IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 April 2023, accepted 8 May 2023, date of publication 11 May 2023, date of current version 18 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3275086

RESEARCH ARTICLE

A Guided Neural Network Approach to Predict Early Readmission of Diabetic Patients

AVISHEK ANISHKAR RAM[1], ZAIN ALI[1], VANDANA KRISHNA[2], NANDITA NISHIKA[1], AND ANURAGANAND SHARMA[1], (Senior Member, IEEE)

[1]School of Information Technology, Engineering, Mathematics and Physics (STEMP), The University of the South Pacific, Suva, Fiji
[2]Department of Medicine, Fiji National University, Suva, Fiji

Corresponding author: Anuraganand Sharma (anuraganand.sharma@usp.ac.fj)

This work was supported by School of Information Technology, Engineering, Mathematics and Physics (STEMP), The University of the South Pacific, Fiji.

ABSTRACT Diabetes is a major chronic health problem affecting millions globally. Effective diabetes management can reduce the risk of hospital readmission and the associated financial losses for both the healthcare system and insurance companies. Hospital readmission is a high-priority healthcare quality measure that reflects the inadequacies in the healthcare system that also increase healthcare costs and negatively influence hospitals' reputation. Predicting readmissions in the early stages prompts great attention to patients with a high risk of readmission. There has been some attempt in applying machine learning predictive models such as ensemble learning with Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM) and Artificial Neural Networks (ANN) to correctly identify if the readmission can happen within 30 days ($<$30 days) or it may never happen or happens after 30 days ($\geq$30 days). We are proposing a new method that is applied to ANN to guide it through its gradient descent optimizers by realizing consistent vs inconsistent data in every batch. Our results show that there are up to 1.5% improvement in classification accuracies in both 2-class and 3-class variations of the experimented benchmark dataset when using the guided optimizer to train the ANN as opposed to the standard optimizer. Guided ANN is also able to achieve better error convergence than standard ANN.

INDEX TERMS Artificial neural network (ANN), hospital readmission, machine learning, error convergence, support vector machine, gradient descent, classification accuracy.

I. INTRODUCTION

The modern Health care industry is increasingly embodying the involvement of Artificial intelligence (AI) in their daily practices. AI has been assisting the health care domain in making decisions on appropriate treatment journeys, analysis of medical reports, making informed clinical decisions, early detection of diseases and many other tasks [1]. Over the years, hospital readmission rate has become a important performance measurement metric for hospitals in order to minimize the impact on healthcare costs and patient outcomes [2]. To improve the performance of health care services, predicting the readmission of a patient has become very important and such that many solutions have been proposed that use AI and Machine Learning techniques to accomplish this [3], [4], [5].

Hospital readmission simply means a patient who had been discharged from a hospital is admitted again within a specified time interval because of the same disease [6]. There could be many causes of readmission, for instance, enough medical care was not provided initially when the patient was admitted to the hospital or subsequent care was not followed properly at home after discharge [7]. Keeping readmission rates low is important as it indicates the quality of health services of a hospital which also makes hospital readmission a major concern, especially in the era of a global pandemic of COVID 19 [8], [9], [10].

Early detection of patients with a high risk of getting readmitted will allow medical professionals to examine and conduct corresponding prevention measures, however, it may

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

lead to an increase in operation cost [11]. Therefore, the correct prediction of readmission to optimize the cost is necessary.

Machine learning techniques have been proven effective for such medical bioinformatics applications and producing quality predictions [3], [4], [6], [7], [8], [12]. Some prominent state-of-the-art models for predicting readmission due to diabetes or other diseases are artificial neural networks (ANN), Support Vector Machine (SVM) and RandomForest (RF) [6], [13], [14], [15]. These algorithms are known for their robustness and good generalization ability, however, SVM is generally not suitable for large datasets and is sensitive to abnormal data distribution or noise whereas RF is more capable in such situations [16], [17]. ANN is also sensitive to abnormal data distribution but its gradient descent process can be strategically maneuvered to compensate for the complex data distribution (also called deterministic noise) [18], [19]. This strategy has been applied in logistic regression in [18] for small datasets but it has never been applied in ANN and for large datasets. Our approach addresses the underlying problem of gradient descent that blindly takes the training data without considering the inconsistency present in the data. The experimented dataset for diabetes readmission is not easy to train due to its size and data distribution. Therefore, this article aims to address these problems by providing a new variant of ANN algorithm based on guided gradient descent to have a better prediction of diabetes readmission. This can improve the quality of hospital services followed by the long-term effect that will be beneficial for hospitals and their patients [20].

Diabetes is a major health issue worldwide with an estimated 1.5 million deaths directly caused by it [21]. Diabetes is a metabolic disorder resulting in hyperglycemia and a plethora of symptoms as a result. Depending on the etiology, it can be classified as diabetes mellitus type I, diabetes mellitus type II, gestational diabetes and other specific type diabetes [22], [23], [24]. This paper will focus on diabetes mellitus type II, henceforth referred as diabetes, as it is the most common form of all [23]. There are multiple factors that increase the risk of a person developing diabetes, some of which include obesity, smoking, sedentary lifestyle, and family history of diabetes. Complications secondary to diabetes make it one of the leading and growing causes of hospital admission and disability [25], [26]. It typically affects 20% of the hospital inpatients [27]. In addition, people with diabetes have hospital admission rates higher than people without diabetes [28]. People with diabetes also have excessive lengths of hospital stay compared to people without diabetes [28]. The economic cost of diabetes mellitus is therefore enormous. Hence, reducing the rate of admissions secondary to diabetes can significantly reduce its global mortality, morbidity and economic burden.

The rest of the paper is organized as follows. Section II describes the machine learning techniques used in the literature to solve the hospital readmission problem. Section III introduces our proposed ANN model in detail and describes the benchmark dataset for hospital readmission for diabetic patients along with the experimental setup used in this paper. Section IV and V demonstrate the findings and describes the experimental results. Section VI concludes the paper with some suggestions for future work.

II. BACKGROUND AND RELATED WORK

A. THE STATE-OF-THE-ART MACHINE LEARNING TECHNIQUES AND READMISSION PREDICTION

Some classic methods to determine risk prediction for hospital readmission include rule-based methods, scores and traditional statistical methods such as logistic regression [13]. Recently, machine learning has emerged as a powerful tool for hospital readmission research. These approaches allow researchers to analyze large datasets and identify complex patterns and relationships between variables. By applying machine learning algorithms to electronic health records, researchers can identify risk factors for readmission or simply predict readmission likelihood for individual patients, and develop personalized intervention strategies to reduce readmission risk [13], [29].

