Improved Adam Optimizer with Warm-Up Strategy and Hyperbolic Tangent Function for Employee Turnover Prediction Using Multilayer Perceptron (MLP)

*Archie O. Pachica
Technological Institute of the Philippines
Quezon City, Philippines
qapachica01@tip.edu.ph
University of Science and Technology of Southern Philippines
Cagayan de Oro City, Philippines
archie.pachica@ustp.edu.ph

Arnel C. Fajardo
Isabela State University
Cauayan, Isabela, Philippines
acfajardo2011@gmail.com

Ruji P. Medina
Technological Institute of the Philippines
Quezon City, Philippines
ruji.medina@tip.edu.ph

***Abstract*— Employee turnover poses significant challenges for organizations, affecting both operational efficiency and financial stability. Accurately predicting employee turnover can help organizations implement proactive measures to retain valuable talent. This research investigates an enhancement to the Adam optimizer by integrating a warm-up strategy with the Hyperbolic Tangent (tanh) function, aimed at reducing the number of iterations or epochs needed to train Multilayer Perceptron (MLP) models for predicting employee turnover. The Adam optimizer, known for its adaptive learning rate and computational efficiency, can require extensive training periods to reach optimal performance. Incorporating a tanh-based warm-up phase improves the stability and convergence speed of the training process. This study shows that the proposed approach not only reduces training duration but also enhances the predictive accuracy of MLP models. The results indicate that the Adam optimizer with the tanh-based warm-up strategy significantly outperforms other optimizers, including Nadam, AdaMax, and AdamP, achieving the highest accuracy (95%) and it required the fewest iterations to converge (50 epochs. These findings suggest that this novel optimization technique can be valuable for organizations seeking efficient and accurate employee turnover prediction models. Future research should explore the applicability of this approach across different datasets and neural network architectures to further validate these findings.**

Keywords*— *Employee Turnover; ADAM; Warm-Up; TanH; MLP

I. INTRODUCTION

Employee turnover poses significant challenges for organizations, impacting productivity, morale, and financial stability [1]. Predicting employee turnover accurately enables companies to implement effective retention strategies, thereby mitigating these adverse effects. Machine learning models, particularly Multilayer Perceptrons (MLPs), have demonstrated their efficacy in modeling complex patterns and predicting turnover with high accuracy [2] [3]. The Adam optimizer [4], is a widely adopted optimization algorithm in training neural networks due to its adaptive learning rate and computational efficiency. However, Adam requires a large number of iterations or epochs to converge, which can be computationally expensive and time-consuming [5]. To address this limitation, warm-up strategies have been proposed to enhance the optimizer's performance. A warm-up strategy involves gradually increasing the learning rate at the beginning of the training process. This technique helps in stabilizing the model training, leading to better convergence properties. One promising approach is the Hyperbolic Tangent (tanh) function induction during the warm-up phase. The tanh function, with its smoothing and bounded properties, can facilitate a more controlled and stable learning process, potentially reducing the number of iterations required for convergence [6]. This research focuses on improving the Adam optimizer by integrating a Tanh-based warm-up strategy to enhance the performance of MLPs in predicting employee turnover. The primary objective is to reduce the number of iterations or epochs needed for training, thereby increasing the efficiency and accuracy of the predictive model. By leveraging the strengths of both the Adam optimizer and the Tanh function, this study aims to provide a novel approach to developing more effective employee turnover prediction models.

