Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 4, August 2025, Page. 1811-1830
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.4899

Development of a Distributed Gradient Boosting Forest Algorithm with Residual Connections in Data Classification

Rayhan Dhafir Respati[1], Sopian Soim*[2], Mohammad Fadhli[3]

[1,2,3]Telecommunication Engineering, Politeknik Negeri Sriwijaya, Indonesia

Email: [2]sopiansoim@gmail.com

Received : Jun 17, 2025; Revised : Jun 27, 2025; Accepted : Jun 30, 2025; Published : Aug 18, 2025

Abstract

The growing complexity and volume of data across various domains necessitate machine learning models that are scalable and robust for large-scale classification tasks. Ensemble methods such as Gradient Boosting Decision Trees (GBDT) demonstrate effectiveness; however, they encounter issues concerning scalability and training stability when applied to very deep architectures. This work presents a novel enhancement using residual connections derived from deep neural networks into the Distributed Gradient Boosting Forest (DGBF) algorithm. By enabling direct gradient propagation across layers, residual connections solve the vanishing gradient problem and so improve gradient flow, accelerate convergence, and stabilise the training process. The Residual DGBF model was assessed using seven distinct datasets across the domains of cybersecurity, financial fraud, phishing, and malware detection. The Residual DGBF consistently surpassed the baseline DGBF in terms of accuracy, precision, recall, and F1-score across all datasets. Particularly in datasets marked by imbalanced classes and complex feature interactions, this suggests improved generalisation and higher predictive accuracy. By proving more stable and strong gradients across the depth of the model, layer-wise gradient magnitude analysis supports these improvements and so confirms the effectiveness of residual connections in deep ensemble learning. This work improves ensemble techniques by combining the scalability and interpretability of decision tree ensembles with the residual architecture optimising benefits. The proposed Residual DGBF enables future research on enhanced deep boosting frameworks by offering a strong and scalable method to address challenging real-world classification tasks.

Keywords : *Distributed Gradient Boosting Forest, Ensemble Learning, Gradient Magnitude, Residual Connections, Vanishing Gradient.*

1. INTRODUCTION

In recent years, ensemble learning methods have become a cornerstone in machine learning, especially for structured and tabular data, due to their robust performance and interpretability [1], [2], [3], [4]. However, real-world applications (particularly in cybersecurity and fraud detection) pose increasingly complex challenges. These include extreme class imbalance, high-dimensional features, and evolving attack patterns that degrade model performance over time. Traditional classifiers, and even standard ensemble techniques, often struggle to generalize effectively under these conditions. In high-stakes domains such as phishing detection, malware analysis, and transaction fraud detection, misclassification can lead to significant security breaches and financial loss. This risk is further exacerbated by class imbalance and evolving behavior patterns, which are pervasive issues in financial fraud and cybersecurity applications [5], [6], [7]. Among these, Gradient Boosting Decision Trees (GBDT) have become rather well-known since their capacity to repeatedly combine weak learners to generate strong predictive models [8], [9]. The iterative character of GBDT helps the model to concentrate progressively on challenging-to-predict samples, so improving general performance [10], [11], [12], [13]

However, classical boosting methods often face scalability issues when applied to very large datasets or complex problem domains [14]. This limitation arises from the sequential nature of the training process, which constrains parallelization and results in extended training durations. These constraints present considerable difficulties in big data contexts and real-time applications where computational efficiency is essential [12], [15], [16]. To address these issues, Distributed Gradient Boosting Forest (DGBF) has been presented as a new design that expands boosting into a distributed, layered ensemble framework [17].

DGBF proposes an ensemble technique based on gradient descent, using Gradient Boosting and Random Forest as its basic learners. Random Forest is an ensemble learning method that builds numerous decision trees during the training phase, ultimately producing final predictions by averaging results in regression tasks or utilizing majority voting in classification tasks [18]. Gradient Boosting (GB) provides a robust ensemble method in machine learning and develops predictive models sequentially, with the objective of reducing errors at each iteration. This methodology optimizes differentiable loss functions, providing significant versatility across different problem types [19]. DGBF diverges from conventional sequential boosting by training multiple decision trees concurrently at each layer, thereby creating a deep forest structure akin to neural network architectures. This design enables DGBF to acquire complex data representations while utilizing the interpretability of tree models and efficiently scaling across distributed computing environments [17].

Several studies have recently explored ensemble deep learning architectures in various domains. Ganaie et al. [1] and Mohammed and Kora [2] presented comprehensive reviews on ensemble methods, highlighting their robustness and challenges in scaling. In the field of cybersecurity, Hossain and Islam [3] proposed a hybrid ensemble approach for botnet detection, emphasizing the need for improved feature learning. Additionally, Borisov et al. [4] reviewed the limitations of deep neural networks on tabular data, reinforcing the relevance of tree-based methods like DGBF for structured data. While DGBF has demonstrated promise in distributed learning [18], its ability to propagate gradients efficiently across deep layers remains underexplored.

Despite its promising capabilities, DGBF, like many deep learning-inspired ensemble methods, can suffer from the vanishing gradient problem [17], [20]. As the ensemble deepens with more layers, the gradient signals propagated during training may weaken exponentially, leading to slower convergence and reduced learning effectiveness [1], [21]. Deep neural networks identify this phenomenon, which limits the depth models can be trained with efficiency.

A key advance in mitigating this problem came with the inclusion of residual connections in deep neural networks, as exemplified by ResNet [22], [23], [24], [25], [26], [27]. Residual learning employs bypass connections that transmit data input across layers, maintaining gradient magnitude and enabling the training of extremely deep models without deterioration [24]. Specifically, residual connections allow models to learn identity mappings, which ease the flow of gradients and prevent vanishing, making the training of very deep architectures feasible. A comprehensive survey on deep residual learning elaborates on its theory, applications, and benefits, confirming residual learning as a powerful mechanism for stabilizing and enhancing deep model training [22].

Apart from convolutional networks, this architectural innovation has been extensively applied in transformer models in natural language processing [28], [29]. However, residual connection variants such as Post-Layer Normalization (Post-LN) and Pre-Layer Normalization (Pre-LN) each suffer from different limitations: Post-LN models can suffer from gradient vanishing that hinders training very deep Transformers, while Pre-LN models risk representation collapse that reduces model capacity [29]. The ResiDual Transformer architecture has been proposed to address these issues by integrating the advantages of both variants through the use of dual residual connections (Pre-Post-LN). This method concurrently tackles gradient vanishing and representation collapse, thereby ensuring stable gradient

flow and preserving substantial representational diversity [29], [30]. Theoretical analyses prove that ResiDual provides a lower bound on the gradient norm and preserves model expressiveness, and empirical results confirm its superior performance on machine translation benchmarks of varying scales [29].

While residual learning has been extensively adopted in deep neural network architectures, particularly in convolutional and transformer models [22], [24], [28], its integration into ensemble-based tree learners such as DGBF has not been systematically studied. Existing implementations of DGBF focus primarily on distributed efficiency and structural layering, without addressing training stability in deeper configurations. This indicates a gap in leveraging residual mechanisms to improve gradient flow and convergence within distributed tree-based boosting frameworks. Motivated by developments in residual learning architectures, our research incorporates dual residual connections into the DGBF framework, introducing a Residual DGBF model that maintains gradient magnitude across layers and reduces convergence instability. This facilitates more profound and stable distributed boosting forests that leverage gradient stabilisation approaches inspired by neural networks. The novelty of this study lies in the explicit integration of dual residual connections into the DGBF framework, forming a Residual Distributed Gradient Boosting Forest (Residual DGBF). To the best of our knowledge, this is the first attempt to combine gradient-stabilizing residual paths with a layered boosting structure, enabling deeper, more stable training of ensemble tree models while preserving interpretability and scalability.