Several studies have demonstrated the effectiveness of machine learning in predicting hospital readmissions. Hosptial readmission can be for variaous diseases such as dental, heart, diabetes, pediatric or even COVID-19 [14], [15], [29], [30], [31]. Mostly ANN, SVM and RF models are considered state-of-the-art techniques [13], [14], [15]. For instance, a study in [6] shows three machine learning models: Random Forest, Naive Bayes and Tree Ensemble were developed for diabetes readmission prediction where the Random Forest was the best performing model. Similarly, studies in [14] and [32] show ANN and SVM perform best respectively.

B. ARTIFICIAL NEURAL NETWORK

Artificial Neural Network (ANN) is a type of model for machine learning(ML) that has become popular and helpful in classification, clustering, pattern recognition and prediction in many disciplines [33]. ANN model is analogous to animal brain cells that process and recognizes the vast amount of data to solve problems. ANN has nodes analogous to neurons in brain cells which are also known as perceptrons that are arranged in a layer or in vectors. ANN has three types of layers that are the input layer, one or more hidden layers and the output layer as shown in FIGURE 1 where the training happens through backpropagation technique [34]. The goal of backpropagation is to modify the weights (vectors) so as to train the neural network to map arbitrary inputs to outputs correctly. The goal is to learn the weights for all linkages in a multi-layered network. The minimum of the error function in weight space is calculated using the method of gradient descent. The resultant weight which offers the minimum error function is the solution of the learning problem. [35]. However, improper optimization techniques may cause the network to reside in the local minima during training without any improvement in reaching an optimal solution [36].

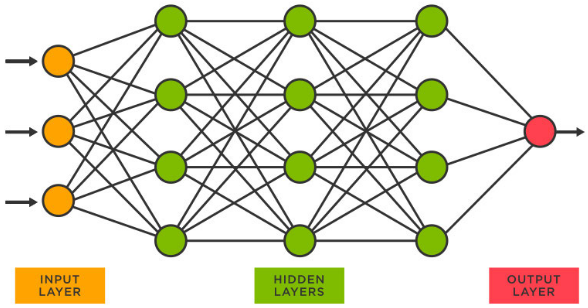


FIGURE 1. Architecture of artificial neural network.

C. ENHANCED ANN THROUGH A VARIETY OF GRADIENT DESCENT TECHNIQUES

There have been many attempts to enhance the ANN through a variety of gradient descent techniques such as standard Stochastic Gradient Descent [37], Adam [38], Adagrad [39], Adadelta/RMSprop [40], Momentum [41] etc. These methods have been used in parallel computations for large datasets as well [18], [42], [43]. These methods exploit the search space to have a better convergence rate to minimize an objective function $J(\theta)$ parameterized by a model's parameters $\theta \in \mathbb{R}^d$ in the opposed direction of the gradient of the objective function $\nabla_x J(\theta) w.r.t$ to the parameters. The learning rate η decides the size of the step to take to reach a (local) minimum [44]. Quite often, the mini-batch gradient descent is favored in training large Deep Neural Network (DNN) models which commonly deal with large datasets, as mini-batch gradient descent allows for fast training time and effective utilization of available computing resources [45]. Mini-batch gradient descent performs an update to the model's parameters θ for every mini-batch of n training samples:

$$\theta = \theta - \eta \cdot \nabla_\theta J\left(\theta; x^{(i:i+n)}; y^{(i:i+n)}\right)$$

while being quite efficient, mini-batch gradient descent can also lead to more stable convergence [44].

Earlier we proposed Guided Stochastic Gradient Descent (GSGD) in [37] for logistic regression - a shallow machine learning model as an improvement over other state-of-the-art variations of gradient descent. GSGD realizes that inconsistent data is a major factor impacting the gradient descent and convergence for SGD, and it tries to overcome this by temporarily hiding them in hopes that they will become consistent in later iterations while GSGD continues to carry out weight updates using the consistent data. Inconsistent data instances simply are the data instances within the neighborhood of instance ρ; which individually performs better, while the average error value $\bar{E}$ performs worse than the average error value of the previous iteration $\bar{E}_{t-1}$, and vice-versa. GSGD has proven to achieve better convergence and classification accuracy than the canonical SGD and its other variations [37].

III. MATERIALS AND METHODS

A. GUIDED ANN FOR BETTER CONVERGENCE

The original GSGD algorithm in [37] and [18] was designed for shallow machine-learning models such as logistic regression. In [46], a variation of GSGD was proposed for Convolution Neural Networks (CNN), a deep learning Model. Based on the tested benchmark datasets, the GSGD for CNN achieves better convergence and improves classification accuracy up to 3% in general when compared to its canonical counterpart.

Our proposed method tries to overcome the deficiency faced by ANN when working with inconsistent datasets by incorporating the GSGD optimizer in a similar approach taken in [46]. The original GSGD [37] cannot be used for large datasets and complex models such as ANN because of its expensive verification step that requires entire training set to determine the data inconsistency in every iteration. The proposed algorithm uses only a subset of training data as verification data to have better convergence. Additionally, we have successfully backpropagated the gradient generated through a guided approach. So the guided approach now works for multi-layer perceptron as opposed to single-layer perception in its original form in [37]. Its flowchart is given in FIGURE 2 where the algorithm starts with a random selection of a data instance whose gradient is computed to update the weight vector. After ρ iterations, the weight vector is further refined with consistent neighboring data instances. It is important to note that the value of ρ must be chosen wisely, as a very large value would result in the algorithm executing in its original form with typical gradient computation and weight update, and GSGD algorithm having very little to no effect in improving its efficiency. For this paper, the value of ρ was selected with Bayesian Hyperparameter Optimization. See Algorithm 1 for the detailed pseudocode of the algorithm.

Original GSGD uses verification data to compute the error of the training data instance at every iteration, which is not applicable to deep learning. The execution of the algorithm begins with the gradient computation and weight update with the learning rate η in the usual manner for the first ρ iterations. Average batch error ($\bar{E}_t$) is computed at every iteration and compared with the average batch error of the previous data batches ($\bar{E}_{t-1}, \bar{E}_{t-2}, \ldots \bar{E}_{t-\rho}$). After the end of p iterations, all data batches performing consistently on weights $W_{t-\rho}$ to W_t is kept in the consistent datastore ψ. Finally, the entire weight vectors are updated with the consistent data batches in ψ. This algorithm proposes to process all data batches regularly up to ρ iterations before reprocessing with only the identified consistent data batches.