II. REVIEW OF RELATED LITERATURE

Multilayer Perceptron (MLP) neural networks have emerged as a powerful tool in predictive analytics, particularly for forecasting employee turnover, which is crucial for organizational management. Recent studies highlight MLP's effectiveness in modeling complex employee data relationships to predict turnover accurately, such as the development of the Turnover Influence-based Neural Network (TINN) integrating organizational social networks with turnover similarity networks to enhance turnover prediction through dynamic modeling of social influences [7]. The Adam optimizer, proposed by Kingma and Ba in 2015, is now essential in training neural networks due to its adaptive learning rate and moment estimation capabilities, combining advantages from AdaGrad and RMSProp, making it effective for large-scale machine learning problems [4]. However, Adam can sometimes converge slowly, necessitating improvements like warm-up strategies, which involve gradually increasing the learning rate at the beginning of training to stabilize the process and accelerate convergence [8]. Activation functions are crucial in neural networks for introducing non-linearity and enabling complex pattern modeling. The hyperbolic tangent (tanh) function, which outputs values between -1 and 1, is used to improve convergence and reduce training epochs. Compared to ReLU, tanh can offer better convergence properties in specific contexts. Studies have explored tanh's applications, including a novel CMOS-based design for reducing power dissipation and area usage in memristive-based neuromorphic architectures [9], high-accuracy hardware implementation for efficient ANN performance [10], and enhancing Hopfield networks for logic programming [11]. Machine learning models for predicting employee turnover have become sophisticated, providing vital insights for organizations. Models based on k-nearest neighbors (KNN) and random forests (RF) demonstrated superior performance in terms of accuracy and precision, particularly with an IBM dataset encompassing critical employee features [12]. Despite these advances, gaps remain in neural networks and machine learning for turnover prediction, such as underexplored broader social network metrics in MLP networks, the need for improved convergence techniques for Adam, innovative warm-up strategies, and comparative analysis of activation functions like tanh. Addressing these gaps can refine MLP models, optimization algorithms, and machine learning approaches, enhancing predictive accuracy and operational efficiency in managing organizational turnover

III. MATERIALS AND METHODS

A. *Data Collection*

The data utilized in the simulation is the IBM HR Employee Attrition dataset, which is publicly accessible on Kaggle. This dataset contains 1,470 records and includes a total of 35 variables, with 1 dependent variable (Attrition) and 34 independent variables. Among the records, 1,233 are current employees from the “No” attrition group, while the remaining 237 are former employees from the “Yes” attrition group, as depicted in Figure 1.

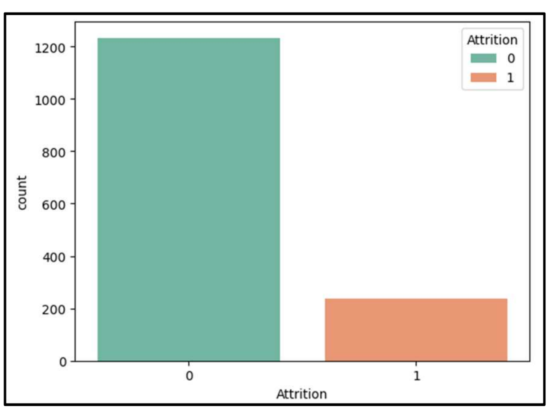

Figure 1: No. of employees who stayed and left

1) *Features of the Dataset*

The dataset includes several employee and job-related features such as EmployeeID, Age, Attrition, BusinessTravel, Department, DistanceFromHome, Education, JobRole, MaritalStatus, MonthlyIncome, PercentSalaryHike,TotalWorkingYears,YearsAtCompany, WorkLifeBalance, YearsSinceLastPromotion, and YearsWithCurrManager.

2) *Preprocessing Steps*

a. Data Cleaning. Irrelevant columns are removed from the dataset such as "EmployeeCount", "EmployeeNumber", "Over18", and "StandardHours". Consistency in categorical variables is ensured.
b. Encoding Categorical Variables. Categorical variables are converted to a numeric format using label encoding, making them suitable for machine learning algorithms.
c. Handling Imbalanced Data. The 'Attrition' target variable is significantly imbalanced, a technique like SMOTE is applied to balance the classes before modeling.
d. Splitting the Dataset. The dataset is divided into training and testing sets to evaluate the performance of the machine learning models. A split ratio is 80:20 for training and testing, respectively.

