RESEARCH ARTICLE

Adaptive Deep Neural Network for Solving Multiclass Problems

Komgrich Onprasonk[1], **Tammanoon Panyatip**[1,*], and **Panatda Phothinam**[1]

[1]Department of Computer Science and Information Technology, Faculty of Science and Health Technology, Kalasin University Kalasin 46230, Thailand

*Corresponding author: Tammanoon Panyatip, tammanoon@ksu.ac.th

Article Information

Article History:
Received: 19 October 2024
Revised: 15 May 2025
Accepted: 8 August 2025
Published: 19 October 2025

Keywords:
Deep Neural Network
Machine Learning
Multiclass
Neural Network

Abstract

In recent years, multiclass classification has gained significant attention due to its wide-ranging applications in fields such as healthcare, finance, and image recognition. The ability to accurately classify data into multiple categories is essential for developing intelligent and robust systems. This research compares the performance of several machine learning and deep learning algorithms for multiclass classification tasks, with a focus on adaptive techniques in neural networks. The evaluated algorithms include Support Vector Machines (SVM), One-vs-Rest Logistic Regression (OvR-LR), Deep Neural Networks (DNN), Dropout-enhanced DNN, and Adaptive Regularization-based DNN. The experimental evaluation was conducted using both the train–test split and 5-fold cross-validation methods to ensure result reliability and generalizability. The Adaptive Regularization-DNN model achieved the highest performance among all tested approaches, with an accuracy of 98.75% under the train–test split and 97.3% under cross-validation. These results highlight the model's robustness and its effectiveness in minimizing overfitting in structured multiclass classification problems. Performance metrics including precision, recall, F1-score, and accuracy were used to provide a comprehensive evaluation of each model's capabilities.

1. Introduction

In today's data-driven world, Big Data is characterized by immense volume, variety, and velocity, presenting challenges that exceed the processing capabilities of traditional database systems. Data can exist in both structured and unstructured forms, encompassing both quantitative and qualitative aspects. Choosing the appropriate features for classification tasks is critical and largely depends on the nature of the data and the objectives of the task. These tasks can involve either binary classification or, more commonly in complex scenarios, multiclass classification (Moral et al., 2022), where data must be categorized into multiple distinct, non-overlapping classes.

The importance of multiclass classification lies in its widespread applications across various domains, such as image recognition (Gao and Zhou, 2021; Liu et al., 2021), speech processing (Pawar and Dhage, 2020), and medical diagnosis (Tiwari et al., 2022), where categorization into more than two classes is often essential. For example, in medical imaging, diagnosing diseases from X-ray or MRI data requires precise differentiation among multiple possible conditions (Choudhuri and Paul, 2021), while in natural language processing (Vernikou et al., 2022), classifying text or speech into distinct languages or emotions relies heavily on effective multiclass strategies (Khan and Zubair, 2020; Zehra et al., 2021). This makes multiclass classification a significant problem in machine learning research.

Several machine learning approaches exist for solving multiclass problems, including Support Vector Machines (SVM) (Kumar Sahu and Pandey, 2023; Guo et al., 2021), One-vs-Rest Logistic Regression (Hussain and Ashraf, 2023), and Neural Networks, particularly Deep Neural Networks (DNNs) (Park et al., 2022). Among these, DNNs are widely recognized for their ability to capture complex patterns from large datasets, making them well-suited for multiclass classification (Chen et al., 2022; Bagla and Kumar, 2023).

ISSN 3088-1803 |

However, multiclass classification presents unique challenges, especially as the complexity of the data grows. This complexity often leads to overfitting (Bejani and Ghatee, 2021), a common issue in deep learning models. The model can become overly focused on the training data, learning specific patterns that do not generalize effectively to new, unseen data. As a result, the model may show strong performance on the training set but struggle to handle test data due to overfitting to the intricate details that are only relevant within the training environment.

Overfitting is a critical issue in deep learning because it hampers a model's ability to generalize and accurately predict new, unseen data. Overfitting occurs when a model captures noise or insignificant details in the training data, reducing its performance on the test data (Rice et al., 2020). To combat this, various regularization techniques have been developed (Thakkar and Lohiya, 2021), including Dropout (Lee and Lee, 2020), L2 regularization (Zhu and Liu, 2021), and adaptive learning methods (Ni et al., 2021), identified overfitting and prolonged training time as two key challenges in multilayer neural network learning, especially in deep learning. To address these issues, dropout and batch normalization have emerged as two widely accepted techniques that help improve model performance and efficiency (Garbin et al., 2020).

Given these challenges, it is crucial to develop adaptive DNN techniques that can efficiently address multiclass classification problems while minimizing the risk of overfitting. Regularization and adaptive techniques (Abuduweili et al., 2021), such as those using dropout and dynamic architecture tuning, allow DNNs to learn generalized patterns without becoming overly complex, leading to better performance on test data (Lv et al., 2020; Liu et al., 2024). This research focuses on developing and evaluating adaptive deep learning methods tailored for multiclass problems, with the goal of improving model performance and generalization ability.

The research objectives are to compare algorithms for solving multiclass classification problems and to enhance deep neural network algorithms.