We evaluate the model across seven diverse and challenging datasets spanning financial fraud and cybersecurity domains, including TUANDROMD, creditcard fraud, transaction fraud detection, phising dataset, phishing email classification, malware behavior detection, and general fraud detection datasets.

We assess the performance of the residual Distributed Gradient Boosting Forest (DGBF) using accuracy, precision, recall, and F1-score as primary metrics. The analysis of layer-wise gradient magnitudes indicates that residual learning stabilises the training process by maintaining strong gradient signals throughout the entire ensemble depth. This study introduces an improved DGBF model that enables more profound ensemble learning, rendering it applicable to various real-world classification tasks.

2. METHOD

The research process in this study is structured into several interconnected phases, as illustrated in Figure 1. It begins with the data collection stage, in which seven publicly available benchmark datasets from cybersecurity and fraud detection domains are selected based on their relevance and representativeness. The second phase is data preprocessing, where raw data undergoes cleaning to remove duplicates and invalid entries, followed by handling missing values through median imputation and encoding categorical features. This phase guarantees that the input data is consistent, numerical, and appropriate for model ingestion. The third phase involves the development of the predictive model. Here, the Distributed Gradient Boosting Forest (DGBF) is implemented and subsequently enhanced using residual connections to form the Residual DGBF model. This architectural innovation is aimed at improving gradient flow and training stability in deep ensemble structures. The fourth phase consists of the experimental setup, including the configuration of hyperparameters, environment settings, and model training protocols. The datasets are divided into training and testing subsets using stratified sampling to preserve class distribution. Finally, the performance of the models is evaluated using classification metrics accuracy, precision, recall, and F1-score as well as layer-wise gradient magnitude analysis to measure gradient stability across layers. Each of these stages is elaborated in the subsequent subsections to offer a thorough perspective of the methodology employed.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 4, August 2025, Page. 1811-1830
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.4899

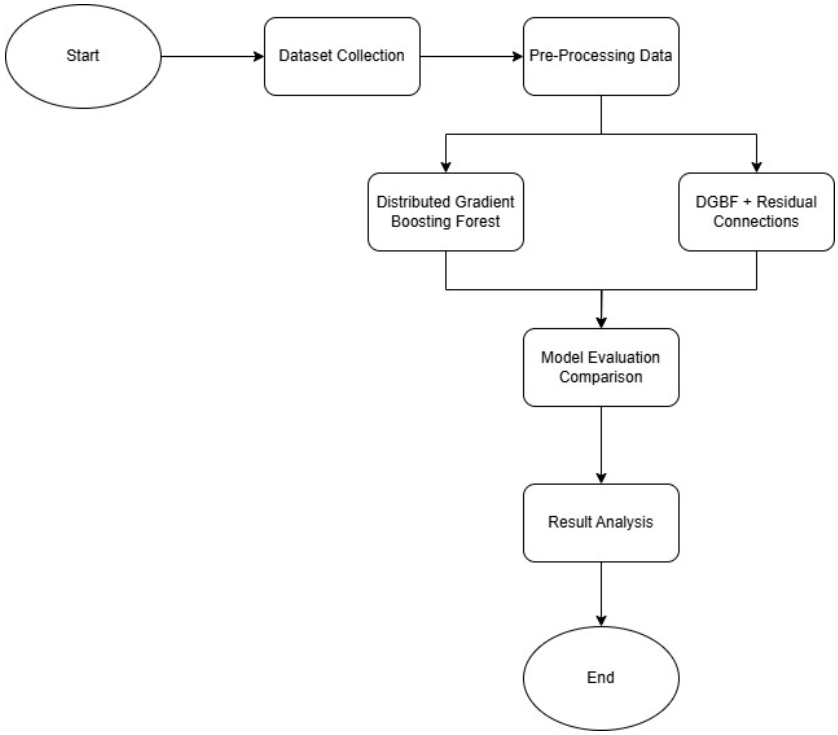


Figure 1. Research Flowchart

2.1. Data Collection

Data collection techniques encompass the systematic methods employed by researchers to gather relevant information or data necessary for the study. These techniques constitute a fundamental and strategic component of the research methodology, as the quality and accuracy of collected data directly impact the validity and reliability of the research findings [31]. This work evaluates the performance of the suggested models using seven different datasets gathered and applied. Every dataset reflects unique domains and features, so offering a varied collection of samples to fully evaluate the generalizability and resilience of the systems. Crucially important elements affecting model performance are the variations in the number of instances, feature types, and class distributions among the datasets. Table 1 lists the main features of every dataset together with the target variable, name, source, total sample count, number of attributes. The deliberate choice of these datasets guarantees that the evaluation spans a broad spectrum of real-world classification difficulties, so strengthening the validity and applicability of the experimental results.

Each dataset was carefully selected not only for its domain diversity (ranging from cybersecurity to financial fraud), but also for its distinct data characteristics that pose specific modeling challenges. For instance, the *creditcard* dataset is highly imbalanced, with fraudulent transactions comprising less than 0.2% of the total samples. This makes it a suitable benchmark for evaluating the model's performance under extreme class skew. On the other hand, the *TUANDROMD* and *Android_Malware_Benign* datasets are high-dimensional, containing 242 and 328 features respectively. This introduces sparsity and increases the risk of overfitting, making them ideal scenarios to test the generalization capacity of deep ensemble models.

Additionally, the *Phishing_Email* dataset is notably low-dimensional (only 3 features), offering a unique contrast to test model behavior on minimal input information. The *dynamic_api_call_sequence_per_malware_100_0_306* dataset contains sequential behavioral logs

Vol. 6, No. 4, August 2025, Page. 1811-1830
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.4899

from malware samples, representing temporally ordered, non-i.i.d. data. This structure challenges the model to capture complex time-dependent patterns.

By including datasets that vary in size, dimensionality, class distribution, and domain-specific complexity, this study ensures that the proposed model is rigorously evaluated under diverse real-world conditions. These considerations are essential to test not only classification performance but also the resilience and adaptability of the model to different data structures and noise characteristics.

Table 1. Experiment datasets

Dataset	Source	Sample	Attributes	Target Variable
Android_Malware_Benign	[32]	4.464	328	Label
creditcard	[33]	284.807	31	Class
dataset_phishing	[34]	11.430	89	status
dynamic_api_call_sequence_per_malware_100_0_306	[35]	43.876	102	malware
Phishing_Email	[36]	18.640	3	Email Type
Transaction_Fraud_Detection_2023	[37]	568.630	31	Class
TUANDROMD	[38]	4.464	242	Label

2.2. Pre-Processing Data

Data pre-processing is a critical step in arranging raw data for effective analysis and modeling. It refers to a set of approaches used to improve data quality, preserve consistency, and convert data into a format suitable for machine learning algorithms. This work's pre-processing pipelines include data cleaning to eliminate duplicates and erroneous values, handle missing values to complete incomplete records, encode category variables into numerical form, and partitioning the dataset into training and testing subsets for thorough model evaluation [39]. The steps outlined collectively enhance the accuracy and reliability of the predictive models developed in this research. The preprocessing phase involved removing duplicate records, handling missing values using median imputation, and replacing infinite values with the maximum finite value per feature. Categorical variables were encoded using LabelEncoder, while the data was split into training and test sets with a stratified 80:20 ratio to preserve class distribution.

2.2.1. Data Cleaning

Data cleansing is a critical step in the preparation process that ensures the dataset's quality and dependability. This strategy involves identifying and eliminating duplicate entries, resolving conflicts, and correcting erroneous or aberrant information. Duplicate items were eliminated in this study to prevent biased model training and enhance overall data quality. Furthermore, infinite values were detected and replaced with suitable alternatives to avert computational problems during analysis.

