NeuroFuzzyAdam: A Fuzzy-Enhanced Adaptive Optimization Algorithm for Deep Learning

SUSILO HARIYANTO*, SITI KHABIBAH, RETNO PUTRI DWI RAHMAWATI,
BIBIT WALUYO AJI
Department of Mathematics,
Diponegoro University,
Jl. Prof. H. Soedarto, SH. Tembalang, Semarang, 50275,
INDONESIA

Abstract: - Deep neural networks are commonly trained using adaptive optimization methods such as Adam because they converge quickly and perform well under stochastic training conditions. Despite these advantages, recent research has revealed several important drawbacks of Adam. In particular, the optimizer tends to converge toward sharp minima, exhibits oscillatory behavior during loss optimization, and can produce unstable parameter updates when training in highly noisy environments. In this work, we propose NeuroFuzzyAdam (NF Adam), a novel optimizer that integrates fuzzy logic-based scalar modulation directly into the Adam update rule to improve learning stability and generalization. By introducing a bounded correction term via hyperbolic tangent transformations, NF Adam adaptively regulates step sizes based on moment estimates while preserving Adam's key strengths. We provide theoretical convergence guarantees under convexity and bounded-gradient assumptions, matching Adam's O(1/T) regret bound. Empirically, we evaluate the NF Adam across both image (MNIST, Fashion-MNIST, CIFAR-10) and tabular (UCI Iris, Wine, Breast Cancer) datasets using consistent architectures. Results show that NF Adam improves training smoothness, reduces gradient norm variance, and achieves competitive classification accuracy compared to Adam, SGD, RMSprop, and AdamW. These findings suggest that incorporating fuzzy logic into optimization dynamics presents a promising direction for enhancing robustness in neural network training, especially in scenarios with high gradient noise or non-smooth loss surfaces.

Key-Words: - Adaptive optimization, Adam optimizer, Fuzzy logic, Deep learning, Fuzzy scalar modulation.

Received: March 13, 2026. Revised: May 27, 2026. Accepted: June 15, 2026. Published: September 4, 2026.

1 Introduction

Training deep neural networks, especially for modern computer vision tasks, demands robust and efficient optimization. Stochastic gradient descent (SGD) and its momentum variants have long been the workhorses of deep learning, but the complexity of loss landscapes in large models has motivated adaptive methods such as Adam. Adam's elementwise learning rates and bias correction often yield fast convergence, particularly in noisy-gradient regimes [1], making it a default choice in many frameworks. However, adaptive methods exhibit well-known limitations: they often find sharper minima with worse generalization than SGD [2], and in some settings they can even diverge or oscillate. For example, Reddi et al. demonstrate that Adam may fail to converge without modification [3], and recent analyses observe a recurring "epochal sawtooth" oscillation in loss when using Adam on noisy data, [1]. Adaptive optimizers can also suffer from unstable or extreme effective learning rates [4], which can degrade performance on complex vision models. Despite fixes like AMSGrad or AdamW, which prove convergence or improve generalization [5], [6], there remains a need for new mechanisms that stabilize training under noise and sharp loss geometry.

A rich literature has explored improvements to Adam and related methods. AdaGrad and RMSProp pioneered per-parameter adaptation [7], and Adam extended these with momentum [8]. Subsequent algorithms (AMSGrad, AdaBound, RAdam, AdaBelief) attempt to balance fast convergence with generalization. For example, AdaBelief explicitly tunes steps based on "belief" in the gradient, yielding fast training and generalization on image classification benchmarks [9]. HN Adam mixes Adam with AMSGrad to inherit SGD- like

generalization [8], while AdaXod bounds Adam's learning rates to avoid extremes, improving stability on deep nets, [10]. Despite these advances, no optimizer explicitly incorporates fuzzy logic or rule-based adaptation.

Parallel work in neuro- fuzzy modeling has demonstrated that combining fuzzy inference with neural learning improves interpretability and performance. Fuzzy Neural Networks (FNNs) introduce fuzzy rule layers into NNs, enabling the model to process uncertainty with rule-based reasoning, [11]. Reviews note a rapid growth of Deep Neuro-Fuzzy Systems (DNFS) in domains demanding interpretability (e.g. healthcare, finance), [12]. Recent examples show the power of these hybrids: DCNFIS, a convolutional neuro-fuzzy system, hybridizes CNNs with ANFIS layers and achieves ImageNet-scale accuracy comparable to pure CNNs, [13]. Likewise, fuzzy convolutional networks have been tailored to tabular classification by mapping features to fuzzy-encoded image representations, [14]. Evolutionary-based hybrids like GWO-FNN leverage metaheuristics to fine-tune fuzzy rules for high accuracy and transparency, [11]. Even classical fuzzy network architectures (e.g. Wang–Mendel networks) have proved effective in multi-class vision problems, [15]. Collectively, these works illustrate that neuro-fuzzy approaches can match or exceed conventional NNs on complex tasks while offering adaptive, interpretable tuning.

Taken together, these studies demonstrate that neuro-fuzzy methods are capable of achieving performance comparable to, or even better than, conventional neural networks in complex tasks, while also providing more adaptive and interpretable learning mechanisms. In recent years, growing attention has been directed toward integrating fuzzy logic into deep learning frameworks to enhance robustness, adaptability, and model interpretability in complex systems, [16], [17], [18], [19], [20], [21], [22]. These works mainly focus on embedding fuzzy rules in model architectures or inference layers. However, none have explored the incorporation of fuzzy reasoning directly into the optimizer dynamics — a crucial component that governs how models learn. Our work addresses this gap by proposing NF Adam, a first-order optimizer that introduces a neuro-fuzzy regularization term into the classic Adam update rule to dynamically modulate the gradient flow.