In this article, the presentation is structured as follows: Section 2 covers the related work, Section 3 explains the methods, Section 4 presentes the experimental results, and Section 5 provides the conclusion.

2. Related Work

This section reviews existing literature and studies relevant to multiclass classification and the application of various machine learning techniques. Numerous researchers have explored algorithms such as Support Vector Machines (SVM), Logistic Regression, and Deep Neural Networks (DNN) for effective classification across diverse datasets. Additionally, advancements in regularization methods and dropout techniques have been extensively investigated to combat overfitting, thereby enhancing the predictive performance of models in complex classification tasks. This body of work serves as a foundation for the methodologies applied in the current research, demonstrating the evolving landscape of multiclass classification strategies.

2.1 Support Vector Machines (SVM)

Support Vector Machines (SVM) is a method for classifying data by appropriately determining a hyperplane that best separates the data into two parts. SVM uses a kernel function to aid in the classification process (Panyatip et al., 2022). The SVM algorithm works by finding the widest margin between data points from each class. The use of a kernel function allows for the separation of non-linear data, making SVM well-suited for solving complex problems. Tang et al. (2019) explored a new method for multiclass classification with their Regular Simplex Support Vector Machine (RSSVM). Traditional SVM methods for multiclass classification use partitioning strategies, which limit the efficiency and sparsity of the model. To overcome these challenges, RSSVM maps K classes to the vertices of a (K-1)-dimensional regular simplex, treating the problem as a (K-1)-output learning task. Unlike traditional methods that require multiple classifiers, RSSVM uses a single minimization process with linear inequality constraints, enhancing the model's sparsity and computational efficiency.

While SVM is effective for binary and well-separated multiclass problems, it does not adapt its internal structure based on data characteristics and often requires manual tuning of hyperparameters and kernels. This lack of adaptability limits its performance on more complex, nonlinear datasets—a gap that adaptive DNNs aim to address.

2.2 One-vs-Rest (OvR)-Logistic Regression

One-vs-Rest (OvR) (Pawara et al., 2020) Logistic Regression is a technique used to solve multiclass classification problems by employing multiple logistic regression models. In OvR, one logistic regression model is created for each class, where the class of interest is treated as the positive class, and all other classes are considered negative. Therefore, the total number of logistic regression models equals the number of classes in the problem. OvR works effectively in solving multiclass classification tasks by using logistic regression, which often yields good results when the classes are well-separated. Dong et al. (2018) conducted a study on single-label multiclass image classification by deep logistic regression, thoroughly analyzing the standard learning objective functions for multiclass classification in CNNs: softmax regression (SR) for single-label scenarios and logistic regression (LR) for multi-label scenarios. The dataset used was the large-scale multi-label clothing attribute dataset, DeepFashion. The experiment setup involved testing two networks, ResNet-50 and MobileNet. The study observed the following: (1) The LR methods demonstrated a significant improvement over the vanilla algorithm, consistent with results in single-label object classification and person re-identification. (2) Both hard and soft selection strategies showed similar performance across different networks and metrics. Abramovich et al. (2021) conducted a study on multiclass classification by sparse multinomial logistic regression, where they explored high-dimensional multiclass classification using sparse multinomial logistic regression.

2.3 Deep Neural Network (DNN)

A Deep Neural Network (DNN) is an artificial neural network with multiple layers (deep) that consist of several nodes in each layer (Wang et al., 2020), enabling it to learn and classify complex data (Feng et al., 2020). DNN is a machine learning algorithm used to build models consisting of three main components:

1. Input Layer: This layer contains the input data, where each feature is represented by a node. For example, if the dataset contains 10 features, the input layer will have nodes representing $x_1, x_2, x_3, ..., x_{10}$.

2. Hidden Layer: This intermediate layer performs hidden processing, significantly influencing the model's performance. It consists of multiple layers, and each layer contains neurons that process input data using an activation function, as described by the following Equations:

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]} \tag{1}$$

$$A^{[l]} = g\left(Z^{[l]}\right) \tag{2}$$

where $A^{[l-1]}$ is the output from the previous layer $l-1$, $W^{[l]}$ is the weight matrix for layer l, $b^{[l]}$ is the bias vector for layer l, and g is the activation function.

3. Output Layer: This layer takes the results from the hidden layers and provides the final output, as follows:

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[L]} \tag{3}$$

$$Y = A^{[L]} = g\left(Z^{[L]}\right) \tag{4}$$

where L is the output layer, and $A^{[L]}$ is the predicted output. Thus, the general Equation for the neural network can be summarized as follows:

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]} \tag{5}$$

$$A^{[l]} = g\left(Z^{[l]}\right) \tag{6}$$

For each l, $1 \leq l \leq L-1$ represents the output of the final layer, and the output layer follows:

$$Z^{[L]} = W^{[L]} A^{[L-1]} + b^{[L]} \tag{7}$$

$$\hat{Y} = A^{[L]} = g\left(Z^{[L]}\right) \tag{8}$$

where L is the number of layers, and $W^{[L]}$ and $B^{[L]}$ are the weights and biases for layer L, respectively, g is the activation function, and $\hat{Y}$ is the predicted output of the network.