2.2.2. Handling Missing Values

Effective management of missing data is crucial for preserving the integrity and usability of the dataset. Missing values may arise from various factors, such as data entry errors and equipment malfunctions. This study handled missing values by imputing the mean of the corresponding feature columns. This straightforward yet efficacious technique maintains dataset size while mitigating potential bias from absent data. Alternative imputation techniques may be employed based on the characteristics of the data.

2.2.3. Data Encoding

Most machine learning algorithms necessitate numerical input; therefore, categorical variables must be appropriately transformed. Data encoding converts non-numeric categorical features into numerical representations that the model can interpret [40]. In this investigation, all categorical columns were label encoded, with each unique category assigned an integer value. This technique preserves category information while allowing for smooth integration with downstream machine learning models.

2.2.4. Split Data

The conventional method for evaluating model performance on raw data involves dividing the dataset into training and testing subsets. The testing set assesses the model's generalization ability, whereas the training set facilitates its fitting process. Using random sampling with a set random seed to guarantee repeatability, the set of data was split here into 20% for testing. and 80% for training. This partitioning technique reduces overfit and supports strong performance assessment.

2.3. Model Development

2.3.1. Distributed Gradient Boosting Forest

Distributed Gradient Boosting Forest (DGBF) is an innovative ensemble learning method that synthesizes and integrates the concepts of Random Forest and Gradient Boosting into a profound, graph-based architecture. DGBF diverges from conventional boosting techniques that incrementally construct trees to reduce residual errors, instead distributing the learning process across numerous layers of tree ensembles, akin to the hierarchical architecture of neural networks. Each layer in DGBF comprises many decision trees that autonomously learn segments of the gradient signal, which together approximate the comprehensive gradient descent trajectory. This design enables DGBF to attain distributed representation learning without dependence on backpropagation or alteration of the internal architecture of decision trees.

Mathematically, the DGBF prediction function is defined as:

$$F_L(x) = \sum_{l=1}^{L} \frac{1}{T} \sum_{t=1}^{T} h_{l,t}(x) \quad (1)$$

where:

L is the number of boosting layers;

T is the number of trees in each forest (layer);

$h_{l,t}(x)$ is the prediction of the t-th tree in the l-th layer.

The learning process at each boosting step optimizes over the distributed gradient components as:

$$g'_{l,t}(x) = \frac{1}{T}(T.y_i - h_{l-1,t}(x_i)) \quad (2)$$

Where:

$g'_{l,t}(x)$: Distributed residual gradient (pseudo-residual) allocated specifically for tree t in layer l for sample x_i.

y_i: The true target value for sample i.

$h_{l-1,t}(x_i)$: The prediction output of tree t in layer l for the sample x_i.

T: The number of trees in a single layer.

Each tree is trained to minimize the loss with respect to its assigned gradient component:

$$h_{l,t}(x) = \arg\min_{h} \sum_i L(g'_{l,t}(x), h(x_i)) \quad (3)$$

Where:
$h_{l,t}(x)$: the decision tree t in layer l currently being trained.
$\arg\min_{h}$: the function h that minimizes the loss value.
L: the loss function measuring the difference between the target and the prediction.
$g'_{l,t}(x)$: the residual gradient (the new target for the tree) for sample x_i.
$h(x_i)$: the prediction of the tree h on sample x_i.

Dynamic sampling is employed across layers to mitigate overfitting and improve learning variety. Earlier layers of trees are trained on smaller subsamples, with the sample size continuously augmented in later levels. This emulates curricular learning, wherein the model initially acquires overarching patterns and subsequently integrates more intricate information.

DGBF serves as a natural extension of both Random Forest and Gradient Boosting, which are considered special examples. When set up with a single layer, it functions as a conventional Random Forest, however decreasing the number of trees per layer yields a configuration akin to traditional Gradient Boosting. Empirical findings indicate that DGBF consistently surpasses both Random Forest and Gradient Boosting across diverse datasets, affirming its enhanced generalization capability and proficiency in acquiring deep representations.

Illustratively, in contrast to neural networks where the output of every neuron is transferred to the following layer., DGBF propagates averaged tree predictions forward. During training, distributed gradients not activations are transmitted from one layer to the next, enabling each tree to independently learn part of the residual while maintaining coordinated ensemble learning. This architecture allows DGBF to learn complex hierarchical representations similar to neural networks, without modifying the fundamental non-parametric structure of decision trees.

2.3.2. Residual Connection

Residual connection is an architectural innovation that addresses the vanishing issue in training very deep neural networks. The vanishing issue refers to the phenomena in which increasing network depth beyond a certain point leads to higher training error and poorer performance, not primarily caused by overfitting but by optimization difficulties. To address this, residual learning reformulates the target function to be learned.

Formally, let $H(x)$ denote the desired underlying mapping for a set of stacked layers with input x. Instead of directly approximating $H(x)$, residual learning allows the network to approximate a residual function defined as

$$F(x) := H(x) - x \quad (5)$$

which implies the original mapping can be expressed as

$$H(x) = F(x) + x \quad (6)$$

This simple reformulation has profound implications: if the identity mapping is optimal (i.e., no transformation is needed), the network can more easily push the residual $F(x)$ toward zero, rather than forcing the layers to learn an explicit identity mapping. By learning residuals, the network focuses on the disparity between inputs and results, which empirically results in easier optimization and faster convergence.

The practical implementation of this idea is realized through (also called skip connections), which perform identity mapping by skipping one or more layers and directly adding the input x to the output of the stacked layers representing $F(x)$. This addition is element-wise and adds no new parameters or computational cost. As a result, residual networks can be trained end-to-end with normal backpropagation while incurring minimum overhead.

Experimental evidence shows that networks with residual connections yield lower training error and improved generalization even at depths greater than 100 layers. This residual formulation stabilizes gradient flow, mitigates vanishing gradients, and so allows for the successful optimization of deeper architectures, marking a significant development in deep learning methodology.

2.3.3. Distributed Gradient Boosting Forest + Residual Connection

The Distributed Gradient Boosting Forest (DGBF) framework's ability to learn deep hierarchical representations is enhanced by the addition of residual connections, which also solves common training issues with deep ensembles. Residual connections allow the model to pass input or intermediate predictions directly between layers, freeing up following trees to learn residual mistakes rather than the entire goal mapping. This considerably reduces the issues of degradation and vanishing gradients commonly found in deep models.

The integration of DGBF's distributed, layered structure with residual learning enables each layer to enhance the predictions of preceding layers through the learning of incremental residual functions. This residual formulation corresponds with the concept of incrementally minimising prediction errors, while the distributed architecture enables concurrent learning among trees within each layer. As a result, the model benefits from improved optimisation dynamics and increased representational capacity.

Formally, let $F_l(x)$ denote the ensemble prediction at layer l, and $h_{l,t}(x)$ represent the prediction of the t-th tree in layer l, where T is the number of trees per layer. The residual-enhanced DGBF updates its prediction using the formula:

$$F_l(x) = F_{l-1}(x) + \frac{1}{T}\sum_{t=1}^{T} h_{l,t}(x) \qquad (7)$$

Here:
$F_l(x)$ is the updated prediction after the l-th layer,
$F_{l-1}(x)$ is the prediction from the previous $(l-1)$-th layer,
$h_{l,t}(x)$ is the output from the t-th decision tree in the current layer l,
T is the total number of trees in each layer,

The term $\sum_{t=1}^{T} h_{l,t}(x)$ represents the averaged residual function learned by the trees at layer l, which serves as a correction or refinement to the previous prediction.