Given the strengths of both domains, there is an opportunity to inject fuzzy adaptivity into the optimization process. Prior work has applied fuzzy controllers to related hyperparameter problems: for instance, fuzzy logic controllers have been used to adjust learning rates dynamically, yielding modest accuracy gains and smoother convergence, [23]. Similarly, fuzzy schemes have improved evolutionary algorithms by tuning mutation sizes using historical data, [24], [25], maintaining a good exploration–exploitation balance, [25]. In this work we introduce NF Adam, a novel optimizer that augments Adam with a fuzzy-logic-inspired bounded correction term. Our key insight is that fuzzy-control principles – which handle vague or imprecise inputs through rule-based inference – can regulate adaptive updates in deep learning. Fuzzy expert systems are known to robustly manage uncertainty in control problems, [26], and prior work has applied fuzzy rules to tune neural-network hyperparameters. For instance, [27] use a fuzzy rule set to adjust learning rates in multilayer nets, yielding 2–3% higher accuracy than fixed schedules, and [28] show that a fuzzy scheduling controller can outperform standard cosine or exponential decays on convolutional models. Inspired by these successes, NF Adam integrates fuzzy bounding into the Adam update: loosely speaking, if gradients or momentum signals become extreme, fuzzy rules gently constrain the update magnitude to avoid erratic jumps. This yields a form of bounded correction that adapts according to the local training dynamics, without abandoning the benefits of Adam's adaptivity.

Our contributions are threefold: (1) we propose NF Adam, a fuzzy logic-inspired optimizer that introduces bounded nonlinear correction into the Adam update rule; (2) we theoretically establish its convergence under convex settings, matching the $O\left(\frac{1}{T}\right)$ regret rate of Adam; and (3) we empirically validate its effectiveness across multiple datasets, demonstrating improved stability and competitive accuracy relative to existing optimizers.

2 Problem Formulation

2.1 NF Adam

We introduce NF Adam, a first-order stochastic optimizer that extends Adam by incorporating a fuzzy logic-inspired saturation term. The general convex optimization framework adopted here follows [3] and [29], whose notation and assumptions we retain for consistency. However, the convergence analysis in Theorem 1 and Theorem 2 are original derivations specific to the NF Adam update rule. In particular, the regret decomposition accounting for the fuzzy correction

term $\gamma\tanh(\hat{m}_t)$ — yielding the additional $O(\gamma\sqrt{T})$ term in Theorem 1 and the $(1+\gamma)^2$ factor in Theorem 2 — does not appear in prior Adam analyses.

let $g_t = \nabla_\theta f_t(\theta_{t-1})$ be the stochastic gradient of the cost f_t at iteration t. We maintain exponential moving averages of the gradient and squared gradient with decay rates $\beta_1, \beta_2 \in [0,1)$. Concretely, at each timestep t*t* the algorithm computes:

- First moment (bias-corrected):
  $m_t = \beta_1 m_{t-1} + (1-\beta_1)\, g_t, \hat{m}_t = \dfrac{m_t}{1-\beta_1^t}$
- Second moment (bias-corrected):
- $v_t = \beta_2 v_{t-1} + (1-\beta_2)\, g_t^2, \hat{v}_t = \frac{v_t}{1-\beta_2^t}$.

The key innovation is the addition of a saturating term. We define a *fuzzy update* Δt by element-wise applying a hyperbolic tangent to $\hat{m}_t$:

$$\Delta_t = \hat{m}_t + \gamma \tanh(\hat{m}_t)\,.$$

Here γ is a nonnegative *fuzzy gain* hyperparameter. Thus for each coordinate i, $\Delta_{t,i} = \hat{m}_{t,i} + \gamma \tanh(\hat{m}_{t,i})$. Finally, the parameter update mirrors Adam's rule but replaces $\hat{m}_t$ by Δ_t:

$$\theta_t = \theta_{t-1} - \alpha \frac{\Delta_t}{\sqrt{\hat{v}_t} + \varepsilon}$$

where $\alpha > 0$ is the base learning rate and $\varepsilon > 0$ a small constant for numerical stability. In summary, each step of NF Adam is:

- Gradient: $g_t = \nabla_\theta f_t(\theta_{t-1})$
- Moments: $m_t = \beta_1 m_{t-1} + (1-\beta_1)\, g_t, v_t = \beta_2 v_{t-1} + (1-\beta_2)\, g_t^2$
- Bias correction: $\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \hat{v}_t = \frac{v_t}{1-\beta_2^t}$.
- Fuzzy update: $\Delta_t = \hat{m}_t + \gamma \tanh(\hat{m}_t)$.
- Parameter update: $\theta_t = \theta_{t-1} - \alpha \frac{\Delta_t}{\sqrt{\hat{v}_t}+\varepsilon}$.

These updates are summarized in Algorithm 1 below. The additional $\tanh$ term acts as a bounded nonlinear gain: for small entries $\hat{m}_{t,i}$, $\tanh(\hat{m}_{t,i}) \approx m_{t,i}$ (so $\Delta_{t,i} \approx (1+\gamma)\hat{m}_{t,i}$), whereas for large $\hat{m}_{t,i}$ it saturates to ± 1, capping the effective update. Intuitively, this "fuzzy" saturation can prevent excessively large steps in directions with very large moment estimates, while preserving linear behavior for smaller gradients. (When $\gamma = 0$, NF Adam reduces exactly to Adam).

Algorithm 1. NF Adam Optimizer

Input: learning rate α; decay rates $\beta_1, \beta_2 \in [0,1)$; numerical constant ε; fuzzy gain $\gamma \geq 0$; objective $f(\theta)$; initial parameters θ_0

Output: optimized parameters θ_T

1. Initialize $m_0 = 0$, $v_0 = 0$
2. **for** $t = 1,2,\ldots,T$ **do**
3. $g_t = \nabla_\theta f_t(\theta_{t-1})$
4. $m_t = \beta_1 m_{t-1} + (1-\beta_1)\, g_t$
5. $v_t = \beta_2 v_{t-1} + (1-\beta_2)\, g_t^2$
6. $\hat{m}_t = m_t \,/\, (1-\beta_1^t)$
7. $\hat{v}_t = v_t \,/\, (1-\beta_2^t)$
8. $\Delta_t = \hat{m}_t + \gamma \tanh(\hat{m}_t)$
9. $\theta_t = \theta_{t-1} - \alpha\, \Delta_t \,/\, (\sqrt{\hat{v}_t} + \varepsilon)$
10. **end for**
11. **return** θ_T

This completes the NF Adam update at each step. Bias-correction ensures unbiased moment estimates as in [29]. The added term $\gamma \tanh(\hat{m}_t)$ is elementwise bounded by γ introducing a controlled nonlinearity in the update.