B. EXPERIMENT DATA

The dataset used in this experiment was originally presented in the paper [47] and was also added to UCI Library under the name "Diabetes 130-US hospitals for years 1999-2008 Data Set". The dataset has over 50 features representing patient and hospital outcomes with 3 classes: patients who were

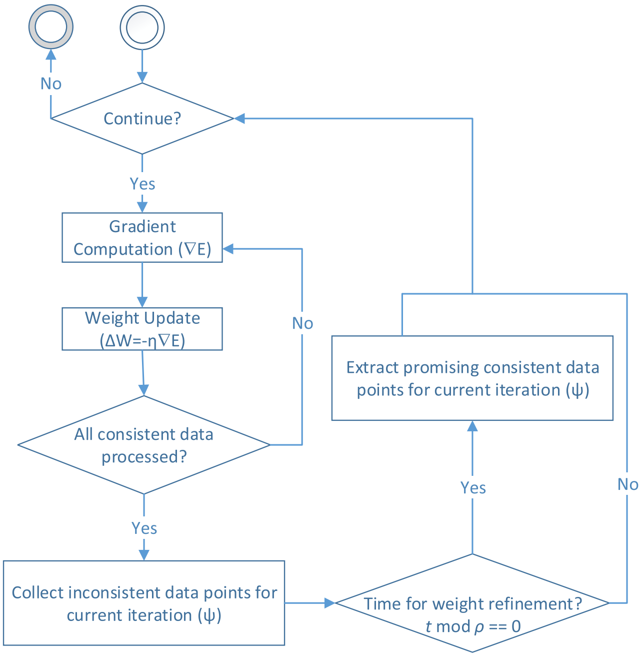

No
Continue?
Yes
Gradient Computation (∇E)
Weight Update (ΔW=-η∇E)
No
All consistent data processed?
Yes
Collect inconsistent data points for current iteration (ψ)
Time for weight refinement? t mod ρ == 0
Yes
Extract promising consistent data points for current iteration (ψ)
No

FIGURE 2. GSGD flowchart.

Algorithm 1 Pseudocode for Guided SGD for ANN

```
// Input: training data set examples
   $d$, total iterations ($T$) and
   neighborhood threshold ($\rho$).
// Algorithm can be tweaked to
   return the best $W$ so far.
gSGD ($d, T, \rho$)
for $t = 1 : T$ do
    compute gradient $v(d_i)$
    $W_t = W_t - \eta v(d_i)$
    $\overline{E}_t$ = approximateAvgError()
    $\psi$ = collectConsistentBatches($d_i, d_{i-1}, \ldots d_{i-\rho}$)
    if $\|\psi\| = \rho$ then
        $\psi$ = getConsistentBatches($\psi$,$\overline{E}_t$)
        for $i = 1 :\| \psi \|$ do
            $W_t = W_t - nv(\psi_i)$
        end
    end
end
return $W_T$
```

TABLE 1. Shows dataset details.

	2-Class Dataset	3-Class Dataset
Train:Test Split	80:20	80:20
Class Distribution	No: 77156 (89.14%) <30: 9399 (10.86%)	No: 46448 (53.66%) <30: 9399 (10.86%) >30: 30708 (35.48%)
Total Features	44	44
Total Records	86,555	86,555

readmitted within 30 days of discharge, patients who were readmitted after 30 days of discharge and patients who had no record of readmittance within the 10-year study period. The dataset contained over 100,000 diabetic patient records.

Two datasets were prepared from the original, the first dataset consisted of the original 3 classes of classifications but for the second dataset, we were interested to identify if a patient will be readmitted within 30 days or not readmitted within 30 days period. Therefore, the "Readmitted after 30 days" class and "no readmission record" class were combined into one, having a total of two classes of classifications for the second dataset.

The datasets were also pre-processed and normalized before conducting the experiment. For patients who had died in between the data collection period, their data was removed. The pre-processed data contained the following attributes such as race, gender, age, admission type, time in hospital, number of lab tests performed, HbA1c test result, diagnosis, number of medications, etc.

After pre-processing, the final dataset had 86,555 records which were split into ratios of 80:20 for the training set and Validation set. The training set was further split into 80:20 to get the validation set. Since the class distribution was imbalanced (89.14% not readmitted and 10.86% readmitted), a commonly used oversampling technique – Synthetic Minority Oversampling Technique (SMOTE) [48] – was applied to synthesize the minority classes in the training set. SMOTE is also considered a "de facto" standard for pre-processing imbalanced data. SMOTE addresses this problem by randomly generating synthetic samples for the minority class through interpolation, which helps to balance the class distribution and improve the overall performance of machine learning models. SMOTE has been used in many application domains such as finance, healthcare and engineering [49], [50], [51]. The detailed description and pseudocode for SMOTE is available in [48], [51], and [52].

C. EXPERIMENTAL SETUP

The ANN architecture setup for this research is described in TABLE 2. The architecture comprises of two hidden layers before the final output layer. After every hidden layer there is a batch normalization layer that deals with the issue of internal covariate shift which is defined as the change in the distribution of network activations due to the change in network parameters during training [53], this issue complicates the training of deep neural networks by requiring lower learning rates and careful parameter initialization. The mini-batch normalization technique will make training artificial neural networks faster and more stable through the normalization of the layers' inputs by re-scaling [53]. After the normalization layer, we have the activation layer that is using sigmoid, a non-linear function for activations in the succeeding layer.

The Experiment was structured as follows: The ANN-GSGD model was developed and trained in Python using the PyTorch library so as all the training and testing

TABLE 2. Architecture of the ANN model.

<table>
<tr><th>Dataset 1:
Diabetes Readmission
(2 classes)</th><th>Dataset 2:
Diabetes Readmission
(3 classes)</th></tr>
<tr><td colspan="2">Input Layer</td></tr>
<tr><td colspan="2">Hidden Linear Layer of 30 Neurons</td></tr>
<tr><td colspan="2">Batch Normalization Layer of 30 Neurons</td></tr>
<tr><td colspan="2">Sigmoid(Nonlinearity function)</td></tr>
<tr><td colspan="2">Hidden Linear Layer of 5 Neurons</td></tr>
<tr><td colspan="2">Batch Normalization Layer of 5 Neurons</td></tr>
<tr><td colspan="2">Sigmoid(Nonlinearity function)</td></tr>
<tr><td colspan="2">Output Layer</td></tr>
</table>

code. The code is available in https://github.com/ECOLS-research-group/GSGD-ANN. Five popular optimizers: RMSProp, Adagrad, SGD, Adam and Adadelta were used in the experiment as the canonical optimizers. The guided variants of these optimizers were also developed by incorporating the proposed guided ANN algorithm. The guided variants will be referred to as G-RMSProp, G-Adagrad, G-SGD, G-Adam, G-Adadelta in this paper. In the experimentation phase, a total of 30 successive runs were carried out to evaluate the Guided ANN model with the Canonical ANN model. These 30 successive runs were carried out for both the datasets and the different optimizers mentioned above.

Bayesian parameter tuning was also carried out to identify the best hyperparameters for the ANN model with different optimizers. TABLE 3 highlights all the hyperparameters that were used in this experiment. All experiments for this research were conducted in a windows-based machine having an $i7 - 11^{th}$ Gen processor and 16 GB of RAM.