Similar to the success of residual networks in deep learning, where learning residual mappings has been shown to lower gradient flow and improve convergence, by focusing on residuals the model simplifies the optimisation task. The residual connections solve problems including vanishing or inflating gradients and the degradation problem often found in very deep models by offering consistent gradient transmission across layers. This produces a model that is both articulate and simpler to train. Empirical evidence demonstrates that the residual-enhanced DGBF attains superior convergence speed and accuracy relative to traditional tree ensemble methods devoid of residual connections. Furthermore, the residual design maintains consistent gradient magnitudes throughout layers, facilitating the effective training of deep ensembles while preserving interpretability and computational efficiency.

2.4. Evaluation Metrics

This study employed a set of standard evaluation metrics to measure the performance of the proposed classification models. These metrics include accuracy, precision, recall, and F1-score, which are particularly relevant in binary classification tasks involving imbalanced datasets [41]. Accuracy evaluates the overall correctness of the model, representing the proportion of correctly classified instances. Precision indicates the model's ability to minimize false positives, defined as the ratio of true positives to all positive predictions. Recall (also referred to as sensitivity) demonstrates the model's ability to accurately identify all pertinent positive instances. The F1-score serves as a balanced metric, integrating both precision and recall into a singular value through the calculation of their harmonic mean.

$$Accuracy = \frac{TN+TP}{TN+TP+FP+FN} \quad (8)$$

$$F1 - Score = \frac{2*TP}{2*TP+FP+FN} \quad (9)$$

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

These metrics were chosen due to their robustness in evaluating the classification performance of models under class imbalance, a common challenge in fraud detection and cybersecurity datasets. In addition to these traditional metrics, layer-wise gradient magnitude was analyzed to assess the internal stability of deep ensemble models, particularly those incorporating residual connections [42], [43]. Let gl denote the gradient vector at layer l, and d the number of units in that layer. The magnitude of the gradient is defined as:

$$|g_l| = \sqrt{\sum_{i=1}^{d} \left(\frac{\partial \mathcal{L}}{\partial h_{l,i}}\right)^2} \quad (12)$$

This formulation helps monitor the strength of gradient signals during backpropagation. Stable and non-vanishing gradients across layers suggest improved convergence and effective learning. Thus, the integration of both conventional evaluation metrics and gradient-based analysis provides a comprehensive performance assessment of the proposed model.

3. RESULT

This section presents the comprehensive results obtained from the entire research process, following the methodological sequence outlined in the previous chapter. The process began with the selection of seven diverse datasets covering cybersecurity and fraud detection domains. These datasets were preprocessed through several stages, including data cleaning, handling of missing values, and encoding of categorical variables. Subsequently, two models were implemented and evaluated: the baseline Distributed Gradient Boosting Forest (DGBF) and its enhanced variant using residual connections (DGBF + RC). Each model was trained and tested using an 80:20 split on each dataset. The evaluation focused on key performance metrics including accuracy, precision, recall, and F1-score, and further analyzed layer-wise gradient magnitudes to observe training stability and convergence behaviors.

As outlined in the methodology (Section 2), the research followed a sequential process including dataset selection, preprocessing, model development, and evaluation. Table 1 in the methodology section summarizes the characteristics of the datasets used. In this section, we present the results

Vol. 6, No. 4, August 2025, Page. 1811-1830
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.4899

obtained from applying the proposed Residual DGBF model across those datasets. The results are organized following the stages described in the methods, beginning with the experimental setup, followed by performance evaluation using standard classification metrics and gradient-based training analysis.

3.1. Experimental Setup

The tests sought to evaluate the efficacy of the suggested model under controlled and replicable conditions. Seven benchmark datasets, covering various areas and complexities, were utilized to assess the generalizability of the findings. The preprocessing of each dataset included managing missing values, encoding categorical variables, and normalizing numerical characteristics, as detailed in the methodology section. The datasets were divided into training and testing subsets in an 80:20 ratio, and all experiments were conducted multiple times with varying random splits to reduce variability.

The Distributed Gradient Boosting Forest (DGBF) model featuring residual connections was executed using hyperparameters that were optimized via initial tuning. The primary hyperparameters of the proposed model, such as the number of layers, the number of trees per layer, and the maximum tree depth, were consistently fixed across all datasets to facilitate fair and comparable evaluation. Table 2 summarizes the configuration of these hyperparameters, offering a clear overview of the model settings utilized in the experiments. Baseline models, including standard Random Forest and Gradient Boosting, were trained under comparable conditions utilizing default parameters from commonly used libraries.

Table 2. Hyperparameter Configuration

Hyperparameter	Value	Description
Number of Layers (L)	12	Number of boosting iterations (layers)
Number of Trees per Layer (T)	10	Number of decision trees in each layer
Learning Rate	0.1	Step size for gradient updates

An extensive evaluation of classification performance was conducted using measures such as accuracy, precision, recall, and F1-score. The magnitudes of the gradients were recorded for each layer to analyse the training dynamics of the deep ensemble. The studies were conducted within a computing environment equipped with Windows 11 Home Single Language 64-bit (10.0, Build 26100) as the operating system, Intel Core 5 Ultra 125H, Intel Arc Graphics, and 16GB RAM, ensuring sufficient resources for efficient training and evaluation.

All seven datasets utilized in this study were obtained from publicly available repository Kaggle, as detailed in Table 1. These datasets were selected to represent diverse real-world classification challenges in cybersecurity and fraud detection domains. The data acquisition process ensured that only datasets with clear labeling and sufficient sample size were included.

Prior to model training, each dataset underwent a consistent preprocessing pipeline as described in Section 2.1. This involved duplicate removal, missing value imputation using median or mean strategies, label encoding is applied to categorical features, followed by a stratified split into training and testing sets at an 80:20 ratio to maintain class distribution. Preprocessing was automated using Python and standard libraries such as pandas and scikit-learn.

The model training and evaluation were conducted using the Residual DGBF model implemented in Python. Each experiment was run multiple times with randomized seeds to ensure result stability. Hyperparameters such as the number of layers, trees per layer, and learning rate were kept constant across datasets as summarized in Table 2. Performance metrics, including accuracy, precision, recall,

Vol. 6, No. 4, August 2025, Page. 1811-1830
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.4899

and F1-score, were computed using standard formulas (Equations 8–11), and gradient magnitude analysis was applied for deeper model interpretation.

3.2. Performance Evaluation

Adding residual connections to the Distributed Gradient Boosting Forest (DGBF) design significantly improves the classification performance over a wide range of datasets, as demonstrated by the comprehensive results in Table 3. As presented in Table 3, the residual-enhanced variant (DGBF + RC) consistently surpasses the baseline DGBF model in terms of accuracy, precision, recall, and F1-score across all datasets. This quantitative comparison highlights the positive impact of residual learning on deep ensemble methods.

By allowing each layer to explicitly learn the residual errors in respect to the predictions of previous layers, the residual connection approach solves basic optimisation challenges usually observed in deep ensembles, including vanishing gradients and training deterioration. This method increases gradient flow inside the model, so enabling the DGBF + RC to converge faster and more consistently to produce better solutions. Crucially for datasets with class imbalances or complex feature distributions, the new model achieves a more fair trade-off between precision and recall.

The quantitative performance presented in Table 3 indicates that the DGBF + RC model consistently surpasses the standard DGBF in almost all assessed datasets. For instance, in the TUANDROMD dataset as shown in Table 3, accuracy increases by approximately 1.5%, rising from 94.74% to 96.24%. Precision improves notably from 0.9310 to 0.9655, while recall also advances from 0.8438 to 0.8750, indicating improved detection of positive classes and a reduction in false positives. The aggregate enhancements elevate the F1-score from 0.8852 to 0.9180, signifying improved robustness and efficacy in categorisation.