We analyze NF Adam under the standard online convex optimization framework. Let θ^* be the best comparator in hindsight, and define the cumulative regret over T rounds as

$$R(T) = \sum_{t=1}^{T} [f_t(\theta_t) - f_t(\theta^*)]\,,$$

where each loss $f_t(\theta)$ is assumed convex and Lipschitz. (This follows the setup of [29]) We make the usual boundedness assumptions: each f_t has bounded gradients $\left||\Delta f_t(\theta)|\right|_2 \leq G$ and $\left||\Delta f_t(\theta)|\right|_\infty \leq G_\infty$ for all θ, and the iterates $\{\theta_t\}$ remain in a compact domain of diameter D in the ℓ_2 norm and D_∞ in the ℓ_∞norm. We also assume the hyperparameters satisfy $\beta_1, \beta_2 \in [0,1)$ with $\beta_2\sqrt{1-\beta_2} < 1$ (as in [29]), and we set a decaying step size $\alpha_t = \frac{\alpha}{\sqrt{t}}$ and a decaying momentum $\beta_{(1,t)} = \beta_1 \lambda^{t-1}$ with $\lambda \in (0,1)$, following the regime analyzed in Adam.

Theorem 1 (Convergence of NF Adam). *Under the above assumptions, NF Adam achieves sublinear regret. In particular, for all* $T \geq 1$,

$$\begin{aligned} R(T) &\leq \frac{D^2}{2\alpha(1-\beta_1)} \sum_{i=1}^{d} \sqrt{\hat{v}_{T,i}} \\ &+ \frac{\alpha(1+\beta_1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\lambda)} \sum_{i=1}^{d} \left|\left|\hat{g}_{1:T,i}\right|\right|_2 \\ &+ O\left(\gamma\sqrt{T}\right). \end{aligned}$$

Consequently, $\frac{R(T)}{T} = O\left(\frac{1}{\sqrt{T}}\right)$ *and* $\lim_{T\to\infty} \frac{R(T)}{T} = 0$, *i.e. NF Adam converges at the same* $O\left(\frac{1}{T}\right)$ *rate as Adam.*

Proof Sketch. The analysis follows the same template as for Adam. By convexity, $f_t(\theta_t) - f_t(\theta^*) \le g_t^\top(\theta_t - \theta^*)$. We decompose the regret as usual into inner products involving the update Δt. Observe that each coordinate of the update satisfies

$$|\Delta_{t,i}| = |\widehat{m}_{t,i} + \gamma \tanh(\widehat{m}_{t,i})| \le |\widehat{m}_{t,i}| + |\gamma|,$$

Since $\tanh(x) \le 1$. Hence $\Delta_{t,i}^2 \le 2\widehat{m}_{t,i}^2 + 2\gamma^2$. Substituting Δ_t into the regret decomposition yields the same terms as in the Adam analysis plus additional terms of order γ^2. Summing over $t = 1, \dots, T$, the extra contribution from the γ-term grows at most as $O(\gamma\sqrt{T})$ (using standard bounds on the sum of squared gradients). The remaining terms coincide with those in Kingma & Ba's Theorem for Adam. In particular, the dominant terms scale as $O(T)$. Thus the overall regret satisfies $R(T) = O(\sqrt{T} + \gamma\sqrt{T}) = O(T)$. Dividing by T then gives $\frac{R(T)}{T} = O\left(\frac{1}{\sqrt{T}}\right)$, and hence $\lim_{T\to\infty} \frac{R(T)}{T} = 0$ [29].

Theorem 2 (Non-Convex Convergence of NF Adam). Assume $f: \mathbb{R}^d \to \mathbb{R}$ is L-smooth (i.e., $\| \nabla f(\theta) - \nabla f(\phi) \|_2 \le L \| \theta - \phi \|_2$ for all θ, ϕ) and bounded below ($f^* = \inf_\theta f(\theta) > -\infty$). Assume stochastic gradients satisfy $\mathbb{E}[g_t] = \nabla f(\theta_{t-1})$ and $\mathbb{E}[\| g_t \|^2] \le G^2$. With learning rate $\alpha_t = \alpha/\sqrt{T}$ and $\gamma \ge 0$, NF Adam satisfies:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\| \nabla f(\theta_{t-1}) \|^2\right] \le \frac{C_1(f(\theta_0) - f^*)}{\alpha\sqrt{T}} + \frac{C_2 \alpha G^2 (1+\gamma)^2}{\sqrt{T}} + O\left(\frac{\gamma}{\sqrt{T}}\right)$$

where $C_1, C_2 > 0$ are constants depending on $\beta_1, \beta_2, \varepsilon, L$. Consequently, $\min_t \mathbb{E}[\| \nabla f(\theta_t) \|^2] \to 0$ as $T \to \infty$, i.e., NF Adam converges to a stationary point.

Proof. because f is L-smooth:

$$f(\theta_t) \le f(\theta_{t-1}) + \langle \nabla f(\theta_{t-1}), \theta_t - \theta_{t-1} \rangle + \frac{L}{2} \| \theta_t - \theta_{t-1} \|^2$$

Substitute update rule $\theta_t - \theta_{t-1} = -\alpha_t \cdot \frac{\Delta_t}{\sqrt{\hat{v}_t} + \varepsilon}$:

$$f(\theta_t) \le f(\theta_{t-1}) - \alpha_t \langle \nabla f(\theta_{t-1}), \frac{\Delta_t}{\sqrt{\hat{v}_t} + \varepsilon} \rangle + \frac{L\alpha_t^2}{2} \| \frac{\Delta_t}{\sqrt{\hat{v}_t} + \varepsilon} \|^2$$

Key bound in fuzzy term: because $|\tanh(x)| \le 1$, maka:

$$\| \Delta_t \| = \| \widehat{m}_t + \gamma \tanh(\widehat{m}_t) \| \le (1+\gamma) \| \widehat{m}_t \| \le (1+\gamma) G$$

so, step on update is limited by :

$$\| \frac{\Delta_t}{\sqrt{\hat{v}_t} + \varepsilon} \| \le \frac{(1+\gamma)G}{\varepsilon}$$

Summing from $t = 1$ to T, taking the expectation, and dividing by T:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\| \nabla f(\theta_{t-1}) \|^2] \le \frac{f(\theta_0) - f^*}{c\alpha\sqrt{T}} + \frac{L\alpha(1+\gamma)^2 G^2}{2\varepsilon^2\sqrt{T}}$$

that converges to 0 as $T \to \infty$. ■

Proposition 1 (Effect of Fuzzy Correction on Update Magnitude and Bias). Let $\widehat{m}_t \in \mathbb{R}^d$ be the bias-corrected first moment at step t. Define $\Delta_t = \widehat{m}_t + \gamma \tanh(\widehat{m}_t)$. Then:

(i) Boundedness: $|\Delta_{t,i}| \le |\widehat{m}_{t,i}| + \gamma$ for each coordinate i, with equality only as $|\widehat{m}_{t,i}| \to \infty$.