B. *Model Architecture*

The architecture of an MLP with an improved ADAM optimizer is shown in Figure 2. It represents a sophisticated approach in the field of neural networks, particularly for tasks like predicting employee turnover. An MLP is a type of feedforward artificial neural network that consists of multiple layers of nodes, each typically employing a nonlinear activation function. The basic structure includes an input layer, one or more hidden layers, and an output layer.

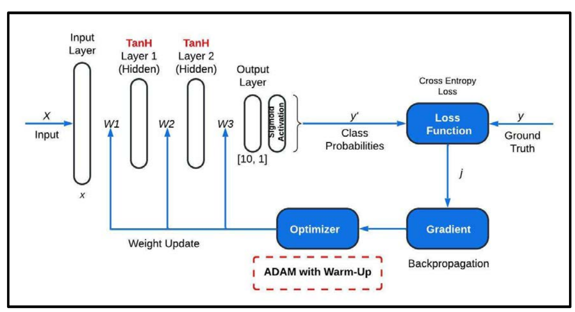

Figure 2: MLP Architecture with Improved Adam

The flow of the Multilayer Perceptron (MLP) architecture using the improved ADAM optimizer starts with the input layer, where the input data (x) is fed into the network. This data is then passed through multiple hidden layers (Layer 1 and Layer 2), each consisting of nodes or neurons that apply activation functions such as hyperbolic tangent function (tanh) to the incoming data and propagate the transformed data forward. The weights ($W1$,$W2$,$W3$) associated with each layer are adjusted based on the optimizer's updates. After processing through the hidden layers, the data reaches the output layer that uses the sigmoid activation function, which produces the output (y^). This output is then compared to the ground truth (y) using a loss function, typically cross-entropy loss, to calculate the loss (J). The gradient of the loss with respect to the network parameters is computed through backpropagation. The improved ADAM optimizer then uses these gradients to update the weights of the network. The optimizer incorporates a warm-up strategy, which gradually increases the learning rate at the beginning of the training process to stabilize updates and prevent large, destabilizing changes in the weights. This approach ensures smoother and more effective training, leading to better convergence and improved model performance. The cycle of forward propagation, loss calculation, backpropagation, and weight updates continues iteratively until the model parameters converge, resulting in a trained MLP model optimized for predicting outcomes.

C. *Optimization Techniques*

1) Modified Adam Algorithm with Warm-Up

The Adam optimizer is modified to include a warm-up phase where the learning rate gradually increases from a lower value to the initial learning rate over a specified number of iterations as shown in Figure 3.

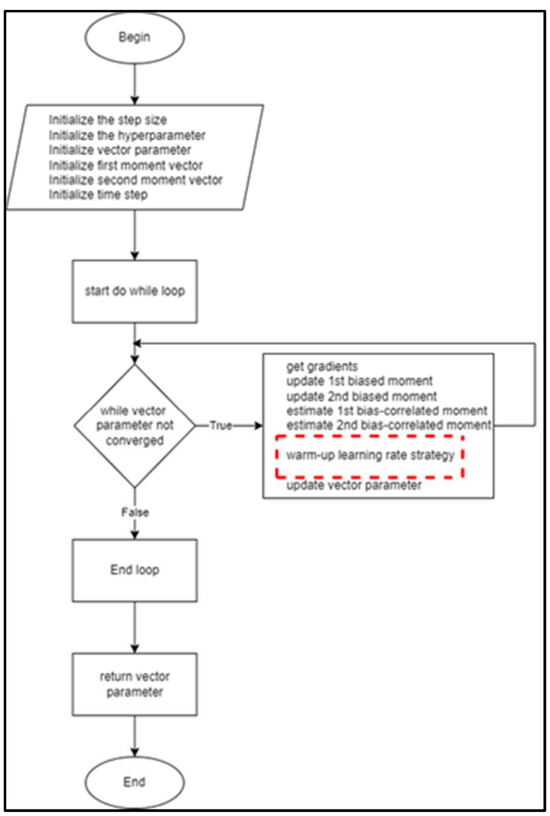

Figure 3: Modified Adam Optimizer

The flow of the ADAM optimizer algorithm with an integrated warm-up strategy begins with the initialization of the step size, hyperparameters, vector parameters, and moment vectors. The process enters a loop where the vector parameters are updated iteratively until convergence is achieved. Within each iteration, gradients are obtained, and the biased first and second moments are updated. The warm-up strategy is employed during this phase, gradually increasing the learning rate to stabilize the early training process. This approach helps prevent large, destabilizing updates to the parameters, ensuring smoother and more effective training.