In order to compare the performance of our Guided ANN model against other popular machine learning models, we also used our experimental data to train SVM classifier model and Random Forest classifier model, the performance of these models was evaluated using the same validation dataset used for ANN, the results have also been collected and presented in this paper.

IV. RESULTS

A. STATISTICAL ANALYSIS OF CLASSIFICATION ACCURACY

In this section, we provide experimental results of the canonical optimizers and guided optimizers on the Diabetes Readmission datasets of 2 classes and 3 classes. We also analyze and evaluate the performance of each optimizer against their corresponding guided variant.

TABLE 4 highlights the best and average classification accuracy values obtained from running the test datasets on guided ANN. It also shows the Area Under the Receiver Operating Characteristics Curve (AUROC) and the standard deviation of the accuracy values obtained from all the 30 runs carried out in the experiment. We used average AUROC for the 3-class problem. The results show that, on average, the Guided variant performs better than the canonical versions. For the 2-Class dataset, G-Adadelta outperformed the canonical Adadelta by 1.5% and also became one of the best-performing optimizers for the 2-Class dataset. Other guided optimizers also managed to marginally outperform their canonical versions. When looking at the best classification accuracy yielded by the optimizers, both guided and canonical optimizers were able to obtain over 89% accuracy while the guided variant always performed the best; Adam was the only optimizer where the guided and canonical optimizer both obtained the same classification accuracy.

When looking at the classification accuracy results of the 3-class dataset in TABLE 4, it is evident that the guided optimizers outperformed its canonical variants but only marginally. G-RMSProp highlighted by † and RMSProp was the best-performing optimizers overall in the 3-Class dataset when looking at mean results. G-RMSProp managed to outperform its canonical variant by 0.6%.

The Standard Deviation(σ) values in TABLE 4 indicate that the results obtained by Guided optimizers are far less dispersed from the mean when compared to the results obtained by the canonical optimizers.

FIGURE 3 and 4 also highlight the different classification accuracies obtained by the ANN Model throughout the epochs using different optimizers. When analyzing the performance of our prominent optimizer for 2-class highlighted in FIGURE 3(a), it shows that the ANN model had started fitting the data really well by the 10^{th} epoch, and there were steady improvements by both the G-Adalelta and Adadelta on the classification accuracy while G-Adadelta had the upper hand till the last epoch.

Furthermore, our prominent optimizer for the 3-class dataset, G-RMSProp, was also able to have its ANN model fit the data quite well. it also had consistent improvement on the classification accuracy throughout the epochs when compared to its canonical version as shown in FIGURE 4(d). FIGURES 7 and 8 highlight the tradeoff between precision and recall at different thresholds, it is evident from the graph that both the classifier are performing best on the ‘‘Not Readmitted’’ class or the ‘‘No’’ class.

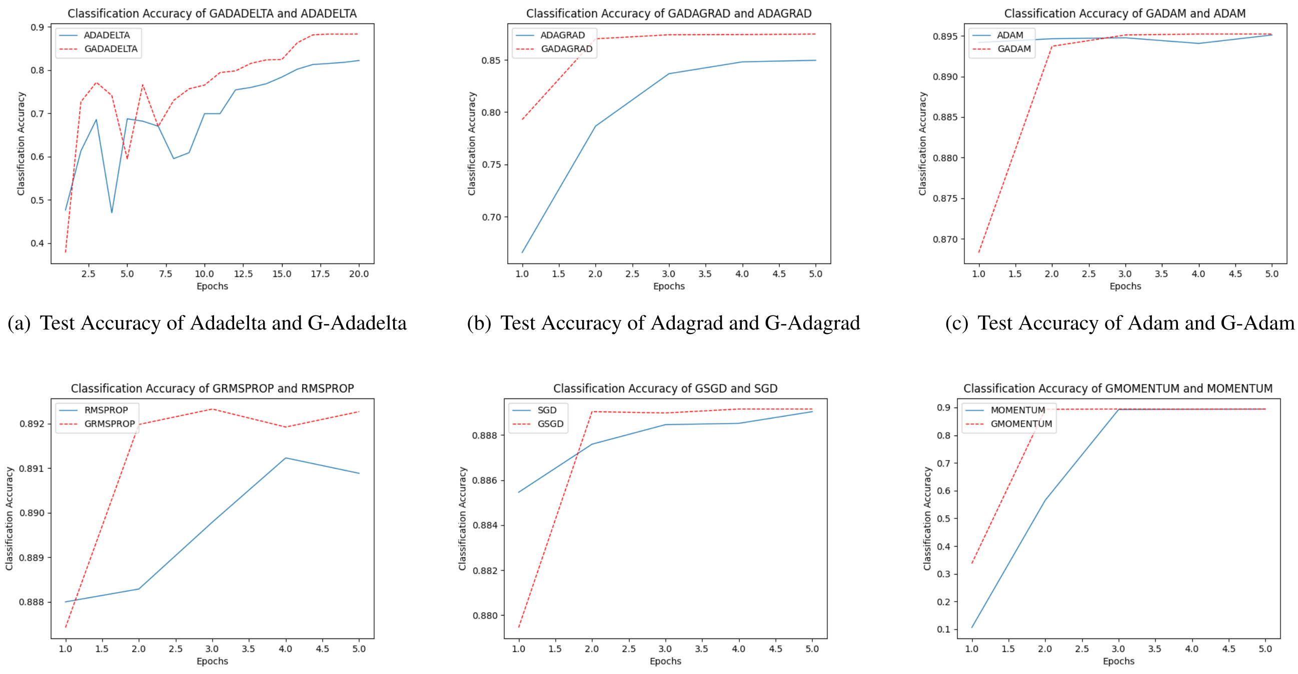
B. CONVERGENCE ANALYSIS

FIGURES 5 and 6 highlight the error convergence of the ANN model using different optimizers. Even though the error convergence rate of both guided and standard ANN is $O(1/T)$ [18], the guided approach shows better convergence in a limited time budget. FIGURE 5(a) highlights the error convergence of our prominent 2-class optimizer throughout the epochs. Both G-Adadelta and Adadelta started attaining stable convergence right after epoch eight but G-Adadelta managed to have a faster convergence than Adadelta.

Error convergence on the 3-class dataset also showed similar patterns as convergence on the 2-Class dataset. G-RMSProp and RMS-Prop were our prominent optimizers for this dataset but G-RMSProp had a much faster convergence than RMSProp.