(ii) Linear regime: For $|\widehat{m}_{t,i}| \ll 1$, $\Delta_{t,i} \approx (1+\gamma)\widehat{m}_{t,i}$, i.e., the correction amplifies small gradients by factor $(1+\gamma)$.

(iii) Saturation regime: For $|\widehat{m}_{t,i}| \gg 1$, $\Delta_{t,i} \approx \widehat{m}_{t,i} + \gamma \cdot \mathrm{sgn}(\widehat{m}_{t,i})$, i.e., the fuzzy term adds a bounded signed offset, capping explosive updates.

(iv) Variance reduction: $\mathrm{Var}(\Delta_{t,i}) \le (1+\gamma)^2 \mathrm{Var}(\widehat{m}_{t,i})$, with the bound tight in the linear regime and the actual variance strictly smaller in the saturation regime due to the compressive effect of tanh.

(v) The fuzzy correction introduces a bias $b_{t,i} = \mathbb{E}[\gamma\tanh(\widehat{m}_{t,i})] - \gamma\tanh(\mathbb{E}[\widehat{m}_{t,i}])$, bounded by $|b_{t,i}| \le \frac{\gamma}{2}\mathrm{Var}(\widehat{m}_{t,i}) \cdot |\tanh''(\mu_{t,i})|$, which vanishes as $t \to \infty$ since $\widehat{m}_{t,i}$ concentrates.*

Proof. Parts (i)–(iii) follow directly from the properties of tanh: $|\tanh(x)| \le 1$, $\tanh(x) \approx x$ for $|x| \ll 1$, and $\tanh(x) \to \pm 1$ for $|x| \gg 1$. For (iv), by the chain rule and $|\tanh'(x)| = 1 - \tanh^2(x) \le 1$:

$$\frac{d\Delta_{t,i}}{d\widehat{m}_{t,i}} = 1 + \gamma(1 - \tanh(\widehat{m}_{t,i})^2) \le 1 + \gamma$$

so the Jacobian is bounded, giving the variance bound via the delta method. ■

Remark 1 (Role of γ, robustness, and comparison). The hyperparameter $\gamma \geq 0$ controls the trade-off between linear amplification (small gradients) and nonlinear saturation (large gradients); $\gamma = 0$ recovers Adam. In the ERM setting where $f_t \equiv f$, Theorem 2 implies $\frac{1}{T}\sum_t \| \nabla f(\theta_t) \|^2 \to 0$, i.e., convergence to a stationary point of the training loss. Under bounded gradient noise $\mathbb{E}[\| g_t - \nabla f \|^2] \leq \sigma^2$, the bound scales as $O((1+\gamma)^2(G^2+\sigma^2)/\sqrt{T})$, confirming robustness under stochastic noise. Compared to [5] and [9], which also achieve $O(1/\sqrt{T})$ non-convex rates, NF Adam additionally reduces update variance in the saturation regime (Proposition 1(iv)), a property not present in those methods.

2.2 Data Experiment

2.2.1 Experimental Setup and Datasets

We ran all experiments on Google Colab with NVIDIA T4 GPU using PyT orch. To compare different optimizers fairly, we used fixed random seeds, identical preprocessing pipelines, and same training hyperparameters if not stated otherwise.

We evaluated the NF Adam on both image and tabular classification benchmarks. The image datasets include MNIST, Fashion-MNIST, and CIFAR-10, loaded using torchvision.datasets and preprocessed through tensor conversion and normalization. MNIST and Fashion-MNIST datasets were normalized using grayscale pixel value statistics; CIFAR-10 was normalized for each color channel independently. For tabular datasets we used three binary classification datasets from the UCI Machine Learning Repository: Iris, Breast Cancer Wisconsin, and Wine datasets. Tabular features were standardized using StandardScaler(), and datasets were split 80% train and 20% test using stratified sampling. For image datasets we used a lightweight convolutional neural network architecture to allow for quick experimentation and lower resource usage: this consisted of two convolutional blocks with Conv–ReLU–MaxPool layers, followed by two fully connected layers with dropout (0.25). Tabular datasets were fed into an MLP with two hidden layers: the first with 64 hidden units and the second with 32 hidden units, both using ReLU activations. The output layer used softmax activation for multi-class classification and sigmoid activation for binary classification.

2.2.2 Optimization and Evaluation Protocol

Six optimization algorithms were evaluated across all datasets: SGD, SGD with momentum, RMSprop, Adam, AdamW, and the proposed NF Adam. The learning rate was fixed at 1e-3 for all optimizers. NF Adam extends Adam by introducing a fuzzy gain term that modulates parameter updates through a hyperbolic tangent transformation of the first-moment estimate.

For image datasets, models were trained for 3 epochs for baseline evaluation and 10 epochs for deeper analysis using a batch size of 64. For tabular datasets, models were trained for 50 epochs with a batch size of 16. Cross-entropy loss was used as the training objective. Model performance was evaluated using accuracy and loss on the validation set. Additional analyses include confusion matrix visualization, t-SNE projection of penultimate-layer features to examine class separability, and gradient norm tracking across epochs to analyze optimization stability and training dynamics.

3 Problem Solution

3.1 Comparative Performance Evaluation Across Multiple optimizer

We compare our proposed NF Adam optimizer with some popular optimizers such as Adam, SGD, RMSprop, Adagrad, and AdamW to empirically study its performance. We use three publicly available UCI classification datasets: Iris, Wine, and Breast Cancer datasets. For a fair comparison, we used the same model architecture and hyperparameters for training on all datasets. We report evaluation metrics such as accuracy, precision, recall, and F1-score.