2) Warm-Up Formula

This study used the warm-up strategy formula which involves gradually increasing the learning rate from a lower value to a higher one over a specified number of initial training iterations or epochs. This approach helps in stabilizing the training process, especially at the beginning when weights are randomly initialized. The formula for a warm-up is shown in Equation (1), which looks like this: Learning Rate = start + (end – start) * (current step)/(warm-up steps), where "start" is the initial learning rate, "end" is the target learning rate, "current step" is the current epoch, and "Warm-up steps" is the number of epochs over which the learning rate increases.

$$Learning\ Rate = start + (end - start) * \frac{current\ step}{warm\ up\ steps} \qquad (1)$$

3) Hyperbolic Tangent Activation

The hyperbolic tangent (tanh) activation function is used in the hidden layers during model building. With its zero-centered nature and output range from -1 to 1, tanh can accelerate learning by reducing the time needed for significant weight adjustments in the initial training phases. This can result in fewer epochs required for fast convergence.

D. *Experimental Setup*

The model was trained using the modified Adam optimizer with the warm-up strategy. Various hyperparameters, such as the learning rate, batch size, and number of epochs, were tuned to achieve optimal performance.

E. *Tools and Frameworks*

The following tools and frameworks were employed in this study:

1) Python

Python, known for its simplicity and readability, was the primary programming language used in this study for implementing the MLP model and conducting experiments. Its extensive libraries and frameworks that support machine learning and data analysis made it the ideal choice for this research.

2) TensorFlow

TensorFlow, an open-source machine learning framework developed by Google, was employed to build and train the MLP model. Its ability to handle large-scale computations and flexible architecture allowed for the development of complex neural network models, making it a crucial tool in this study.

3) Scikit-Learn

Scikit-Learn, a powerful machine learning library in Python, was used for data preprocessing, model evaluation, and hyperparameter tuning. This library provided the necessary functions for normalizing the dataset, splitting it into training and testing sets, and performing cross-validation to ensure the robustness of the model.

4) Keras

Keras, a high-level neural networks API written in Python, was utilized to simplify the construction and training of the MLP model. Running on top of TensorFlow, Keras offered a user-friendly API, enabling quick experimentation with different neural network architectures.

5) Google Colab

Google Colab, a free cloud service, allowed for the execution of Python code in a Jupyter notebook environment. This platform was instrumental in running experiments and training the MLP model, providing free access to GPUs and facilitating ease of sharing and collaboration for computationally intensive tasks.

F. Evaluation Metrics

Model performance was evaluated using the following metrics:

1.) Accuracy

Equation (2) shows the accuracy formula, defined as the ratio of true positive (TP) and true negative (TN) outcomes to the total number of predictions made by a machine learning model. This formula helps assess the model's overall effectiveness in predicting whether employees will leave or stay. The true positives represent the correct predictions of employee turnover, while true negatives indicate the correct predictions of employees who remain.

$$Accuracy = \frac{\text{TP} + \text{TN}}{Total\ Predictions} \qquad (2)$$

2.) Precision

Equation (3) shows precision which helps evaluate how many of the predicted employee turnovers (true positives) were actually correct, reducing the likelihood of false alarms (false positives). A high precision indicates that the model is accurate in predicting actual employee turnover, suggesting fewer incorrect predictions. This can guide HR professionals in making informed decisions about employee retention strategies while reducing unnecessary actions based on incorrect predictions.