C. CLASSIFICATION REPORT ANALYSIS

TABLE 5 and TABLE 6 visualizes the classification performance of our prominent optimizers on the 2-class and 3-class



(a) Test Accuracy of Adadelta and G-Adadelta

(b) Test Accuracy of Adagrad and G-Adagrad

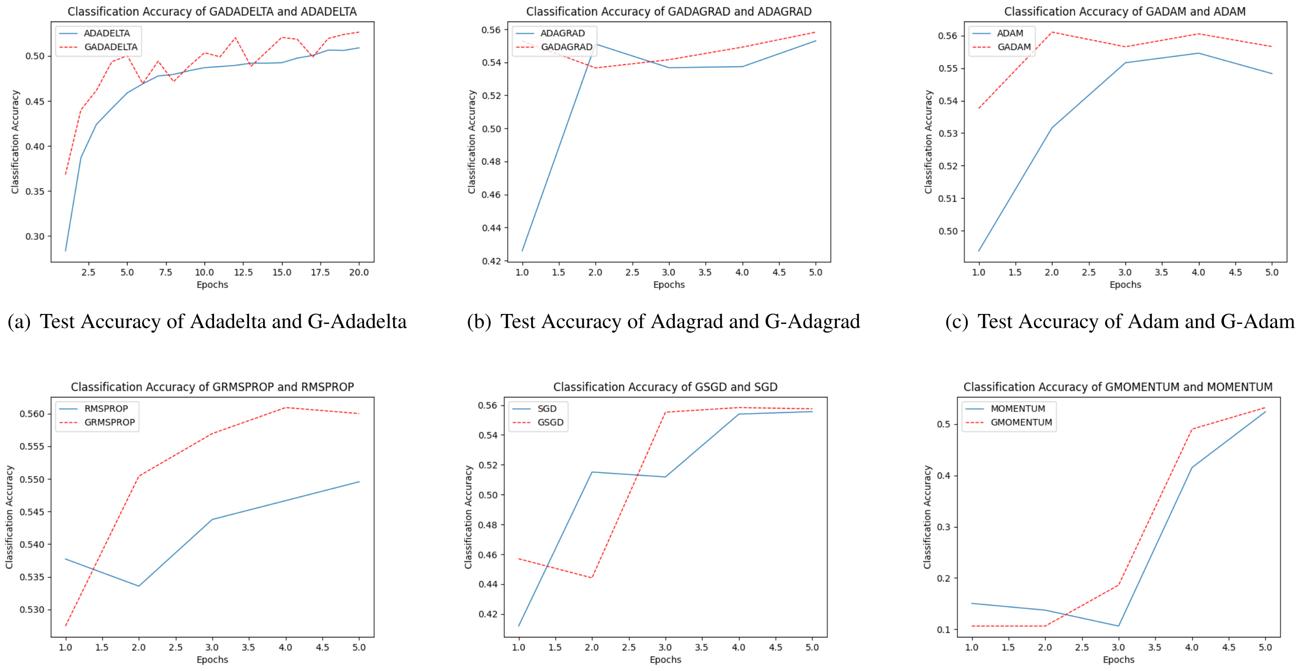
(c) Test Accuracy of Adam and G-Adam

(d) Test Accuracy of RMSProp and G-RMSProp

(e) Test Accuracy of SGD and G-SGD

(f) Test Accuracy of Momentum-SGD and G-Momentum-SGD

FIGURE 3. Validation Accuracy on Diabetes Readmission (2-class) on a scale of 0-1.



(a) Test Accuracy of Adadelta and G-Adadelta

(b) Test Accuracy of Adagrad and G-Adagrad

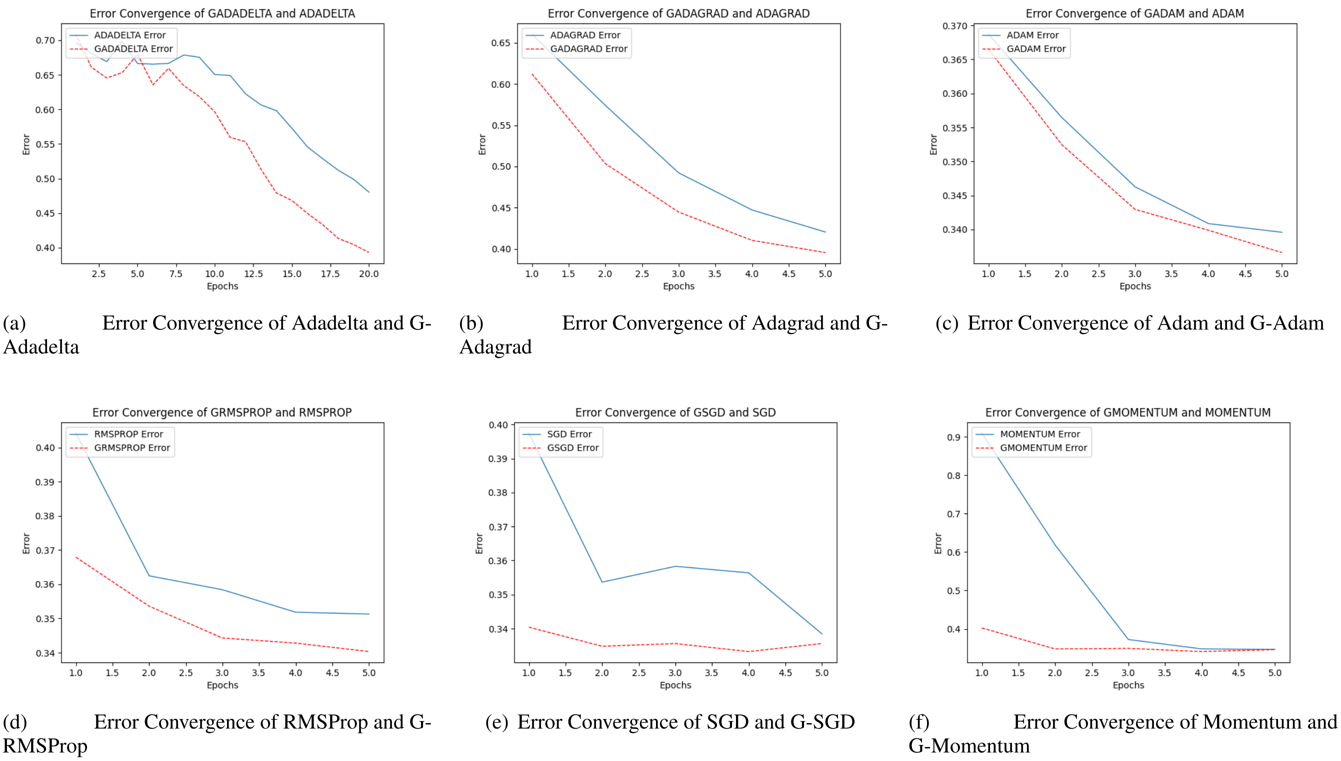
(c) Test Accuracy of Adam and G-Adam

(d) Test Accuracy of RMSProp and G-RMSProp

(e) Test Accuracy of SGD and G-SGD

(f) Test Accuracy of Momentum-SGD and G-Momentum-SGD

FIGURE 4. Validation Accuracy on Diabetes Readmission (3-class) on a scale of 0-1.