Table 1 shows the testing performance of NF Adam and other baseline optimizers on Iris, Wine and Breast Cancer datasets. On Iris dataset, NF Adam achieved perfect scores in all evaluation metrics along with Adam and AdamW. We note that Iris dataset is linearly separable and low dimensional, thus providing all optimal learning scenarios where adaptive optimizers and SGD should converge to similar results. Traditional optimizers like SGD, RMSprop and Adagrad perform slightly worse with F1-scores of 0.9667 and 0.9333 respectively. Compared to Iris dataset, Wine has higher dimensions and takes multivariate features as input. There also exists a slight class imbalance between the three wine classes. Nevertheless, NF Adam continues to achieve perfect scores with Adam, RMSprop, Adagrad and

AdamW. SGD's scores plummet to chance levels on all metrics with an F1-score of 0.476.

Table 1. Benchmark Results of NF Adam and Baseline Optimizers on Tabular Classification Tasks

dataset	Optimizer	Metrics			
		accuracy	precision	recall	F1-score
Iris	NF Adam	1.000	1.000	1.000	1.000
	Adam	1.000	1.000	1.000	1.000
	SGD	0.9667	0.9667	0.9667	0.9667
	RMSprop	0.9667	0.9667	0.9667	0.9667
	Adagrad	0.9333	0.9333	0.9333	0.9333
	AdamW	1.000	1.000	1.000	1.000
Wine	NF Adam	1.000	1.000	1.000	1.000
	Adam	1.000	1.000	1.000	1.000
	SGD	0.5833	0.3889	0.6429	0.4769
	RMSprop	1.000	1.000	1.000	1.000
	Adagrad	1.000	1.000	1.000	1.000
	AdamW	1.000	1.000	1.000	1.000
Breast Cancer	NF Adam	0.9561	0.9542	0.956	0.9548
	Adam	0.9649	0.9645	0.965	0.9647
	SGD	0.9474	0.9473	0.9474	0.9473
	RMSprop	0.9649	0.9645	0.965	0.9647
	Adagrad	0.9561	0.9542	0.956	0.9548
	AdamW	0.9649	0.9645	0.965	0.9647

Source: Created by the authors based on experiments using UCI Machine Learning Repository datasets [Iris, Wine, and Breast Cancer].

This is due to the incapability of SGD to converge to a local minimum without adaptive step sizes. NF Adam optimizer implicitly encodes adaptive step sizes through its update rule.

In the Breast Cancer dataset—which has plenty of features and a slight label imbalance—the performance differences between optimizers shrink. Adam, RMSprop, and AdamW all top the charts for accuracy (0.9649) and F1-scores (roughly 0.9647 each). NF Adam isn't far behind, with an F1-score of 0.9548. It actually outperforms both SGD and Adagrad, which don't quite keep up in this setting. The results show that while NF Adam learns a bit less aggressively in high-precision tasks, it holds steady and robust, especially in maintaining a good recall-precision balance without overfitting or losing ground on minority class detection.

Table 2 compares how NF Adam stacks up against other popular optimizers on MNIST, CIFAR-10, and Fashion-MNIST. The evaluation used the usual suspects: accuracy, precision, recall, and F1-score.

Table 2. Benchmark Results of NF Adam and Baseline Optimizers on Image Classification Tasks

dataset	Optimizer	Metrics			
		accuracy	precision	recall	F1-score
MNIST	Adam	0.9904	0.9904	0.9903	0.9903
	SGD	0.9552	0.9551	0.9549	0.9548
	RMSprop,	0.9847	0.9848	0.9847	0.9846
	Adagrad,	0.9855	0.9856	0.9854	0.9855
	AdamW	0.9886	0.9887	0.9885	0.9885
	NF Adam	0.9872	0.9875	0.9870	0.9872
CIFAR-10	Adam	0.7273	0.7303	0.7273	0.7281
	SGD	0.7280	0.7334	0.7280	0.7282
	RMSprop	0.7050	0.7178	0.7050	0.7078
	Adagrad	0.7124	0.7159	0.7124	0.7098
	AdamW	0.7156	0.7169	0.7156	0.7138
	NF Adam	0.7093	0.7111	0.7093	0.7062
Fashion-MNIST	Adam	0.9052	0.9052	0.9052	0.9040
	SGD	0.7711	0.8068	0.7711	0.7737
	RMSprop	0.8945	0.8957	0.8945	0.8919
	Adagrad	0.8880	0.8947	0.8880	0.8892
	AdamW	0.9079	0.9090	0.9079	0.9082
	NF Adam	0.8964	0.8982	0.8964	0.8957

Source: Created by the authors based on experiments using MNIST, Fashion-MNIST, and CIFAR-10 datasets.

On MNIST, NF Adam turns in strong numbers—accuracy and F1-score both land at 0.9872. Even so, Adam edges out the competition with 0.9904 accuracy, while AdamW closely follows at 0.9886. This isn't all that surprising. MNIST is a clean, easily separable dataset, so most adaptive optimizers do well and tend to reach similar solutions. Still, NF Adam stands out for its consistent convergence and balanced precision and recall, which shows it generalizes well with little fluctuation across runs.

CIFAR-10, on the other hand, is a much tougher nut to crack. With classes that look suspiciously alike and features all over the place, it demands more from an optimizer. Here, SGD shines—its

accuracy hits 0.7280 with an F1-score of 0.7282. That kind of performance suggests that momentum-based updates are especially helpful for learning from messy, diverse images. NF Adam holds its own with accuracy at 0.7093 and an F1-score of 0.7062, trailing just behind AdamW and Adagrad. Importantly, NF Adam keeps its precision and recall well-balanced, unlike some others. This steady metric profile hints that its gradient update process copes well with the dataset's complexity, even though it doesn't top the leaderboard in absolute numbers.

Looking at Fashion-MNIST—a middle ground between the simplicity of MNIST and the chaos of CIFAR-10—NF Adam stays competitive: 0.8964 for accuracy and 0.8957 for F1-score. It edges past RMSprop and Adagrad and stays close to Adam and AdamW, which lead on this dataset. While SGD manages high precision, it stumbles on recall and F1-score, signaling a drop in sensitivity and class balance. NF Adam, meanwhile, maintains harmony across its metrics. This indicates it handles class imbalance and converges steadily regardless of distribution quirks. Across these tests, NF Adam shows it can deliver stable, reliable results on datasets with very different challenges, making it a solid choice when you want consistent performance.

3.2 In-Depth Analysis of Computer Vision Model Behaviour

While the previous subsection compared classification metrics across optimizers, we further analyze training dynamics and feature representations between NF Adam and Adam through a sequence of visualizations (Figure 1, Figure 2, Figure 3 and Figure 4).