2.4 Dropout Deep Neural Network (Dropout-DNN)

The Dropout technique is used in deep neural networks to randomly drop some nodes from the hidden layers during training. This helps prevent the model from overfitting by ensuring that it does not rely too heavily on any one parameter during training. The dropout Equation is as follows:

$$a^{(l)} = M^{(l)}\, a^{(l)} \tag{9}$$

where $M^{(l)}$ is a binary mask generated by a Bernoulli distribution, determining which nodes will be active (1) and which will be deactivated (0). The element-wise multiplication (Hadamard product) applies this mask to the input. $a^{(l)}$ represents the input to the next layer after dropout. Garbin et al. (2020) conducted an empirical study that measured the increase in training and prediction times when using dropout and batch normalization. Their findings revealed that non-adaptive optimizers can outperform adaptive ones, but only with the caveat of significantly longer training times required for hyperparameter tuning. In contrast, adaptive optimizers performed well with minimal tuning, offering a more efficient solution in terms of training time. Manita et al. (2022) explored the application of dropout networks in random mode, demonstrating that this approach can be extended to a broad range of networks and even to approximation methods beyond neural networks. The key insight is an algebraic property that shows how deterministic networks can be precisely matched in expectation by their random network counterparts. Dropout is a widely used technique to mitigate overfitting by randomly deactivating neurons during training. However, dropout rates are often fixed and do not adjust dynamically based on the training process or dataset complexity. Our proposed adaptive regularization model improves upon this by tuning the regularization strength during training, resulting in better generalization.

2.5 Regularization Deep Neural Network (Regularization-DNN)

Regularization is a technique used to reduce overfitting in neural networks by adding a regularization term to the model's loss function. This penalty term adjusts the network's parameters and reduces the model's complexity by limiting the magnitude of the weights, thereby improving generalization. Regularization typically adds a penalty to the loss function, helping the model avoid overfitting and improve its predictive performance. Some recent studies, such as Zhao et al. (2019) propose adaptive regularization approaches using prior distributions and dynamic adjustments. However, these methods are not widely tested in structured multiclass tasks and often focus on unstructured data like images. Our study extends this concept by systematically comparing adaptive regularization with dropout and standard DNN approaches under a controlled environment.

While dropout and regularization are well-known overfitting prevention techniques, few studies have conducted a systematic comparison of these methods in structured, balanced multiclass classification tasks. This paper attempts to fill this gap by evaluating and contrasting both techniques under identical experimental conditions.

Summary of the Research Gap

While SVM and logistic regression offer stable but rigid solutions, DNNs bring the power of representational learning but suffer from overfitting. Regularization and dropout are effective countermeasures but are commonly implemented with fixed parameters. Very few studies systematically evaluate these techniques on structured tabular data or introduce adaptive mechanisms to dynamically control complexity during training. This research fills this gap by proposing and evaluating an Adaptive Regularization-DNN that incorporates dynamic L1 regularization into the training process. By comparing it against dropout-based and standard DNNs using consistent evaluation metrics and data conditions, this work contributes a clearer understanding of how adaptability enhances generalization in multiclass classification

3. Methods

This section outlines the methodologies employed in this research to address the multiclass classification problem. Various algorithms, including Support Vector Machines (SVM), One-vs-Rest (OvR) Logistic Regression, and Deep Neural Networks (DNN), were utilized to develop models capable of accurately classifying mobile price categories. Additionally, advanced techniques such as Dropout and Regularization were implemented to enhance model performance and mitigate issues related to overfitting.

3.1 Research Tools

The tools used in this research include the Python programming language and Google Colab. The dataset used in this research is the Mobile Price Classification dataset, obtained from the Kaggle website (`https://www.kaggle.com`), as of August 15, 2023.

3.2 Research Procedure

3.2.1 Data Selection and Inspection

Data Selection: The Mobile Price Classification dataset comprises 21 columns and 2,000 rows, distributed across 4 distinct classes.

Inspection: The dataset was examined for feature correlations and class balance. Each class contained an equal number of 500 rows. The data characteristics are illustrated in Figure 1.

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2

5 rows × 21 columns

Figure 1. The characteristics of the data.

3.2.2 Data Preparation and Cleaning

The data preparation process included handling missing values and removing the target label, 'Price_range', which consisted of four classes. Furthermore, the dataset was reviewed as follows.

1. Data correlations were analyzed to assess the relationships between features. This step is crucial to identify patterns and dependencies among the variables, which can impact the model's performance and accuracy in Fig. 2.

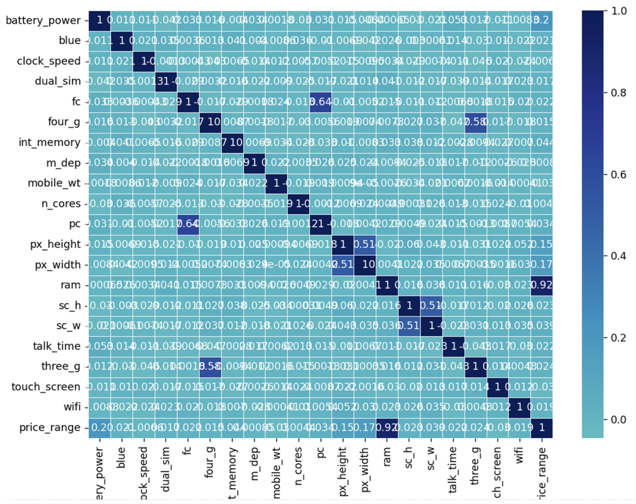

Figure 2. Data correlations.