Table 3. Result Evaluation

Dataset	Model	Accuracy	Precision	Recall	F1-Score
Android_Malware_Benign	DGBF	95.5994	0.9486	0.9632	0.9559
	DGBF + RC	95.7511	0.9515	0.9632	0.9573
creditcard	DGBF	99.9454	0.9403	0.7000	0.8025
	DGBF + RC	99.9471	0.9545	0.7000	0.877
dataset_phishing	DGBF	94.4882	0.9411	0.9477	0.9444
	DGBF + RC	94.6194	0.9420	0.9495	0.9457
dynamic_api_call_sequence_per_	DGBF	98.5869	0.9876	0.9980	0.9928
malware_100_0_306	DGBF + RC	98.6553	0.9882	0.9981	0.9931
Phishing_Email	DGBF	67.2118	0.6754	0.8896	0.7678
	DGBF + RC	67.7480	0.6790	0.8927	0.7713
Transaction_Fraud_Detection_20	DGBF	99.9552	0.9996	0.9995	0.9996
23	DGBF + RC	99.9604	0.9997	0.9995	0.9996
TUANDROMD	DGBF	94.7368	0.9310	0.8438	0.8852
	DGBF + RC	96.2406	0.9655	0.8750	0.9180

Comparable patterns are evident in additional datasets such as creditcard and dynamic_api_call_sequence_per_malware_100_0_306, as shown in Table 3, where precision and F1-score improvements further support the robustness of the residual-enhanced model. The Android_Malware_Benign and dynamic_api_call_sequence_per_malware_100_0_306 datasets demonstrate consistent enhancements across all metrics, thereby reinforcing the generalisability of the residual connection approach.

Although certain datasets, like Phishing_Email, show more modest improvements, the residual-enhanced DGBF consistently outperforms the baseline in key metrics. This indicates that the incorporation of residual learning yields stable enhancements, irrespective of the complexity or noise levels of the dataset.

The findings in Table 3 indicate that residual connections are a crucial architectural improvement in the DGBF framework. They enhance model performance by facilitating deeper and more stable ensemble learning, especially in the context of challenging, high-dimensional, and imbalanced datasets. The findings indicate that the DGBF + RC model serves as a scalable, accurate, and reliable solution for various real-world classification challenges.

3.3. Layer-wise Gradient Magnitude Results

To obtain deeper insight into how residual connections influence the training dynamics of deep ensemble models, we performed a rigorous examination of the layer-wise gradient magnitudes across seven diverse benchmark datasets. Understanding gradient propagation is essential, as the strength and stability of gradients at each layer directly affect the model's ability to learn effectively and converge reliably. Poor gradient flow, especially in deeper layers, can lead to vanishing or exploding gradients, which impedes optimization and degrades performance.

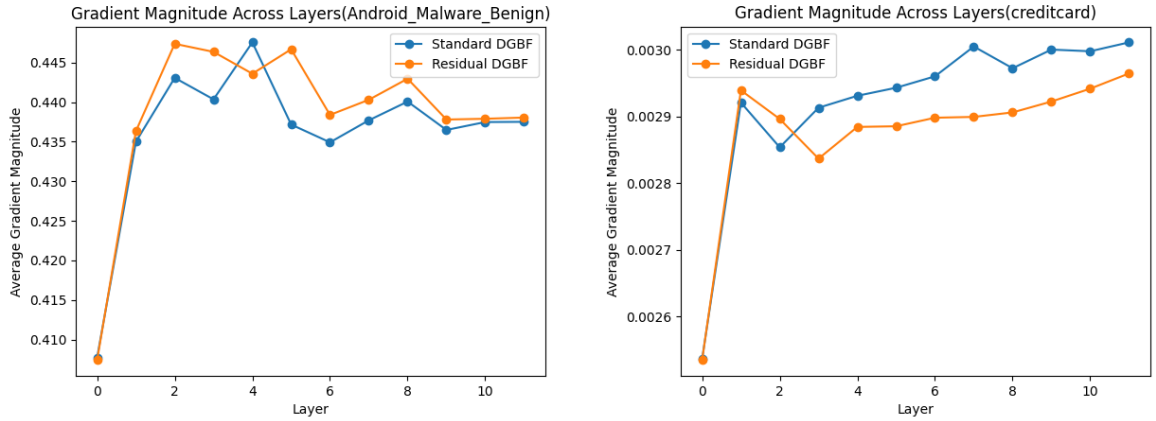
In this study, for each dataset, we computed and visualized the average gradient magnitude at every layer of the Distributed Gradient Boosting Forest (DGBF) in both its standard form and its enhanced variant with residual connections (DGBF + RC). These visualizations, presented as individual figures per dataset, serve as a granular diagnostic tool to observe how gradients behave through the ensemble depth. By comparing the gradient magnitudes layer-by-layer between the two models, we can directly evaluate the effectiveness of residual connections in mitigating common gradient-related issues in deep models.

Such a layer-wise perspective not only highlights the resilience of residual-enhanced models in maintaining robust gradient signals across all layers but also provides empirical evidence explaining the improved convergence speed and predictive accuracy observed in earlier performance evaluations. This comprehensive gradient analysis forms a critical component of our experimental validation, bridging theoretical advantages with practical outcomes in distributed gradient boosting frameworks.

Figure 2(a) depicts the average gradient magnitude across the layers for the Android_Malware_Benign dataset, showing that the residual-enhanced DGBF maintains stronger gradient flow than the standard model, particularly in middle layers. The residual-enhanced DGBF consistently maintains higher gradient magnitudes than the standard model throughout most layers, especially in the middle layers. This sustained gradient strength indicates improved gradient flow and reduced risk of vanishing gradients, which supports more stable and effective training. The stronger gradients in the residual model contribute to better parameter updates and likely explain its superior performance on this dataset.

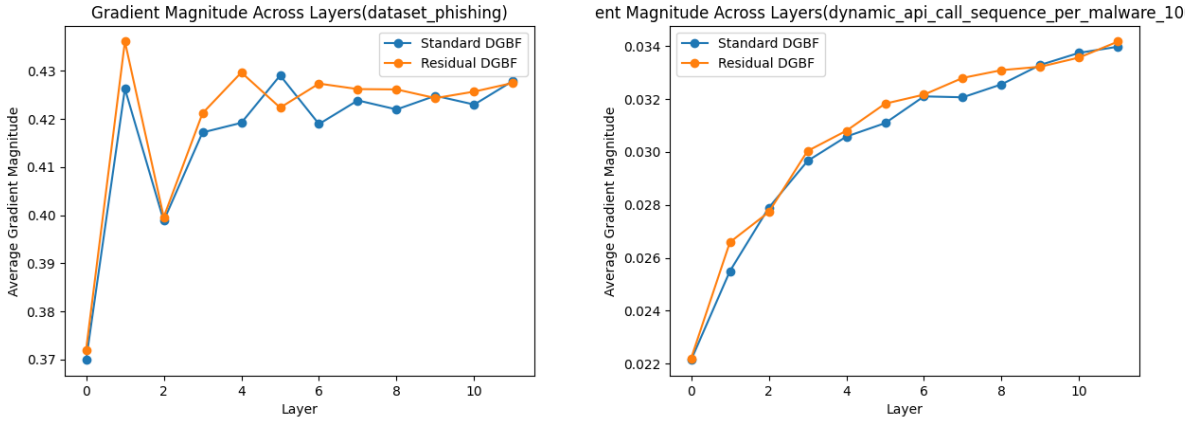
Figure 2(b) presents the layer-wise average gradient magnitudes for the creditcard dataset. The standard DGBF generally exhibits slightly higher gradient magnitudes across most layers compared to the residual-enhanced DGBF. Both models show relatively low gradient values overall, reflecting the dataset's complexity and class imbalance. The residual DGBF's slightly lower gradients may indicate more stable and focused learning, potentially helping reduce overfitting. Despite this, gradient magnitudes remain consistent across layers for both models, suggesting stable training dynamics.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 4, August 2025, Page. 1811-1830
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.4899



(a) (b)

Figure 2. Distribution of gradient magnitude values of both models. (a) Android_Malware_Benign; (b) creditcard.