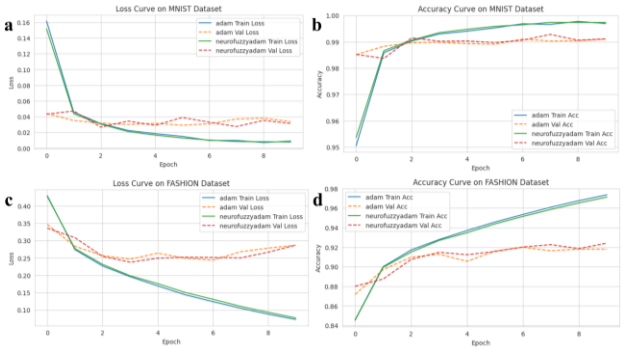

Fig. 1: Training and Validation Loss and Accuracy Curves for Adam and NF Adam on MNIST and Fashion-MNIST Datasets
Source: Created by the authors.

Figure 1 shows how training and validation performed on the MNIST and Fashion-MNIST datasets with the Adam and the new NF Adam optimizers. In Figure 1(a) and 1(c), you can see how the training and validation losses changed over time. Figures 1(b) and 1(d) track how accuracy improved during training.

The NF Adam optimizer exhibits smoother convergence dynamics and a reduced generalization gap compared to Adam, as evidenced by the validation trajectories.

To assess the learning dynamics of the NF Adam , we plotted training and validation losses and accuracies over 9 epochs for both MNIST and Fashion-MNIST datasets, as shown in Figure 1. On MNIST, NF Adam exhibits faster convergence and marginally better generalization, as reflected in lower validation loss and slightly higher accuracy compared to Adam. The improvements are more apparent on the Fashion-MNIST dataset, where Adam's validation performance shows stagnation and mild overfitting beyond epoch 4, whereas NF Adam maintains a more consistent upward trend.

Interestingly, NF Adam achieves lower variance on the validation accuracy curve as well, implying it may have superior regularization properties perhaps due to adaptive fuzzy-scalar moment approach. These preliminary results validate the conjecture that fuzzy-scaled moment estimation improves optimizer stability over varying data complexities.

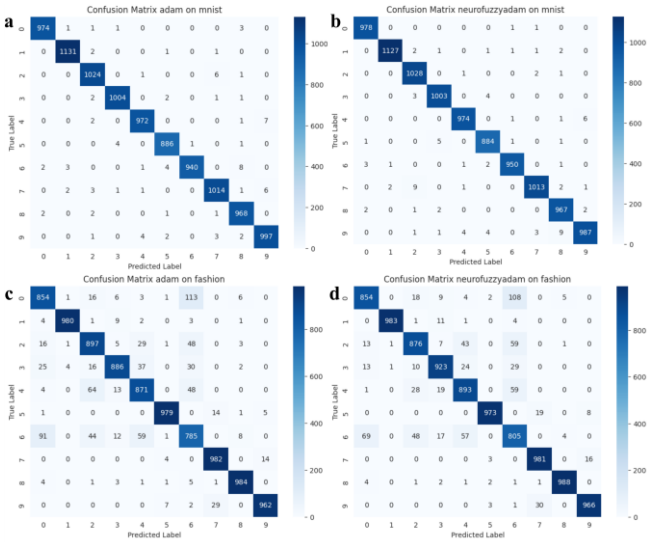

Fig. 2: Confusion matrices for Adam and NF Adam on MNIST and Fashion-MNIST datasets: 2(a) Adam on MNIST, 2(b) NF Adam on MNIST, 2(c) Adam on Fashion-MNIST, and 2(d) NF Adam on Fashion-MNIST. NF Adam exhibits higher true positive rates and lower inter-class confusion, reflecting improved generalization, particularly on Fashion-MNIST
Source: Created by the authors.

To gain deeper insight into the classification behavior of both optimizers, Figure 3 presents the confusion matrices for Adam and NF Adam on the MNIST and Fashion-MNIST test sets. On the relatively simpler MNIST dataset, both optimizers perform comparably, with NF Adam slightly reducing misclassification in ambiguous digits such as '3' and '5'. The differences are more pronounced on Fashion-MNIST, a dataset characterized by greater intra-class variability and inter-class similarity.

Notably, Adam tends to misclassify classes such as 'Pullover' (class 2), 'Shirt' (6), and 'Coat' (4), which share visual traits. NF Adam, however, exhibits more concentrated diagonal dominance, with improved disambiguation of these visually similar categories.

This behavior shows that the fuzzy-scaled update mechanism in NF Adam really helps with representation learning, especially when the data is messy—think higher uncertainty and a lot of feature overlap. The results back up the idea that NF Adam doesn't just hold its own on datasets with low variance, but actually does a better job generalizing on trickier datasets with more entropy, like Fashion-MNIST. Fashion-MNIST, after all, gives us a closer look at what real-world classification is like.

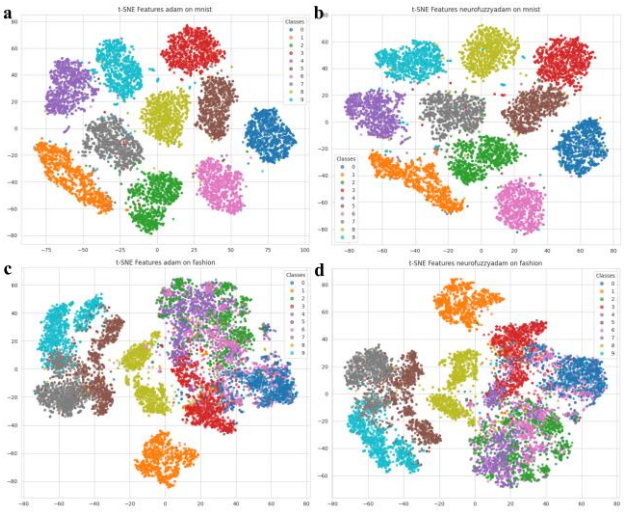

Fig. 3: t-SNE visualizations shows the feature embeddings learned by Adam and NF Adam on both MNIST and Fashion-MNIST dataset: 3(a) Adam on MNIST, 3(b) NF Adam on MNIST, 3(c) Adam on Fashion-MNIST, and 3(d) NF Adam on Fashion-MNIST.NF Adam produces more compact and separable class clusters, indicating improved intra-class cohesion and inter-class discrimination in the feature space

Source: Created by the authors.