2. Outlier Detection: Potential outliers in the dataset were analyzed, as shown in Fig. 3. Detecting outliers is critical for ensuring data integrity and improving model performance, as extreme values can distort the results of the analysis. By identifying and addressing these outliers, we can ensure the model's robustness and prevent bias in predictions.

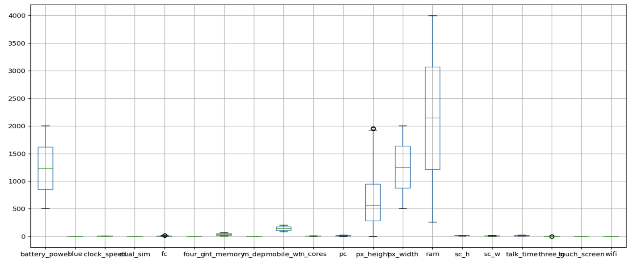

Figure 3. Outlier detection.

3.2.3 Model Development for Classification

The classification models developed in this research included: SVM, OvR Logistic Regression, DNN, Dropout-DNN, and Regularization-DNN. The adjustment of deep neural networks using L1 regularization techniques involves the following methods.

$$J(\theta) = \text{Loss}(\theta) + \lambda \sum_{i=1}^{n} |\omega_i| \tag{10}$$

where $J(\theta)$ is the total loss function, $\text{Loss}(\theta)$ is the loss function without the regularization term, λ is the regularization strength parameter, $\sum_{i=1}^{n} |\omega_i|$ is the sum of the absolute values of the weights.

The L1 regularization Equation added to the loss function to reduce overfitting can be written as follows.

$$J_{\text{regularized}}(W, b) = J(W, b) + \frac{\lambda}{m} \sum_{l=1}^{L} \left\| W^{[l]} \right\|_1 \tag{11}$$

where $J(W, b)$ is the normal loss function of the DNN, λ is the regularization strength parameter, m is the number of samples in the training set, $\left\| W^{[l]} \right\|_1$ is the absolute value of the weights in layer j.

In summary, when combining Equations 10 and 11, we obtain the Equation for training DNNs with L1 regularization techniques as follows.

$$J(W, b) = J_{\text{regularized}}(W, b) + \text{L1 Regularization Term} \tag{12}$$

The Regularization-DNN algorithm is designed to mitigate overfitting by incorporating a regularization term into the loss function, which penalizes large weight values in the neural network. Below is a high-level outline of how the Regularization-DNN can be expressed as an Algorithm 1:

Algorithm 1 Adaptive Regularization-DNN

Require: Dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, learning rate η, regularization strength λ, number of iterations T, batch size
Ensure: Trained weights W and biases b
Randomly initialize weights W and biases b
Set regularization parameter λ
for epoch $t = 1$ to T **do**
  Divide dataset D into mini-batches
  for each mini-batch **do**
    Forward Pass: Compute predicted output $\hat{\gamma} = f(x; W, b)$
    Loss Calculation:

$$L = L_{\text{original}} + \lambda \sum_j W_j^2$$

    Backward Pass: Compute gradients $\frac{\partial L}{\partial W}$, $\frac{\partial L}{\partial b}$
    Update Weights and Biases:

$$W \leftarrow W - \eta \cdot \frac{\partial L}{\partial W},$$

$$b \leftarrow b - \eta \cdot \frac{\partial L}{\partial b}$$

  end for
end for
Output: Trained W and b

3.3 Statistics Used to Measure Performance

The statistics used to measure performance include Precision, Recall, F1-Score, and Accuracy, with the details as follows.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

$$F1\text{-Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{15}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

where

- TP (True Positive) refers to the data that is predicted as positive and matches the true label as positive.
- TN (True Negative) refers to the data that is predicted as negative and matches the true label as negative.
- FN (False Negative) refers to the data that is predicted as negative but matches the true label as positive.

- FP (False Positive) refers to the data that is predicted as positive but matches the true label as negative.

3.4 Comparative Design of Regularization Techniques

To better understand the behavior of overfitting prevention strategies in multiclass classification tasks, this study was designed to systematically compare Dropout-DNN and Adaptive Regularization-DNN under identical experimental settings. Unlike many prior studies that apply these techniques in isolation or within image-based datasets, this work focuses on structured tabular data with balanced classes, allowing for clearer performance attribution.

Both Dropout and Adaptive Regularization were implemented in the same DNN architecture, using the same optimizer, learning rate, and evaluation metrics. This ensures that any observed differences in performance can be attributed to the regularization method rather than other confounding variables.

4. Experimental Results

This section presents the findings from the conducted experiments aimed at evaluating the effectiveness of various multiclass problem-solving algorithms. The performance of each algorithm was assessed using a comprehensive set of metrics, including Precision, Recall, F1-Score, and Accuracy.