(a) Error Convergence of Adadelta and G-Adadelta

(b) Error Convergence of Adagrad and G-Adagrad

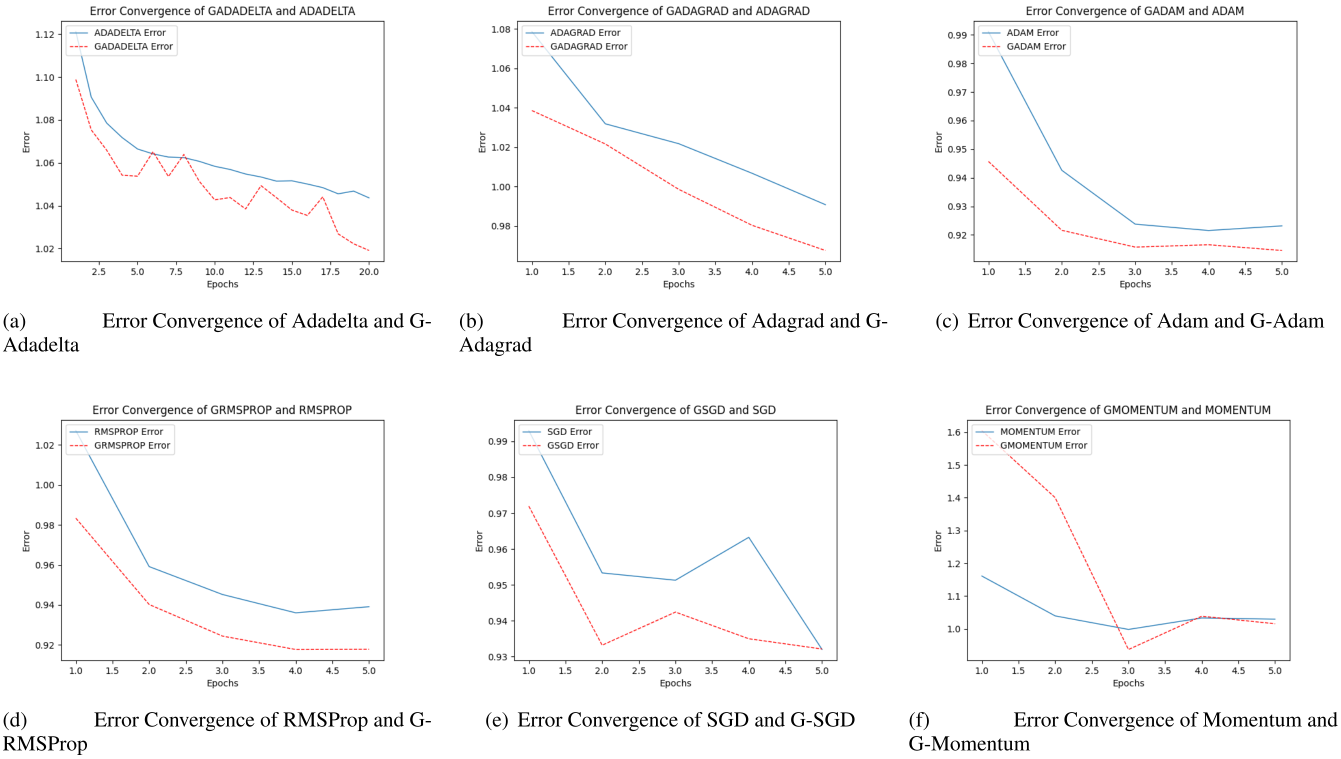
(c) Error Convergence of Adam and G-Adam

(d) Error Convergence of RMSProp and G-RMSProp

(e) Error Convergence of SGD and G-SGD

(f) Error Convergence of Momentum and G-Momentum

FIGURE 5. Error Convergence on Diabetes Readmission (2-class) Validation dataset.



(a) Error Convergence of Adadelta and G-Adadelta

(b) Error Convergence of Adagrad and G-Adagrad

(c) Error Convergence of Adam and G-Adam

(d) Error Convergence of RMSProp and G-RMSProp

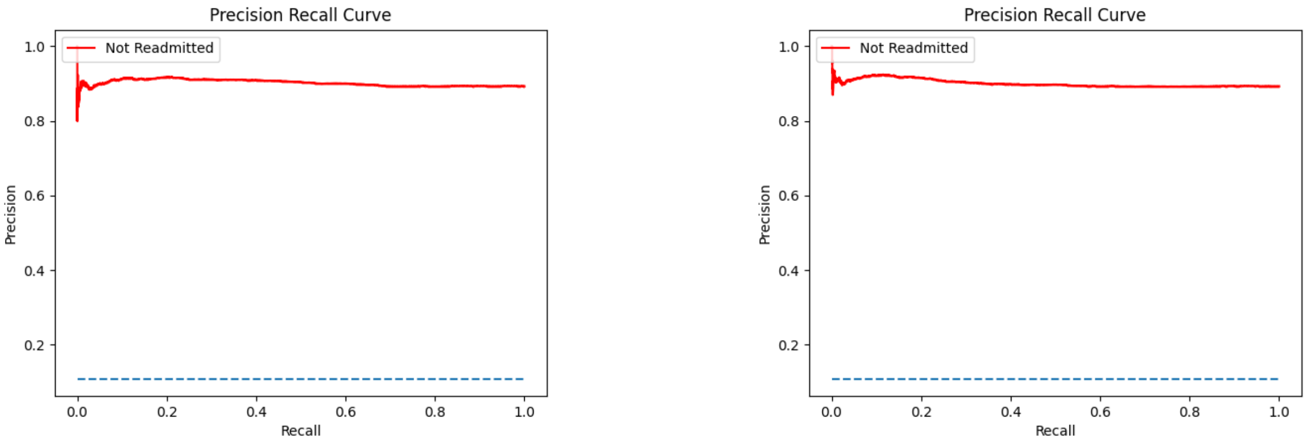
(e) Error Convergence of SGD and G-SGD

(f) Error Convergence of Momentum and G-Momentum

FIGURE 6. Error Convergence on Diabetes Readmission (3-class) Validation dataset.

TABLE 3. Bayesian parameter values for the ANN model where η is the learning rate, λ is the lambda parameter for regularization and ρ is the neighbourhood threshold.