(a) (b)

Figure 3. Distribution of gradient magnitude values of both models. (a) dataset_phishing; (b) dynamic_api_call_sequence_per_malware_100_0_306.

Figure 3(a) depicts the average gradient magnitude across layers for the dataset_phishing dataset. The residual-enhanced DGBF generally maintains higher gradient magnitudes than the standard DGBF, particularly in the early and middle layers. Both models exhibit a sharp increase in gradient magnitude at the initial layers, followed by fluctuations in subsequent layers. The higher gradient values in the residual model suggest more effective gradient flow and stronger learning signals, which contribute to improved training stability and potentially better predictive performance on this dataset.

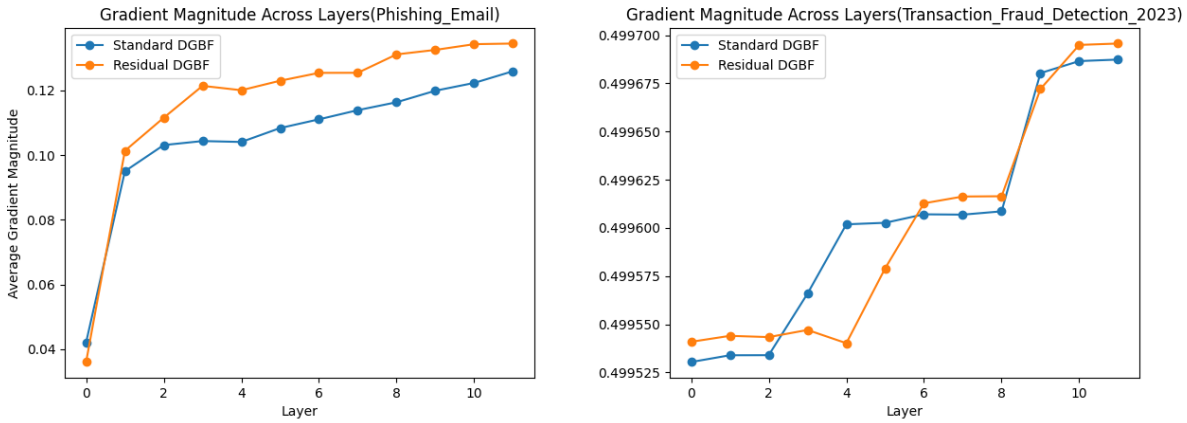
Figure 3(b) illustrates the average gradient magnitude across layers for the dynamic_api_call_sequence_per_malware_100_0_306 dataset. Both the standard DGBF and residual-enhanced DGBF models show a gradual increase in gradient magnitude from the initial to the deeper layers, indicating strengthening gradient signals as the layers progress. The residual DGBF slightly outperforms the standard model by maintaining marginally higher gradient magnitudes in most layers, particularly in the early and middle stages. This suggests that residual connections contribute to improved gradient propagation and training stability, enabling the model to learn more effectively from complex sequential malware data.

Figure 4(a) shows the average gradient magnitude across layers for the Phishing_Email dataset. The residual-enhanced DGBF maintains consistently higher gradient magnitudes than the standard

Vol. 6, No. 4, August 2025, Page. 1811-1830
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.4899

DGBF across all layers, with a widening gap as the layer depth increases. This indicates that residual connections significantly enhance the flow of gradients and help stop the problem of the diminishing gradient, especially in deeper layers. The stronger gradients facilitate more stable and effective training, which is crucial for the complex and noisy characteristics of phishing email data.

Figure 4(b) illustrates the average gradient magnitudes across layers for the Transaction_Fraud_Detection_2023 dataset. Both the standard DGBF and residual-enhanced DGBF maintain very high and nearly identical gradient magnitudes throughout all layers, indicating strong and stable gradient propagation in this dataset. The residual DGBF shows a slight edge in maintaining marginally higher gradients in some intermediate layers, which may contribute to its improved convergence and predictive performance. The consistently high gradient values reflect the model's effective training dynamics on this relatively balanced and well-structured dataset.



(a) (b)

Figure 4. Distribution of gradient magnitude values of both models. (a) Phishing_Email; (b) Transaction_Fraud_Detection_2023.

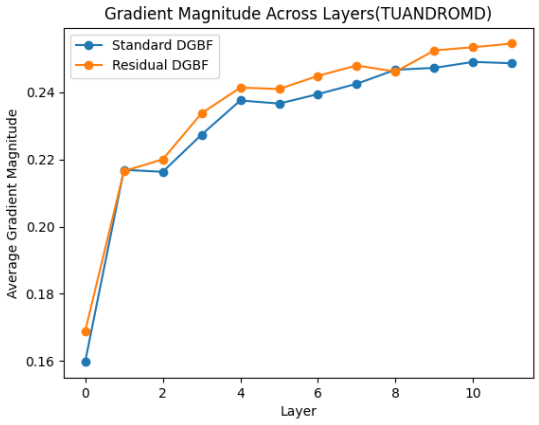


Figure 5. Distribution of gradient magnitude values of both models on the TUANDROMD dataset.

Figure 5 presents the average gradient magnitude across layers for the TUANDROMD dataset. The residual-enhanced DGBF consistently maintains higher gradient magnitudes compared to the standard DGBF, particularly from the middle to deeper layers. This indicates that residual connections improve gradient preservation, enabling more stable gradient flow during training. The gradual increase

Vol. 6, No. 4, August 2025, Page. 1811-1830
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.4899

and higher values of gradient magnitudes in the residual model suggest better learning dynamics, which likely contribute to the improved classification performance observed for this dataset.

The layer-wise gradient magnitude analysis across all seven datasets consistently demonstrates the effectiveness of integrating residual connections within the Distributed Gradient Boosting Forest framework. The residual-enhanced models show stronger and more stable gradient signals throughout the network layers compared to the standard DGBF models. This improved gradient flow helps mitigate common optimization challenges such as vanishing gradients, thereby promoting more reliable training and faster convergence. The empirical evidence from these gradient visualizations complements the quantitative improvements observed in classification performance, reinforcing the crucial role of residual connections in enhancing deep ensemble learning.

4. DISCUSSIONS

The experimental results presented in the previous chapter demonstrate the clear advantage of integrating residual connections into the Distributed Gradient Boosting Forest (DGBF) framework. Across a variety of datasets with differing characteristics and complexities, the residual-enhanced DGBF (DGBF + RC) consistently outperformed the baseline DGBF model in key classification metrics such as accuracy, precision, recall, and F1-score. This improvement validates the hypothesis that residual learning significantly enhances the model's ability to capture complex patterns and mitigate common optimization challenges inherent in deep ensemble structures.

One of the fundamental contributions of residual connections is their ability to improve gradient propagation throughout the network layers, as evidenced by the layer-wise gradient magnitude analysis. The stronger and more stable gradient signals observed in the residual-enhanced models facilitate more effective parameter updates, which contribute to faster convergence and improved generalization. This is particularly important in deep ensemble models where vanishing gradients can severely hamper training efficiency and model performance.