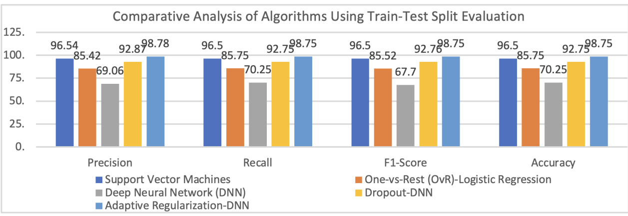

Figure 4. Comparative Analysis of Algorithms using Train-Test Split Evaluation.

4.1 Comparative Analysis of Algorithms using Train–Test Split Method

To evaluate the effectiveness of various algorithms in solving multiclass classification problems, a comparative analysis was conducted using the train–test split approach, in which the dataset was divided into 80% for training and 20% for testing. The algorithms evaluated included SVM, OvR-Logistic Regression, a standard DNN, a DNN with dropout, and the proposed DNN with adaptive regularization. The evaluation focused on key performance metrics including precision, recall, F1-score, and accuracy. The results of this comparative analysis are summarized in Table 1, and Fig. 4, which illustrates the relative performance of each model under consistent experimental conditions.

Table 1. Performance comparison of multiclass classification algorithms using train–test split

Method	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
SVM	96.54	96.50	96.50	96.50
OvR-Logistic Regression	85.42	85.75	85.52	85.75
DNN	69.06	70.25	67.70	70.25
Dropout-DNN	92.87	92.75	92.76	92.75
Adaptive Regularization-DNN (Proposed)	98.78	98.75	98.75	98.75

4.2 Results of Deep Neural Network Algorithm Tuning using Cross-Validation

To enhance the performance of the DNN in solving multiclass classification problems, the researcher conducted algorithm tuning using two regularization techniques: (1) Dropout-DNN, which randomly deactivates a subset of neurons during training to prevent overfitting, and (2) Adaptive Regularization-DNN, which applies a regularization term to the network's loss function to improve generalization. These tuned models were compared against the baseline DNN without regularization.

In addition, the evaluation included two traditional machine learning classifiers—SVM and OvR-Logistic Regression—for benchmarking purposes. All models were evaluated using 5-fold cross-validation to ensure robustness and to minimize bias from a single data split.

The results of the algorithm performance evaluation are summarized in Table 2 and visualized in Fig. 5. The Adaptive Regularization-DNN outperformed all other models in every metric, demonstrating its effectiveness in handling multiclass problems with high precision, recall, F1-score, and accuracy.

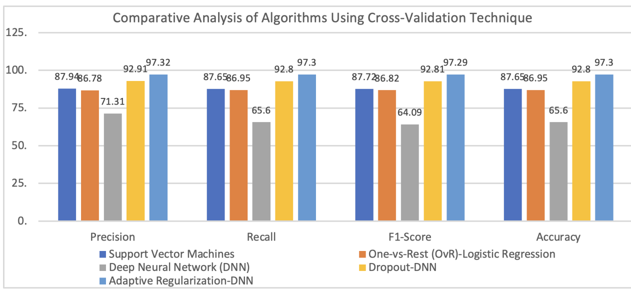

Figure 5. Comparative Analysis of Algorithms Using Cross-Validation Technique.

Table 2. Performance comparison of multiclass classification algorithms using 5-fold cross-validation.

Method	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
SVM	87.94	87.65	87.72	87.65
OvR-Logistic Regression	86.78	86.95	86.82	86.95
DNN	71.31	65.60	64.09	65.60
Dropout-DNN	92.91	92.80	92.81	92.80
Adaptive Regularization-DNN (Proposed)	97.32	97.30	97.29	97.30

4.3 Comparative Performance Analysis and Model Behavior

To gain deeper insights into the classification behavior of each model, confusion matrices were generated and analyzed. These matrices, illustrated in Fig. 6 to Fig. 8, provide a class-level breakdown of correct and incorrect predictions across the four classes (0 to 3) in the Mobile Price Classification dataset in Table 3.

Table 3. Per-class F1-scores.

Class	DNN	Dropout-DNN	Adaptive Regularization-DNN
0	0.98	0.95	0.99
1	0.76	0.91	0.98
2	0.78	0.93	0.98
3	0.99	0.97	0.99

From the confusion matrices, several key observations can be made:

Baseline DNN (see Fig. 6) exhibits frequent misclassifications, particularly between neighboring classes such as class 1 and 2, and class 2 and 3. This reflects the model's limited ability to separate classes with subtle differences.

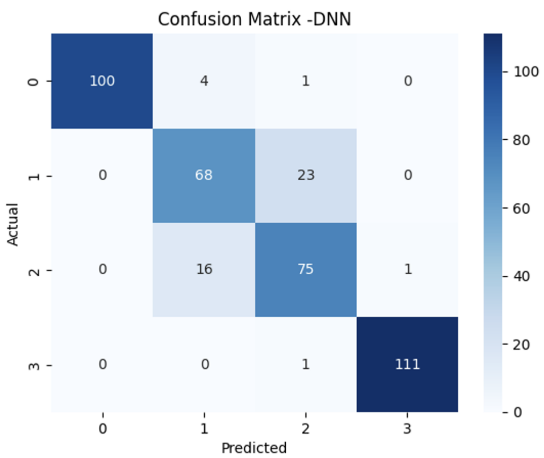

Figure 6. Confusion matrix of Deep Neural Network (DNN).