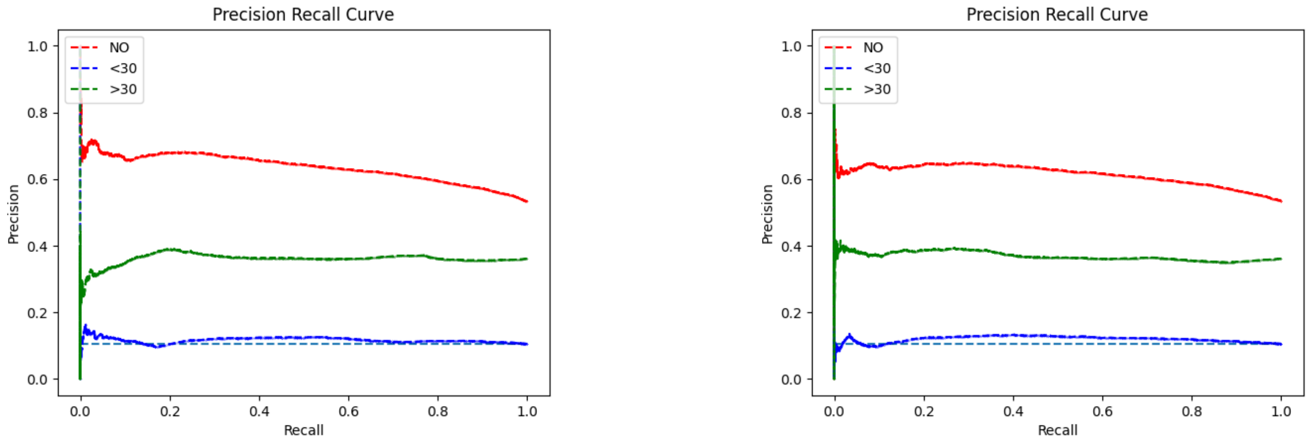
Optimizers	η	λ	Mini-batch Size	Epochs	ρ
SGD/G-SGD	0.18018122514569	1e-06	40	5	20
RMSProp/G-RMSProp	0.003098121514569	1e-06	40	5	20
Adagrad/G-Adagrad	0.02465229341384616	1e-06	40	5	20
Adam/G-Adam	0.0035465229341384616	1e-06	40	5	20
Adadelta/G-Adadelta	0.2008229522561126	1e-06	40	20	20



(a) Precision-Recall Curve for G-Adadelta

(b) Precision-Recall Curve for Adadelta

FIGURE 7. Precision-recall curve for Adadelta and G-Adadelta validated on Diabetes Readmission (2-Class) Dataset.



(a) Precision-Recall Curve for G-RMSProp

(b) Precision-Recall Curve for RMSProp

FIGURE 8. Precision-recall curve for RMSProp and G-RMSProp validated on Diabetes Readmission (3-Class) Dataset.

diabetes readmission datasets. TABLES 5 and 6 present the confusion matrix for the validation dataset, the prediction class 'NOT' indicates that the patient will not be readmitted within 30 days and class 'Readmitted' indicates that the patient will be readmitted within 30 days. Both the classifiers performed really well on the 'NOT' class but not so well on the 'Readmitted' class which is also the minority class. When looking at the visualizations for classification performance on the 3-class dataset, it also shows similar results as the 2-class classifiers. The 3-class classifiers also performed really well on the 'NO' class, but both the classifiers did not perform well on '<30' class while performing adequately on the '>30' class. G-RMSProp performed better than RMSProp on the class '>30'.

It is also important to look into the F1 score, which is the harmonic mean of precision and recall, as the dataset is highly imbalanced. The F1-scores are highlighted in bold in TABLE 7 on a scale of 0 to 1 for all the classes and averages. The best F1 score results have been highlighted in bold.

TABLE 4. Best and average test accuracy for each dataset and optimizer.

Optimizer	Best	Average	Standard Deviation	AUROC
Dataset: Diabetes Readmission (2-Class)				
GSGD	**89.65**	**89.16**	0.24	0.610
SGD	89.45	89.08	0.27	0.600
G-Adadelta	**89.16**	**88.43**†	0.55	0.544
Adadelta	89.06	86.92	2.11	0.540
G-Adagrad	**89.51**	**88.26**	1.22	0.606
Adagrad	89.25	86.81	2.48	0.606
G-Adam	89.55	**89.22**	0.14	0.573
Adam	89.55	89.15	0.31	0.551
G-RMSProp	**89.51**	**89.14**	0.18	0.592
RMSProp	89.44	89.06	0.24	0.560
G-Momentum	89.61	**88.31**	0.54	0.500
Momentum	89.59	87.86	9.79	0.500
Dataset: Diabetes Readmission (3-Class)				
GSGD	**56.82**	**55.69**	2.38	0.547
SGD	56.67	55.15	6.98	0.538
G-Adadelta	**55.09**	**53.39**	0.80	0.564
Adadelta	54.82	53.06	0.88	0.554
G-Adagrad	55.94	**53.35**	1.45	0.553
Adagrad	55.92	52.60	2.01	0.553
G-Adam	**57.06**	**56.22**	0.40	0.559
Adam	56.68	55.74	0.53	0.558
G-RMSProp	**56.74**	**56.25**†	0.52	0.603
RMSProp	56.51	55.65	0.62	0.609
G-Momentum	**55.73**	**54.16**	0.52	0.519
Momentum	55.51	53.75	0.63	0.517

TABLE 5. Table showing confusion matrix for models trained with 2-class diabetes dataset.

Confusion Matrix		Predicted Labels	
		NOT	Readmitted
G-Adadelta True Labels	NOT	82.86%	6.16%
	Readmitted	10.37%	0.62%
Adadelta True Labels	NOT	79.88%	9.13%
	Readmitted	9.87%	1.12%

TABLE 6. Table showing confusion matrix for models trained with 3-class diabetes dataset.

Confusion Matrix		Predicted Labels		
		NO	<30	>30
G-RMSProp True Labels	NO	46.89%	0.21%	6.28%
	<30	7.47%	0.52%	3.01%
	>30	26.83%	0.12%	9.06%
RMSProp True Labels	NO	47.01%	0.55%	5.81%
	<30	7.71%	0.14%	2.69%
	>30	27.49%	0.79%	7.80%

If we consider all classes to be equally important for the experiment, then the macro average becomes the ideal average metric and in the case of 2-class results, the canonical variant managed to outperform the guided variant by 1%, but in the case of 3-class results, the guided variant has managed to outperform the canonical variant by approximately 1%.

Alternatively, the weighted average is preferred where the support values of classes need to be considered when calculating the average F1 score, in the case of our 2-class and 3-class results, the guided variant managed to obtain better weighted average F1 scores than the canonical variants by performing approximately 1% better on both the datasets.