$$Precision = \frac{TP}{TP + FP} \qquad (3)$$

3.) Recall

Equation (4) shows recall which measures the percentage of actual employee turnovers correctly predicted by the model. A high recall indicates that the model is effective at detecting employee turnover cases, minimizing the likelihood of false negatives. This is crucial when predicting employee turnover, as missing actual cases could lead to underestimating turnover risk and inadequate retention strategies.

$$Recall = \frac{TP}{TP + FN} \qquad (4)$$

4.) F1-Score

Equation (5) shows F1-Score which helps ensure a balanced approach, reflecting the model's performance in predicting employee turnover while minimizing false positives and false negatives. This metric can guide HR professionals in assessing the effectiveness of predictive models and implementing employee retention strategies with a higher degree of confidence.

$$F1Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \qquad (5)$$

5.) Number of Iterations/Epochs

The number of iterations, referring to the count of times the model's parameters are updated during training, is closely related to epochs, which represent complete passes through the entire training dataset. In this study, the primary objective is to reduce the number of iterations or epochs needed for the MLP model to achieve convergence, meaning the point where the model's performance stabilizes and no longer improves significantly with further training.

IV. RESULTS AND DISCUSSION

A. Model Performance

A performance comparison of various optimizers used to predict employee turnover using a Multilayer Perceptron (MLP) model is shown in Table 1. The optimizers evaluated include Adam with a TanH-based warm-up strategy, AdaMax, Nadam, and AdamP. Key performance metrics such as accuracy, precision, recall, and F1 score are reported as percentages.

Table 1: Optimizer's Performance Comparison

Optimizer	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Adam w/ TanH-based Warm-Up	95	99	99	99
AdaMax	94	98	98	99
Nadam	93	98	99	99
AdamP	93	98	99	98

Table 1 indicates that the Adam optimizer with a TanH-based warm-up strategy outperforms the other optimizers across all evaluated metrics. It achieves the highest accuracy of 95%, along with superior precision, recall, and F1 score, each at 99%. This suggests that incorporating a TanH-based warm-up strategy significantly enhances the performance of the Adam optimizer. AdaMax also performs well, with an accuracy of 94% and precision, recall, and F1 scores all at 98%. This shows that AdaMax is a robust alternative but still falls short compared to the improved Adam optimizer with the warm-up strategy. Nadam and AdamP show similar performance levels, each with an accuracy of 93%. While they achieve high precision and recall scores of 98% and 99%, respectively, their overall performance is slightly lower than that of the Adam optimizer with the TanH-based warm-up strategy.

B. *Iterations*

A comparison of the number of iterations required by different optimizers to converge during the training of MLP model to predict employee turnover is presented in Table 2. The optimizers evaluated include Adam with a TanH-based warm-up strategy, Nadam, AdaMax, and AdamP. The iterations are measured in terms of epochs, indicating the number of times the entire dataset is processed during training.

Table 2: Optimizer's Iterations Comparison

Optimizer	Iteration
Adam w/ TanH-based Warm-Up	50 Epochs
Nadam	75 Epochs
AdaMax	100 Epochs
AdamP	110 Epochs

Table 2 shows that the Adam optimizer with a TanH-based warm-up strategy requires the fewest iterations to converge, needing only 50 epochs. This significantly lower number of epochs highlights the efficiency of incorporating a TanH-based warm-up strategy in reducing the training time while achieving convergence. In contrast, Nadam requires 75 epochs to converge, which is 50% more than the Adam with TanH-based warm-up. Although Nadam performs better than AdaMax and AdamP, it still requires more iterations than the improved Adam optimizer. AdaMax and AdamP require even more epochs, with AdaMax converging in 100 epochs and AdamP in 110 epochs. This suggests that while these optimizers are effective, they are less efficient compared to the Adam optimizer with the TanH-based warm-up strategy.