Dropout-DNN (see Fig. 7) significantly improves classification accuracy. Misclassifications are reduced across all classes, though some overlap remains between classes 2 and 3, which are often close in feature space.

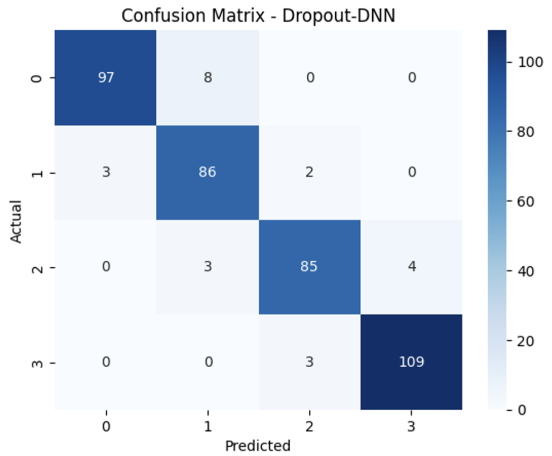

Figure 7. Confusion matrix of Dropout-DNN.

Adaptive Regularization-DNN (see Fig. 8) demonstrates superior performance, with near-perfect classification across all classes. The model exhibits strong generalization with minimal confusion even between adjacent classes.

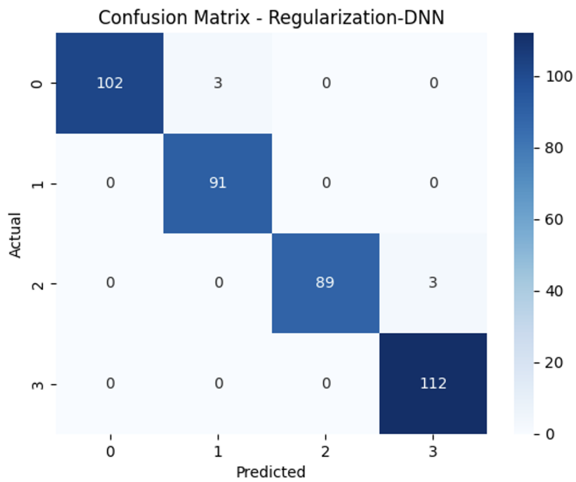

Figure 8. Confusion matrix of Adaptive Regularization-DNN.

4.4 Critical Analysis

The performance of the proposed mobile phone price classification model was critically compared with several existing studies discussed in the literature. Table 4 presents a comparison of eight studies conducted between 2020 and 2025, including the current work. The objective was to evaluate whether the proposed Adaptive Regularization-DNN provides a significant improvement over traditional machine learning and deep learning approaches.

Table 4. Comparison of our best model Adaptive Regularization-DNN

No.	Title	Author	Best Model	Accuracy (%)
1	Classification of Mobile Price using Machine Learning	Sunariya et al. (2024)	SVM	98.00
2	Classification of Mobile Phone Price Dataset using Machine Learning Algorithms	Hu (2022)	SVM	95.50
3	Mobile Phone Price Classification using Machine Learning	Aksoy Ercan and Şimşek (2023)	SVM	96.16
4	Mobile Phone Price Prediction with Feature Reduction	Chen (2023)	Pearson Correlation	95.80
5	Performance Evaluation of Different Supervised Learning Algorithms for Mobile Price Classification	Pipalia and Bhadja (2020)	Gradient Boosting	90.00
6	Comparison of KNN and DNN Classifiers Performance in Predicting Mobile Phone Price Ranges	Güvenç et al. (2021)	DNN	94.00
7	Our Study Baseline		DNN	70.75
8	Our Study		Adaptive Regularization-DNN	98.75

The majority of related works employed SVM as the best-performing model, achieving accuracies ranging from 95.5% to 98%. Notably, the study by Sunariya et al. (2024) reported the highest accuracy at 98% using SVM. Similarly, studies by Hu (2022) and Aksoy Ercan and Şimşek (2023) demonstrated strong performance with SVM, achieving 95.5% and 96.16%, respectively.

Other works explored different techniques such as Gradient Boosting (90% accuracy), Pearson Correlation-based feature reduction (95.8%), and DNN with varying performance. For instance, the study by Güvenç et al. (2021) achieved 94% using DNN, while the baseline model from our own study yielded a lower accuracy of 70.75%.

In contrast, the proposed Adaptive Regularization-DNN in this study outperformed all other approaches, reaching an accuracy of 98.78%, exceeding both traditional machine learning and prior deep learning models. This improvement highlights the strength of integrating regularization techniques into deep learning architectures, particularly when dealing with structured tabular data in multiclass classification problems.

