models using machine learning: a comprehensive study with scikit-learn, *International Uni-Scientific Research Journal*, 2024, **5**, 119-129, doi: 10.59271/s45500.024.0624.16.
- [14] S. Selvarajan, A comprehensive study on modern optimization techniques for engineering applications, *Artificial Intelligence Review*, 2024, **57**, 194, doi: 10.1007/s10462-024-10829-9.
- [15] D. Bertsimas, I. Paskov, Stable regression: On the power of optimization over randomization in training regression problems, *Journal of Machine Learning Research*, 2020, **21**, 1-25.
- [16] A. Taheri, K. RahimiZadeh, A. Beheshti, J. Baumbach, R. V. Rao, S. Mirjalili, A. H. Gandomi, Partial reinforcement optimizer: an evolutionary optimization algorithm, *Expert Systems with Applications*, 2024, **238**, 122070, doi: 10.1016/j.eswa.2023.122070.
- [17] S. Sun, Z. Cao, H. Zhu, J. Zhao, A survey of optimization methods from a machine learning perspective, *IEEE Transactions on Cybernetics*, 2020, **50**, 3668-3681, doi: 10.1109/TCYB.2019.2950779.
- [18] A. Jadon, A. Patil, S. Jadon, A comprehensive survey of regression-based loss functions for time series forecasting, *Data Management, Analytics and Innovation*, Springer Nature Singapore, 2024, 117-147, doi: 10.1007/978-981-97-3245-6_9.
- [19] J. Kaur, H. D. Shelke, S. R. Jadkar, H. M. Pathan, R. Kumar, Optimization of photovoltaic characteristics of CIGS/Si heterojunction solar cells, *ES Energy & Environment*, 2022, **17**,

56-63, doi: 10.30919/esee8c743.

- [20] L. Weng, R. Zhu, D. Wu, B. Ding, B. Zheng, J. Zhou, Eraser: eliminating performance regression on learned query optimizer, *Proceedings of the VLDB Endowment*, 2024, **17**, 926-938, doi: 10.14778/3641204.3641205.
- [21] S. Ruder, An overview of gradient descent optimization algorithms, 2017, doi: 10.48550/arXiv.1609.04747.
- [22] J.-K. Fang, C.-M. Fong, P. Yang, C.-K. Hung, W.-L. Lu, C.-W. Chang, AdaGrad gradient descent method for AI image management, *IEEE International Conference on Consumer Electronics - Taiwan*, Taoyuan, Taiwan, September 28-30, 2020, 1-2, doi: 10.1109/ICCE-Taiwan49838.2020.9258085.
- [23] Y. Liu, W. Huangfu, H. Zhang, K. Long, An efficient stochastic gradient descent algorithm to maximize the coverage of cellular networks, *IEEE Transactions on Wireless Communications*, 2019, **18**, 3424-3436, doi: 10.1109/TWC.2019.2914040.
- [24] K. Kida, S. Ubukata, A. Notsu, K. Honda, Comparison of gradient descent methods in online fuzzy co-clustering, *International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, New Taipei, Taiwan, China, November 7-10, 2019, 9-14, doi: 10.1109/ifuzzy46984.2019.9066194.
- [25] T. Watanabe, H. Iima, Nonlinear optimization method based on stochastic gradient descent for fast convergence, *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan, October 7-10, 2018, 4198-4203, doi: 10.1109/SMC.2018.00711.
- [26] Y. O. Hodlevskiy, T. A. Vakaliuk, Optimal gradient descent algorithm for LSTM neural network learning, *IEEE 4th International Conference on Smart Information Systems and Technologies (SIST)*, Astana, Kazakhstan, May 15-17, 2024, 245-249, doi: 10.1109/SIST61555.2024.10629398.
- [27] Y. Gao, J. Li, Y. Zhou, F. Xiao, H. Liu, Optimization methods for large-scale machine learning, *18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China, December 17-19, 2021, **60**(2), 304-308, doi: 10.1109/iccwamtip53232.2021.9674150.
- [28] A. Murthy, P. S. Rao, N. S. Pallavi, N. Kharvi, B. R. Neha, K. Poojary, Optimizing convolutional neural networks: a comparative study of gradient-descent, Adam, and RMSprop optimizers for accuracy and loss in apple leaf disease detection, *Second International Conference on Networks, Multimedia and Information Technology (NMITCON)*, Bengaluru, India, August 9-10, 2024, 1-6, doi: 10.1109/NMITCON62075.2024.10699138.
- [29] R. Anand, J. Andrew, I. Makki, Optimizing land mine detection across diverse mining environments: a hyperspectral data approach with regression models, *International Journal of Intelligent Networks*, 2024, **5**, 351-363, doi: 10.1016/j.ijin.2024.08.004.
- [30] S. Kumar, S. R. Maity, L. Patnaik, Effect and optimization of tribological parameters upon wear responses of TiAlN coating, *Emerging Materials Research*, 2023, **12**, 68-77, doi: 10.1680/jemmr.22.00015.
- [31] M. Muthukani, P. Paramanathan, A subgradient supported

ellipsoid method for convex multiobjective optimization problems, *OPSEARCH*, 2025, **62**, 1239-1261, doi: 10.1007/s12597-024-00849-y.

- [32] R. Bhavani, P. Paramanathan, An improved secant-like method and its convergence for univariate unconstrained optimization, *Numerical Optimization in Engineering and Sciences*, Springer Singapore, Singapore, 2020, 339-345, doi: 10.1007/978-981-15-3215-3_33.
- [33] W. L. Cava, P. Orzechowski, B. Burlacu, F. O. de Franc, a, M. Virgolin, Y. Jin, M. Kommenda, J. H. Moore, Contemporary symbolic regression methods and their relative performance, 2021, doi: 10.48550/arXiv.2107.14351.
- [34] S. Patrick, Agriculture crop production in India, 2023, <https://www.kaggle.com/datasets/asishpandey/crop-production-in-india/code>.
- [35] T. wei Lo, Taiwan air quality index data 2016 ~ 2024, <https://www.kaggle.com/datasets/taweilo/taiwan-air-quality-data-20162024/code>.

Publisher's Note: Engineered Science Publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits the use, sharing, adaptation, distribution and reproduction in any medium or format, as long as appropriate credit to the original author(s) and the source is given by providing a link to the Creative Commons license and changes need to be indicated if there are any. The images or other third-party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

©The Author(s) 2025.