V. CONCLUSION

This research study aimed to enhance the efficiency of the Adam optimizer by incorporating a warm-up strategy using the Hyperbolic Tangent (tanh) function to reduce the number of iterations or epochs required for training Multilayer Perceptron (MLP) models in predicting employee turnover. The results demonstrated that the Adam optimizer with the TanH-based warm-up strategy significantly outperformed other optimizers, including Nadam, AdaMax, and AdamP, in terms of both performance metrics and training efficiency. Therefore, integrating a TanH-based warm-up strategy with the Adam optimizer not only improves the predictive accuracy and reliability of MLP models for employee turnover but also enhances the training efficiency by reducing the number of required iterations. This novel optimization technique provides a valuable tool for organizations seeking efficient and accurate employee turnover prediction models, thereby enabling them to implement proactive retention strategies and maintain operational efficiency. Future research could further explore the applicability of this approach across different datasets and neural network architectures to validate and extend these promising findings.

ACKNOWLEDGMENT

This research was supported by the Commission on Higher Education–Staff and Instructors' Knowledge Advancement Program (CHED – SIKAP): Application Number 2022-10-FT-223228.

REFERENCES

[1] R. Chakraborty, K. Mridha, R. N. Shaw, and A. Ghosh, "Study and Prediction Analysis of the Employee Turnover using Machine Learning Approaches," in 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies, GUCON 2021, Institute of Electrical and Electronics Engineers Inc., Sep. 2021. doi: 10.1109/GUCON50781.2021.9573759.

[2] Y. Zhao, M. K. Hryniewicki, F. Cheng, B. Fu, and X. Zhu, "Employee turnover prediction with machine learning: A reliable approach," in Advances in Intelligent Systems and Computing, Springer Verlag, 2018, pp. 737–758. doi: 10.1007/978-3-030-01057-7_56.

[3] K. Dayal, M. Shukla, and S. Mahapatra, "Disease Prediction Using a Modified Multi-Layer Perceptron Algorithm in Diabetes," EAI Endorsed Trans Pervasive Health Technol, vol. 9, May 2023, doi: 10.4108/eetpht.9.3926.

[4] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980

[5] M. Liu, D. Yao, Z. Liu, J. Guo, and J. Chen, "An Improved Adam Optimization Algorithm Combining Adaptive Coefficients and Composite Gradients Based on Randomized Block Coordinate Descent," Comput Intell Neurosci, vol. 2023, pp. 1–14, Jan. 2023, doi: 10.1155/2023/4765891.

[6] P. Gopalani and A. Mukherjee, "Global Convergence of SGD On Two Layer Neural Nets," Oct. 2022, [Online]. Available: http://arxiv.org/abs/2210.11452

[7] M. Teng, H. Zhu, C. Liu, and H. Xiong, "Exploiting Network Fusion for Organizational Turnover Prediction," ACM Trans Manag Inf Syst, vol. 12, no. 2, Jun. 2021, doi: 10.1145/3439770.

[8] P. Goyal et al., "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour," Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.02677

[9] F. M. Shakiba and M. Zhou, “Novel Analog Implementation of a Hyperbolic Tangent Neuron in Artificial Neural Networks,” IEEE Transactions on Industrial Electronics, vol. 68, no. 11, pp. 10856–10867, Nov. 2021, doi: 10.1109/TIE.2020.3034856.

[10] S. Konwer, M. Sojan, P. Adeeb Kenz, S. K. Santhosh, T. Joseph, and T. S. Bindiya, “Hardware Realization of Sigmoid and Hyperbolic Tangent Activation Functions,” in Proceedings of the 2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 84–89. doi: 10.1109/IAICT55358.2022.9887382.

[11] S. A. Alzaeemi, “Accelerating Hyperbolic Tangent Activation Function in Neuro Symbolic Integration using Agent Based Modelling,” International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 3, pp. 3366–3375, Jun. 2020, doi: 10.30534/ijatcse/2020/136932020.

[12] M. Atef, D. S. Elzanfaly, and S. Ouf, “Early Prediction of Employee Turnover Using Machine Learning Algorithms 135 Original Scientific Paper”