5. Conclusion

This study evaluated and compared the performance of several algorithms for solving multiclass classification problems, including Support Vector Machines (SVM), One-vs-Rest (OvR) Logistic Regression, a baseline Deep Neural

Network (DNN), a Dropout-enhanced DNN, and a Regularization-based DNN. The evaluation was conducted using both train–test split and 5-fold cross-validation to ensure robustness and generalizability of the results. Performance metrics such as accuracy, precision, recall, and F1-score were used to provide a comprehensive assessment.

The results indicate that the Adaptive Regularization-DNN consistently outperformed other methods across all evaluation metrics, achieving an average accuracy of 97.3% in the cross-validation setting. This demonstrates the effectiveness of incorporating regularization in deep neural networks for reducing overfitting and improving generalization. In comparison, SVM and Dropout-DNN also yielded strong performance, with accuracies of 87.65% and 92.8%, respectively while OvR-Logistic Regression and the standard DNN performed less effectively.

These findings highlight the importance of model regularization in deep learning and reinforce the need for structured experimentation in multiclass classification tasks involving tabular data. Although the techniques applied in this study are established, their systematic comparison under controlled settings contributes to a clearer understanding of their relative strengths. Future research could extend this work by exploring ensemble strategies, hybrid models, or adaptive learning techniques to further enhance classification performance in complex multiclass scenarios.

Acknowledgments

This research has been successfully completed with the support of the website www.kaggle.com, which has compiled the data. I would also like to thank the Faculty of Science and Health Technology at Kalasin University for their support in conducting this research.

References

Abramovich, F., Grinshtein, V., and Levy, T. (2021). Multiclass classification by sparse multinomial logistic regression. *IEEE Transactions on Information Theory*, 67(7):4637–4646. DOI: 10.1109/TIT.2021.3075137.

Abuduweili, A., Li, X., Shi, H., Xu, C.-Z., and Dou, D. (2021). Adaptive consistency regularization for semi-supervised transfer learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6919–6928. DOI: 10.1109/CVPR46437.2021.00685.

Aksoy Ercan, S. and Şimşek, M. (2023). Mobile phone price classification using machine learning. *International Journal of Advanced Natural Sciences and En-*

gineering Researches, 7(4):458–462. DOI: 10.59287/i-janser.791.

Bagla, P. and Kumar, K. (2023). Breaking down health fakes: A hybrid DNN model for multi-class classification on a self-constructed dataset. *Sādhanā*, 48(4). DOI: 10.1007/s12046-023-02300-2.

Bejani, M. M. and Ghatee, M. (2021). A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review*, 54(8):6391–6438. DOI: 10.1007/s10462-021-09975-1.

Chen, L., Wang, Y., and Li, H. (2022). Enhancement of DNN-based multilabel classification by grouping labels based on data imbalance and label correlation. *Pattern Recognition*, 132:108964. DOI: 10.1016/j.patcog.2022.108964.

Chen, M. (2023). Mobile phone price prediction with feature reduction. *Highlights in Science, Engineering and Technology*, 34:155–162. DOI: 10.54097/hset.v34i.5440.

Choudhuri, R. and Paul, A. (2021). Multi class image classification for detection of diseases using chest X ray images. In *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 769–773.

Dong, Q., Zhu, X., and Gong, S. (2018). Single-label multi-class image classification by deep logistic regression. *CoRR*, abs/1811.08400.

Feng, S., Zhao, C., and Fu, P. (2020). A deep neural network based hierarchical multi-label classification method. *Review of Scientific Instruments*, 91(2). DOI: 10.1063/1.5141161.

Gao, B.-B. and Zhou, H.-Y. (2021). Learning to discover multi-class attentional regions for multi-label image recognition. *IEEE Transactions on Image Processing*, 30:5920–5932. DOI: 10.1109/TIP.2021.3088605.

Garbin, C., Zhu, X., and Marques, O. (2020). Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimedia Tools and Applications*, 79(19–20):12777–12815. DOI: 10.1007/s11042-019-08453-9.

Guo, Y., Zhang, Z., and Tang, F. (2021). Feature selection with kernelized multi-class support vector machine. *Pattern Recognition*, 117:107988. DOI: 10.1016/j.patcog.2021.107988.

Güvenç, E., Çetin, G., and Koçak, H. (2021). Comparison of knn and dnn classifiers performance in predicting mobile phone price ranges. *Advances in Artificial Intelligence Research*, 1(1):19–28.

Hu, N. (2022). Classification of mobile phone price dataset using machine learning algorithms. In *2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML)*, pages 438–443. DOI: 10.1109/PRML56267.2022.9882236.

Hussain, S. F. and Ashraf, M. M. (2023). A novel one-vs-rest consensus learning method for crash severity prediction. *Expert Systems with Applications*, 228:120443. DOI: 10.1016/j.eswa.2023.120443.

Khan, A. H. and Zubair, M. (2020). Classification of multi-lingual tweets, into multi-class model using naïve bayes and semi-supervised learning. *Multimedia Tools and Applications*, 79(43–44):32749–32767. DOI: 10.1007/s11042-020-09512-2.

Kumar Sahu, S. and Pandey, M. (2023). An optimal hybrid multiclass SVM for plant leaf disease detection using spatial fuzzy C-means model. *Expert Systems with Applications*, 214:118989. DOI: 10.1016/j.eswa.2022.118989.