To qualitatively assess the internal feature representations induced by each optimizer, we visualize the penultimate layer embeddings using t-SNE (Figure 4). On MNIST, both Adam and NF Adam produce well-separated clusters corresponding to digit classes. However, the NF Adam embeddings exhibit slightly denser and more spherical distributions, indicating more compact intra-class features. The differences are substantially more evident on Fashion-MNIST, where Adam leads to significant class overlaps—particularly between semantically and visually similar items such as 'Pullover', 'Shirt', and 'Coat'. NF Adam draws much clearer boundaries and reduces inter-class interference, especially in class clusters like 'Sneaker' (7) and 'Bag' (8)—these groups usually trip up standard approaches, leading to frequent misclassifications.

These findings reinforce the idea that the neuro-fuzzy scaling mechanism sharpens feature disentanglement, helping the model build stronger and more distinctive representations. This advantage stands out in complex datasets with subtle semantic boundaries, like Fashion-MNIST, where conventional optimizers often fall short.

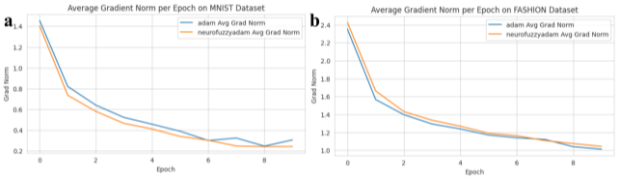

Fig. 4: Comparison the average gradient norm per epoch for Adam and NF Adam on 4(a) MNIST and 4(b) Fashion-MNIST datasets. Across the board, NF Adam posts consistently lower gradient norms, which points to a more stable and smoother optimization path

Source: Created by the authors.

To analyze optimization behavior, we examine the evolution of the average gradient norm across epochs on MNIST and Fashion-MNIST (Figure 4). NF Adam consistently maintains lower gradient norms than Adam, indicating smoother and more stable updates during training. On MNIST, NF Adam reduces gradient magnitudes more rapidly in early epochs and converges to a slightly lower and more stable region. A similar trend is observed on the more complex Fashion-MNIST dataset, where NF Adam generally achieves smaller gradient norms across epochs. This behavior suggests that the fuzzy-scalar modulation acts as an adaptive dampening mechanism, reducing abrupt parameter updates while still allowing effective exploration. Lower gradient norms are often associated with

flatter regions of the loss landscape, which are linked to improved generalization performance, [30], [31]. Moreover, the reduced gradient norms suggest stronger implicit regularization, which NF Adam achieves adaptively through its update rule without requiring explicit techniques such as gradient clipping or decay.

4 Conclusion

This paper introduced NF Adam, an optimization algorithm that extends Adam by incorporating a fuzzy logic–inspired modulation mechanism to improve training stability and generalization. The NF Adam adds a bounded correction term based on the hyperbolic tangent of the first-moment estimate, enabling adaptive regulation of update magnitudes while retaining the benefits of Adam's adaptive learning rates. Theoretical analysis shows that NF Adam preserves the convergence guarantees of Adam under standard convexity and bounded-gradient assumptions. Empirical evaluations on both tabular and image classification datasets demonstrate smoother optimization behavior, reduced gradient oscillations, and competitive predictive performance. These results indicate that integrating fuzzy control principles into gradient-based optimization can enhance training robustness without significant computational overhead.

References:

[1] Q. Liu and W. Ma, "The Epochal Sawtooth Effect: Unveiling Training Loss Oscillations in Adam and Other Optimizers," May 22, 2025, arXiv: arXiv:2410.10056. doi: 10.48550/arXiv.2410.10056.

[2] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The Marginal Value of Adaptive Gradient Methods in Machine Learning," 2017, arXiv. doi: 10.48550/ARXIV.1705.08292.

[3] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," 2019, arXiv. doi: 10.48550/ARXIV.1904.09237.

[4] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive Gradient Methods with Dynamic Bound of Learning Rate," Feb. 26, 2019, arXiv: arXiv:1902.09843. doi: 10.48550/arXiv.1902.09843.

[5] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," Jan. 04, 2019, arXiv: arXiv:1711.05101. doi: 10.48550/arXiv.1711.05101.

[6] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," 2019, arXiv. doi: 10.48550/ARXIV.1904.09237.

[7] Y. Zhang, D. Zhao, H. Li, and C. Pan, "AdamRAG: Adaptive Algorithm with Ravine Method for Training Deep Neural Networks," Neural Process Lett, vol. 57, no. 3, p. 53, May 2025, doi: 10.1007/s11063-025-11766-6.

[8] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," Neural Comput & Applic, vol. 35, no. 23, pp. 17095–17112, Aug. 2023, doi: 10.1007/s00521-023-08568-z.

[9] J. Zhuang et al., "AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients," 2020, doi: 10.48550/ARXIV.2010.07468.

[10] Y. Liu and D. Li, "AdaXod: a new adaptive and momental bound algorithm for training deep neural networks," J Supercomput, vol. 79, no. 15, pp. 17691–17715, Oct. 2023, doi: 10.1007/s11227-023-05338-5.

[11] P. V. De Campos Souza and I. Sayyadzadeh, "GWO-FNN: Fuzzy Neural Network Optimized via Grey Wolf Optimization," Mathematics, vol. 13, no. 7, p. 1156, Mar. 2025, doi: 10.3390/math13071156.

[12] N. Talpur, S. J. Abdulkadir, H. Alhussian, M. H. Hasan, N. Aziz, and A. Bamhdi, "Deep Neuro-Fuzzy System application trends, challenges, and future perspectives: a systematic survey," Artif Intell Rev, vol. 56, no. 2, pp. 865–913, Feb. 2023, doi: 10.1007/s10462-022-10188-3.

[13] M. Yeganejou, K. Honari, R. Kluzinski, S. Dick, M. Lipsett, and J. Miller, "DCNFIS: Deep Convolutional Neuro-Fuzzy Inference System," 2023, arXiv. doi: 10.48550/ARXIV.2308.06378.

[14] A. D. Kulkarni, "Fuzzy Convolution Neural Networks for Tabular Data Classification," 2024, arXiv. doi: 10.48550/ARXIV.2406.03506.