Several previous studies have highlighted the benefits of residual learning and ensemble methods in improving model performance and training dynamics. For instance, He et al. [19] and Wu et al. [23] demonstrated that residual connections significantly enhance gradient flow and enable deeper architectures in convolutional networks without suffering from degradation. Our findings align with this observation, showing that residual-enhanced DGBF models maintain stronger gradient magnitudes and achieve better convergence. In the context of ensemble learning, Mohammed and Kora [2] noted that ensemble deep learning methods often struggle with overfitting and optimization in deep structures. The improvements observed in our residual DGBF framework, particularly in datasets with high dimensionality and class imbalance, support the hypothesis that integrating residual connections into ensemble models helps alleviate these challenges.

The variation in performance gains across datasets also provides insights into the robustness of the residual connection mechanism. For datasets with significant class imbalance or complex feature interactions, such as TUANDROMD and creditcard fraud detection, the residual-enhanced DGBF achieved more pronounced improvements. Meanwhile, datasets with inherently noisier or more ambiguous data, such as Phishing_Email, showed more modest but still consistent benefits. This suggests that residual connections enhance the model's learning dynamics in diverse practical scenarios without overfitting or compromising stability.

Moreover, the consistent improvements in precision and F1-score across most datasets underscore the model's improved balance in handling false positives and false negatives. This balance is crucial in real-world applications like fraud detection and malware classification, where minimizing incorrect classifications directly impacts operational efficiency and security.

While the residual connections significantly improve training dynamics and predictive accuracy, the distributed nature of the DGBF framework ensures scalability and parallelization, ensuring the method is applicable to extensive and high-dimensional datasets. The synergy between distributed boosting and residual learning offers a promising direction for future ensemble methods that require both interpretability and depth.

The integration of residual connections into DGBF not only addresses critical optimization issues but also enhances the model's practical applicability across varied domains. Future work may explore further optimization techniques, adaptive residual mechanisms, or extensions to other ensemble architectures to build on these promising findings.

The consistent performance improvements across multiple datasets underscore the fundamental strength of combining residual learning with distributed gradient boosting. This outcome not only validates the technical soundness of the proposed approach but also suggests that deep ensemble models can be significantly stabilized with relatively lightweight architectural modifications. The author believes that this stability, particularly in the face of vanishing gradients and imbalanced class distributions, represents a critical step toward making deep tree-based models more robust and interpretable.

The findings of this research carry notable implications for both the academic and applied domains. In the academic context, this study contributes to the growing body of literature on hybrid ensemble architectures by demonstrating that residual learning principles, traditionally applied in deep neural networks, can be effectively adapted into tree-based boosting frameworks. In applied settings, the improved performance and training stability of the DGBF + RC model are especially relevant to domains such as fraud detection and malware classification, where data are often imbalanced, noisy, or high-dimensional. By addressing key challenges such as vanishing gradients and convergence instability, this work offers a scalable and interpretable solution that bridges the gap between theory and practice in intelligent classification systems.

5. CONCLUSION

This study proposed an enhanced Distributed Gradient Boosting Forest (DGBF) model integrated with residual connections to address optimization challenges commonly faced in deep ensemble learning. By leveraging residual learning, the model effectively preserves gradient magnitudes across multiple layers, mitigating the vanishing gradient problem and enabling more stable and efficient training of deep boosting forests. Extensive experiments conducted on seven diverse and challenging datasets from domains such as cybersecurity and financial fraud detection demonstrated the superior performance of the residual-enhanced DGBF over the baseline model.

Quantitative evaluations revealed consistent improvements in accuracy, precision, recall, and F1-score metrics, confirming that the integration of residual connections enhances the model's ability to capture complex data patterns and maintain robust generalization across varied scenarios. Layer-wise gradient magnitude analysis further validated that residual connections facilitate stronger and more stable gradient propagation, which underpins the observed gains in training convergence and predictive accuracy. These observations reinforce the conclusion drawn in earlier discussions that residual learning can act as an implicit regularizer in deep ensembles, improving generalization while reducing training instability.

In addition to practical improvements, this research contributes conceptually by demonstrating that residual learning principles, originally developed for deep neural networks, can be effectively translated into tree-based ensemble architectures. This enhances the theoretical foundation of ensemble learning and opens new directions for hybrid model design that balances interpretability and depth. The

approach is particularly valuable for domains with high-dimensional and imbalanced data, where robustness and scalability are essential.

Future work may explore adaptive residual weighting mechanisms, integration with attention modules, or deployment in streaming and real-time environments to further improve the model's efficiency and applicability across broader machine learning tasks.

CONFLICT OF INTEREST

The writers of this work affirm that they have no financial or other conflicts of interest that could influence the results of their study.

ACKNOWLEDGEMENT

We are grateful to our first and second supervisors, Mr. Sopian Soim, S.T., M.T., and Mr. Mohammad Fadhli, S.Pd., M.T., for all of the guidance, support, and constructive criticism they provided over the course of our research and study preparation.

REFERENCES

[1] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *Eng Appl Artif Intell*, vol. 115, p. 105151, Oct. 2022, doi: 10.1016/J.ENGAPPAI.2022.105151.

[2] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 757–774, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.014.

[3] M. A. Hossain and M. S. Islam, "A novel hybrid feature selection and ensemble-based machine learning approach for botnet detection," *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-48230-1.

[4] V. Borisov, T. Leemann, K. Sebler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep Neural Networks and Tabular Data: A Survey," *IEEE Trans Neural Netw Learn Syst*, vol. 35, no. 6, pp. 7499–7519, Jun. 2024, doi: 10.1109/TNNLS.2022.3229161.

[5] J. G. Sherwin Akshay, T. Vinusha, R. Sharon Bianca, C. K. Sarath Krishna, and G. Radhika, "Enhancing Credit Card Fraud Detection: A Comparative Analysis of Anomaly Detection Models," *2024 IEEE International Conference on Computer Vision and Machine Intelligence, CVMI 2024*, 2024, doi: 10.1109/CVMI61877.2024.10781986.

[6] J. G. Sherwin Akshay, T. Vinusha, R. Sharon Bianca, C. K. Sarath Krishna, and G. Radhika, "Enhancing Credit Card Fraud Detection with Deep Learning and Graph Neural Networks," *2024 15th International Conference on Computing Communication and Networking Technologies, ICCCNT 2024*, 2024, doi: 10.1109/ICCCNT61001.2024.10725042.

[7] Z. ; Yang *et al.*, "Leveraging Mixture of Experts and Deep Learning-Based Data Rebalancing to Improve Credit Fraud Detection," *Big Data and Cognitive Computing 2024, Vol. 8, Page 151*, vol. 8, no. 11, p. 151, Nov. 2024, doi: 10.3390/BDCC8110151.

[8] Z. Zhang and C. Jung, "GBDT-MO: Gradient-Boosted Decision Trees for Multiple Outputs," *IEEE Trans Neural Netw Learn Syst*, vol. 32, no. 7, pp. 3156–3167, Jul. 2021, doi: 10.1109/TNNLS.2020.3009776.

[9] Y. Shi, G. Ke, Z. Chen, S. Zheng, and T.-Y. Liu, "Quantized Training of Gradient Boosting Decision Trees," in *Advances in Neural Information Processing Systems*, Dec. 2022, pp. 18822–18833. Accessed: May 17, 2025. [Online]. Available: https://github.com/Microsoft/LightGBM

[10] J. S. Yang, C. Y. Zhao, H. T. Yu, and H. Y. Chen, "Use GBDT to Predict the Stock Market," *Procedia Comput Sci*, vol. 174, pp. 161–171, Jan. 2020, doi: 10.1016/J.PROCS.2020.06.071.