Lee, S. and Lee, C. (2020). Revisiting spatial dropout for regularizing convolutional neural networks. *Multimedia Tools and Applications*, 79(45–46):34195–34207. DOI: 10.1007/s11042-020-09054-7.

Liu, Y., Yang, S., and Bao, Y. (2024). Deep neural network optimization based on binary method for handling multi-class problems. *IEEE Access*, 12:46881–46890. DOI: 10.1109/access.2024.3382195.

Liu, Y., Zhang, Z., Liu, X., Wang, L., and Xia, X. (2021). Deep learning-based image classification for online multi-coal and multi-class sorting. *Computers & Geosciences*, 157:104922. DOI: 10.1016/j.cageo.2021.104922.

Lv, X., Wu, D., and Xia, S.-T. (2020). Matrix smoothing: A regularization for Dnn with transition matrix under noisy labels. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, page 1–6. IEEE. DOI: 10.1109/icme46284.2020.9102853.

Manita, O. A., Peletier, M. A., Portegies, J. W., Sanders, J., and Senen-Cerda, A. (2022). Universal approximation in dropout neural networks. *Journal of Machine Learning Research*, 23(19):1–46.

Moral, P. D., Nowaczyk, S., and Pashami, S. (2022). Why is multiclass classification hard? *IEEE Access*, 10:80448–80462. DOI: 10.1109/access.2022.3192514.

Ni, X., Fang, L., and Huttunen, H. (2021). Adaptive L2 regularization in person re-identification. In *2020 25th International Conference on Pattern Recognition (ICPR)*, page 9601–9607. IEEE. DOI: 10.1109/icpr48806.2021.9412481.

Panyatip, T., Phothinam, P., and Komgrich, O. (2022). Optimization classification of Covid-19 effect on liver cancer. *Journal of Technology Management Rajabhat Maha Sarakham University*, 9(2):66–78.

Park, J., Jung, Y., and Kim, J.-H. (2022). Multiclass classification fault diagnosis of multirotor UAVs utilizing a deep neural network. *International Journal of Control, Automation and Systems*, 20(4):1316–1326. DOI: 10.1007/s12555-021-0729-1.

Pawar, D. and Dhage, S. (2020). Multiclass covert speech classification using extreme learning machine. *Biomedical Engineering Letters*, 10(2):217–226. DOI: 10.1007/s13534-020-00152-x.

Pawara, P., Okafor, E., Groefsema, M., He, S., Schomaker, L. R., and Wiering, M. A. (2020). One-vs-One classification for deep neural networks. *Pattern Recognition*, 108:107528. DOI: 10.1016/j.patcog.2020.107528.

Pipalia, K. and Bhadja, R. (2020). Performance evaluation of different supervised learning algorithms for mobile price classification. *International Journal for Research in Applied Science and Engineering Technology*, 8(6):1841–1848. DOI: 10.22214/ijraset.2020.6302.

Rice, L., Wong, E., and Kolter, Z. (2020). Overfitting in adversarially robust deep learning. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8093–8104. PMLR.

Sunariya, N., Singh, A., Alam, M., and Gaur, V. (2024). Classification of mobile price using machine learning. In *Symposium on Computing and Intelligent Systems 2024 (SCI 2024)*, pages 55–66.

Tang, L., Tian, Y., and Pardalos, P. M. (2019). A novel perspective on multiclass classification: Regular simplex support vector machine. *Information Sciences*, 480:324–338. DOI: 10.1016/j.ins.2018.12.026.

Thakkar, A. and Lohiya, R. (2021). Analyzing fusion of regularization techniques in the deep learning-based intrusion detection system. *International Journal of Intelligent Systems*, 36(12):7340–7388. DOI: 10.1002/int.22590.

Tiwari, P., Upadhyay, D., Pant, B., and Mohd, N. (2022). *Multiclass Classification of Disease Using CNN and SVM of Medical Imaging*, page 88–99. Springer International Publishing. DOI: 10.1007/978-3-031-12641-3_8.

Vernikou, S., Lyras, A., and Kanavos, A. (2022). Multiclass sentiment analysis on COVID-19-related tweets using deep learning models. *Neural Computing and Applications*, 34(22):19615–19627. DOI: 10.1007/s00521-022-07650-2.

Wang, T., Liu, L., Liu, N., Zhang, H., Zhang, L., and Feng, S. (2020). A multi-label text classification method via dynamic semantic representation model and deep neural network. *Applied Intelligence*, 50(8):2339–2351. DOI: 10.1007/s10489-020-01680-w.

Zehra, W., Javed, A. R., Jalil, Z., Khan, H. U., and Gadekallu, T. R. (2021). Cross corpus multi-lingual speech emotion recognition using ensemble learning. *Complex & Intelligent Systems*, 7(4):1845–1854. DOI: 10.1007/s40747-020-00250-4.

Zhao, H., Tsai, Y. H., Salakhutdinov, R., and Gordon, G. J. (2019). Learning neural networks with adaptive regularization. *CoRR*, abs/1907.06288.

Zhu, Y. and Liu, R. (2021). An algorithmic view of L2 regularization and some path-following algorithms. *Journal of Machine Learning Research*, 22(138):1–62.