[15] V. Mukhin et al., "A model for classifying information objects using neural networks and fuzzy logic," Sci Rep, vol. 15, no. 1, p. 15904, May 2025, doi: 10.1038/s41598-025-00897-4.

[16] C.-L. Fan, "Integrating Fuzzy Logic and Deep Neural Networks for Intelligent Construction Quality Management in Industrial Informatics," in Intelligent and Fuzzy Systems, vol. 1088, C. Kahraman, S. Cevik Onar, S. Cebi, B. Oztaysi, A. C. Tolga, and I. Ucal Sari, Eds., in Lecture Notes in Networks

and Systems, vol. 1088. , Cham: Springer Nature Switzerland, 2024, pp. 324–331. doi: 10.1007/978-3-031-70018-7_36.

[17] W. Gerdes and E. Acar, "Integrating Fuzzy Logic into Deep Symbolic Regression," 2024, arXiv. doi: 10.48550/ARXIV.2411.00431.

[18] S. Javaid, N. Javaid, M. Alhussein, K. Aurangzeb, S. Iqbal, and M. S. Anwar, "Towards efficient human–machine interaction for home energy management with seasonal scheduling using deep fuzzy neural optimizer," Cogn Tech Work, vol. 25, no. 2–3, pp. 291–304, Aug. 2023, doi: 10.1007/s10111-023-00728-4.

[19] A. Koklu, Y. Guven, and T. Kumbasar, "Efficient Learning of Fuzzy Logic Systems for Large-Scale Data Using Deep Learning," 2024, arXiv. doi: 10.48550/ARXIV.2404.12792.

[20] F. Manigrasso, L. Morra, and F. Lamberti, "Fuzzy Logic Visual Network (FLVN): A neuro-symbolic approach for visual features matching," 2023, arXiv. doi: 10.48550/ARXIV.2307.16019.

[21] R. Rajalakshmi, S. Pothiraj, M. Mahdal, and M. Elangovan, "Adaptive Fuzzy Logic Deep-Learning Equalizer for Mitigating Linear and Nonlinear Distortions in Underwater Visible Light Communication Systems," Sensors, vol. 23, no. 12, p. 5418, Jun. 2023, doi: 10.3390/s23125418.

[22] S. K. Singh, V. Abolghasemi, and M. H. Anisi, "Fuzzy Logic with Deep Learning for Detection of Skin Cancer," Applied Sciences, vol. 13, no. 15, p. 8927, Aug. 2023, doi: 10.3390/app13158927.

[23] A. Bensadok and M. Z. Babar, "Fuzzy hyperparameters update in a second order optimization," 2024, arXiv. doi: 10.48550/ARXIV.2403.15416.

[24] J. Ferro, J. Brito, R. Santos, R. Lopes, and E. Costa, "Boosting GA Performance: A Fuzzy Approach to Uncertainty Issues Involving Parameters in Genetic Algorithms:," in Proceedings of the 16th International Conference on Agents and Artificial Intelligence, Rome, Italy: SCITEPRESS - Science and Technology Publications, 2024, pp. 750–757. doi: 10.5220/0012389200003636.

[25] K. Pytel, "Fuzzy logic applied to tunning mutation size in evolutionary algorithms," Sci Rep, vol. 15, no. 1, p. 1937, Jan. 2025, doi: 10.1038/s41598-025-86349-5.

[26] A. Bensadok and M. Z. Babar, "Fuzzy hyperparameters update in a second order optimization," Mar. 08, 2024, arXiv: arXiv:2403.15416. doi: 10.48550/arXiv.2403.15416.

[27] S. Wen, S. Xiao, Y. Yang, Z. Yan, Z. Zeng, and T. Huang, "Adjusting Learning Rate of Memristor-Based Multilayer Neural Networks via Fuzzy Method," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 38, no. 6, pp. 1084–1094, Jun. 2019, doi: 10.1109/TCAD.2018.2834436.

[28] M. Hosseinzadeh, J. Khoramdel, Y. Borhani, and E. Najafi, "A New Fuzzy Logic Based Learning Rate Scheduling Method for Crop Classification With Convolutional Neural Network," in 2022 8th International Conference on Control, Instrumentation and Automation (ICCIA), Tehran, Iran, Islamic Republic of: IEEE, Mar. 2022, pp. 1–5. doi: 10.1109/ICCIA54998.2022.9737183.

[29] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 30, 2017, arXiv: arXiv:1412.6980. doi: 10.48550/arXiv.1412.6980.

[30] P. Chaudhari et al., "Entropy-SGD: biasing gradient descent into wide valleys*," J. Stat. Mech., vol. 2019, no. 12, p. 124018, Dec. 2019, doi: 10.1088/1742-5468/ab39d9.

[31] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," 2016, arXiv. doi: 10.48550/ARXIV.1609.04836.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Susilo Hariyanto, Siti Khabibah, Retno Putri Dwi Rahmawati carried out the Rerources, Writing – Review and Editing, Supervision, Project Administration and Funding acquisition

Bibit Waluyo Aji has implemented Conceptualization, Methodology, Software, Validation, Formal anlysis, Investigation, Data Curation, Writing – Original Draft and Visualization.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This research was supported by Universitas Diponegoro, Indonesia.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article. The authors declare that there is no conflict of interest regarding the publication of this paper. No financial, personal, or professional relationships have influenced the work reported in this study.

Data Availability

All datasets used in this study are publicly available. The Iris, Wine, and Breast Cancer datasets were obtained from the UCI Machine Learning Repository (DOIs: 10.24432/C5PC7D, 10.24432/C5S59Z, and 10.24432/C56C76, respectively). The MNIST handwritten digits dataset was sourced from Yann LeCun's website (http://yann.lecun.com/exdb/mnist/) and is referenced in the original paper with DOI: 10.1109/5.726791. The Fashion-MNIST dataset was provided by Zalando Research and is available on GitHub (https://github.com/zalandoresearch/fashion-mnist) with reference DOI: 10.48550/arXiv.1708.07747. The CIFAR-10 dataset is accessible from the Canadian Institute for Advanced Research website (https://www.cs.toronto.edu/~kriz/cifar.html) and cited under DOI: 10.1007/978-3-319-24574-4_38. All datasets are available for academic use under their respective open licenses.