[11] J. Chen, J. Zhong, and Z. Chen, "Research On the Pricing Model of Second-Hand Sailboats Based on GDBT Model," *BCP Business & Management MEEA*, vol. 2023, p. 24, 2023.

[12] Q. Lv, "Hyperparameter tuning of GDBT models for prediction of heart disease," *https://doi.org/10.1117/12.2668449*, vol. 12602, pp. 686–691, Apr. 2023, doi: 10.1117/12.2668449.

Vol. 6, No. 4, August 2025, Page. 1811-1830
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.4899

[13] J. Xu *et al.*, "Predicting cerebral edema in patients with spontaneous intracerebral hemorrhage using machine learning," *Front Neurol*, vol. 15, Oct. 2024, doi: 10.3389/fneur.2024.1419608.
[14] L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, and H. H. Olsson, "Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions," *Inf Softw Technol*, vol. 127, p. 106368, Nov. 2020, doi: 10.1016/J.INFSOF.2020.106368.
[15] H. Seto *et al.*, "Gradient boosting decision tree becomes more reliable than logistic regression in predicting probability for diabetes with big data," *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-20149-z.
[16] Q. Li, Z. Wen, and B. He, "Practical Federated Gradient Boosting Decision Trees," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 4642–4649, Apr. 2020, doi: 10.1609/AAAI.V34I04.5895.
[17] S. and K. N. A. and F. S. A. and R. K. Akram Faiz and Aziz, "Integrating Artificial Bee Colony Algorithms for Deep Learning Model Optimization: A Comprehensive Review," in *Solving with Bees: Transformative Applications of Artificial Bee Colony Algorithm*, K. Raza, Ed., Singapore: Springer Nature Singapore, 2024, pp. 73–102. doi: 10.1007/978-981-97-7344-2_5.
[18] Á. Delgado-Panadero, J. A. Benítez-Andrades, and M. T. García-Ordás, "A generalized decision tree ensemble based on the NeuralNetworks architecture: Distributed Gradient Boosting Forest (DGBF)," *Applied Intelligence*, vol. 53, no. 19, pp. 22991–23003, Jul. 2023, doi: 10.1007/s10489-023-04735-w.
[19] T. Turino, R. E. Saputro, and G. Karyono, "Comparative Analysis of Decision Tree, Random Forest, Svm, and Neural Network Models for Predicting Earthquake Magnitude," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 2, pp. 755–774, Apr. 2025, doi: 10.52436/1.JUTIF.2025.6.2.2378.
[20] N. A. Syam, N. Arifin, W. Firgiawan, and M. F. Rasyid, "Comparison of SVM and Gradient Boosting with PCA for Website Phising Detection," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 2, pp. 691–708, Apr. 2025, doi: 10.52436/1.JUTIF.2025.6.2.4344.
[21] M. Tanveer *et al.*, "Ensemble deep learning in speech signal tasks: A review," *Neurocomputing*, vol. 550, p. 126436, Sep. 2023, doi: 10.1016/J.NEUCOM.2023.126436.
[22] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," Sep. 01, 2022, *MDPI*. doi: 10.3390/app12188972.
[23] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif Intell Rev*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.
[24] F. He, T. Liu, and D. Tao, "Why ResNet Works? Residuals Generalize," *IEEE Trans Neural Netw Learn Syst*, vol. 31, no. 12, pp. 5349–5362, Dec. 2020, doi: 10.1109/TNNLS.2020.2966319.
[25] Y. Hu, L. Deng, Y. Wu, M. Yao, and G. Li, "Advancing Spiking Neural Networks Toward Deep Residual Learning," *IEEE Trans Neural Netw Learn Syst*, 2024, doi: 10.1109/TNNLS.2024.3355393.
[26] W. Xu, Y. L. Fu, and D. Zhu, "ResNet and its application to medical image processing: Research progress and challenges," *Comput Methods Programs Biomed*, vol. 240, p. 107660, Oct. 2023, doi: 10.1016/J.CMPB.2023.107660.
[27] W. Wu, L. Huo, G. Yang, X. Liu, and H. Li, "Research into the Application of ResNet in Soil: A Review," *Agriculture 2025, Vol. 15, Page 661*, vol. 15, no. 6, p. 661, Mar. 2025, doi: 10.3390/AGRICULTURE15060661.
[28] A. Vaswani *et al.*, "Attention Is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 6000–6010. doi: 10.5555/3295222.3295349.
[29] S. Xie *et al.*, "ResiDual: Transformer with Dual Residual Connections," Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.14802
[30] H. Wang, S. Ma, L. Dong, S. Huang, D. Zhang, and F. Wei, "DeepNet: Scaling Transformers to 1,000 Layers," *IEEE Trans Pattern Anal Mach Intell*, 2024, doi: 10.1109/TPAMI.2024.3386927.
[31] G. Daruhadi and P. Sopiati, "Pengumpulan Data Penelitian," *J-CEKI : Jurnal Cendekia Ilmiah*, vol. 3, no. 5, pp. 5423–5443, Aug. 2024, doi: 10.56799/JCEKI.V3I5.5181.

[32] D. Revaldo, "Android Malware Detection Dataset." Accessed: May 19, 2025. [Online]. Available: https://www.kaggle.com/datasets/dannyrevaldo/android-malware-detection-dataset
[33] J. Arvidsson, "Credit Card Fraud." Accessed: May 18, 2025. [Online]. Available: https://www.kaggle.com/datasets/joebeachcapital/credit-card-fraud
[34] S. Tiwari, "Web page Phishing Detection Dataset." Accessed: May 18, 2025. [Online]. Available: https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset
[35] A. Oliveira, "Malware Analysis Datasets: API Call Sequences." Accessed: May 18, 2025. [Online]. Available: https://www.kaggle.com/datasets/ang3loliveira/malware-analysis-datasets-api-call-sequences
[36] Cyber Cop, "Phishing Email Detection." Accessed: May 18, 2025. [Online]. Available: https://www.kaggle.com/datasets/subhajournal/phishingemails
[37] N. Elgiriyewithana, "Credit Card Fraud Detection Dataset 2023." Accessed: May 18, 2025. [Online]. Available: https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023
[38] J. Arvidsson, "Android Malware Detection." Accessed: May 18, 2025. [Online]. Available: https://www.kaggle.com/datasets/joebeachcapital/tuandromd
[39] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/J.GLTP.2022.04.020.
[40] A. Mutholib, N. A. Rahim, T. S. Gunawan, and M. Kartiwi, "Trade-Space Exploration with Data Preprocessing and Machine Learning for Satellite Anomalies Reliability Classification," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3543813.
[41] S. Amini, M. Saber, H. Rabiei-Dastjerdi, and S. Homayouni, "Urban Land Use and Land Cover Change Analysis Using Random Forest Classification of Landsat Time Series," *Remote Sensing 2022, Vol. 14, Page 2654*, vol. 14, no. 11, p. 2654, Jun. 2022, doi: 10.3390/RS14112654.
[42] M. Liu, L. Chen, X. Du, L. Jin, and M. Shang, "Activated Gradients for Deep Neural Networks," *IEEE Trans Neural Netw Learn Syst*, vol. 34, no. 4, pp. 2156–2168, Apr. 2023, doi: 10.1109/TNNLS.2021.3106044.
[43] I. Dubinin and F. Effenberger, "Fading memory as inductive bias in residual recurrent networks," *Neural Networks*, vol. 173, p. 106179, May 2024, doi: 10.1016/J.NEUNET.2024.106179.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 4, August 2025, Page. 1811-1830
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.4899