The datasets were also evaluated on SVM and Random Forest classifiers; the results are highlighted in TABLE 8. The hyper-parameter setup for these models were as follows: linear kernel was used for SVM, and gamma was set to scale while the regularizing parameter c was set to 1. In the case of the Random forest model, the estimators (trees) were set to 100 and *'Gini impurity'* criterion was used for ensuring optimum splits. Random Forest and SVM both have shown similar results in comparison to ANN. The Random Forest is one of the best-performing classifiers for the Diabetes Readmission dataset, which has also been discussed in [6]. Guided-ANN managed to outperform Random Forest Classifiers on all four of the performance metrics: Accuracy, AUROC, Macro Average f1-score and Weighted Average f1-score for the 2-class dataset while only falling short on the Macro Average f1-score performance metric for the 3-class dataset. Compared to the SVM classifier, Guided-ANN outperformed it in all the performance metrics, and on the case of the 3-class dataset, Guided-ANN outperformed SVM on Weighted Average f1-score by more than 2%.

V. DISCUSSION

A significant improvement in the quality of classification has been observed with the introduction of guided stochastic gradient descent on training Artificial Neural Networks. It has clearly outperformed its canonical counterpart by obtaining approximately 1.0 to 1.5% better classification accuracy on both the 2-class and 3-class datasets. Adadelta and its guided variant and RMSProp and its guided variant proved to be the most prominent optimizers in the experiment. The confusion matrix for the results also shows that the performance of the guided optimizer and its canonical version is quite similar. The low dispersion of classification accuracy results shown by the standard deviation obtained by guided optimizers indicates that they can provide consistent accuracy for classification.

Since the dataset is highly imbalanced, we also considered the F1 scores from the classification report of the experiment. Guided optimizer generally performed better than its canonical counterpart on both the weighted average F1 scores and the macro average F1 scores with the exception of the macro average F1 score of Adadelta on the 2-class dataset which was off by approximately 1%. The overall experiment shows that the ANN model trained with a guided optimizer performs better than the ANN model that has been trained with canonical optimizers. As expected, the guided approach shows improvement due to its handling of training data. It takes the 'consistent' samples to make the gradient move closer to the true gradient in every iteration as shown in [37] too. The precision-recall curve for the 2-class and 3-class datasets in FIGURE 7 and FIGURE 8 together with AUROC values in TABLE 4 shows that the data distribution of classes is quite indistinguishable. Even in such conditions, the guided approach provides better classification results. We also compared the performance of Guided-ANN against the Random forest classifier and SVM classifier and it also managed to outperform them.

TABLE 7. Classification report for two of the best-performing optimizers (Adadelta and RMSProp) on the diabetes dataset.

Classification Report		Guided				Canonical			
		precision	recall	f1-score	support	precision	recall	f1-score	support
Diabetes Readmission(2-Class) Adadelta	Not Readmitted	0.888779	0.930824	**0.909316**†	15410	0.890069	0.897404	0.893722†	15410
	micro avg	0.834739	0.834739	**0.834739**	17312	0.810016	0.810016	0.810016	17312
	macro avg	0.489999	0.49354	0.489455	17312	0.499682	0.499701	**0.499621**	17312
	weighted avg	0.801154	0.834739	**0.817059**	17312	0.804288	0.810016	0.807125	17312
	samples avg	0.834739	0.834739	**0.834739**	17312	0.810016	0.810016	0.810016	17312
Diabetes Readmission(3-Class) RMSProp	NO	0.577558	0.878368	**0.696888**†	9241	0.57184	0.880749	0.693448†	9241
	<30	0.113924	0.004934	0.009459†	1824	0.093385	0.013158	**0.023066**†	1824
	>30	0.493551	0.251161	**0.332909**†	6247	0.478738	0.216264	0.297938†	6247
	micro avg	0.560016	0.560016	**0.560016**	17312	0.549561	0.549561	0.549561	17312
	macro avg	0.395011	0.378154	**0.346418**	17312	0.381321	0.370057	0.338151	17312
	weighted avg	0.498396	0.560016	**0.493119**	17312	0.487834	0.549561	0.480097	17312
	samples avg	0.560016	0.560016	**0.560016**	17312	0.549561	0.549561	0.549561	17312

TABLE 8. Table showing performance of other machine learning classifiers against Guided-ANN.

Dataset Group	Classifier	Accuracy	AUROC	Macro Average f1-score	Weighted Average f1-score
2-Class	Random Forest	88.78	0.501	47.33	83.70
	SVM	88.90	0.502	47.06	83.69
	Guided-ANN	**89.16**	0.610	**47.37**	**84.29**
3-Class	Random Forest	55.27	0.541	**37.58**	51.19
	SVM	56.25	0.536	35.43	49.46
	Guided-ANN	**56.74**	0.559	36.39	**51.39**

VI. CONCLUSION

We presented a Guided Artificial Neural Network model that uses the guided stochastic gradient descent algorithm to train the classifier. Our results show that the Guided Artificial Neural Network performs better than its canonical counterpart by obtaining at least 1% to 1.5% better classification accuracy on the 2-class and 3-class Diabetes readmission dataset, while also achieving faster convergence. Guided ANN is also able to achieve relatively good F1 scores in comparison to the canonical variant for the same dataset. Guided ANN has shown that it can significantly improve the classification accuracy and overall performance of the model in the prediction of early readmissions of patients due to diabetes, there are various benefits of being able to predict hospital readmissions such as reducing the strain on the health care resources, improving patient care [20].

There are several possible future directions from this work such as the application of GSGD in classification tasks in other medical domains or the application of GSGD to Recurrent Neural Networks (RNN). Since the Diabetes Readmission was a highly imbalanced dataset, it will also be interesting to investigate if other oversampling techniques such as SMOTified-GAN [52] can counter the effects of data imbalance and further improve the Guided ANN classifier model as opposed to the standard SMOTE technique [48] that was carried out in this research.

CODE AND DATA

The authors provide Python code and data for extending this work further.[1]

[1] https://github.com/ECOLS-research-group/GSGD-ANN

REFERENCES

[1] R. Kejriwal and Mohana, "Artificial intelligence (AI) in medicine and modern healthcare systems," in *Proc. Int. Conf. Augmented Intell. Sustain. Syst. (ICAISS)*, Nov. 2022, pp. 25–31.

[2] S. F. Jencks, M. V. Williams, and E. A. Coleman, "Rehospitalizations among patients in the medicare fee-for-service program," *New England J. Med.*, vol. 360, no. 14, pp. 1418–1428, Apr. 2009, doi: 10.1056/NEJMsa0803563.

[3] W. Liu, C. Stansbury, K. Singh, A. M. Ryan, D. Sukul, E. Mahmoudi, A. Waljee, J. Zhu, and B. K. Nallamothu, "Predicting 30-day hospital readmissions using artificial neural networks with medical code embedding," *PLoS ONE*, vol. 15, no. 4, pp. 1–12, Apr. 2020, doi: 10.1371/journal.pone.0221606.

[4] S. Davis, J. Zhang, I. Lee, M. Rezaei, R. Greiner, F. A. McAlister, and R. Padwal, "Effective hospital readmission prediction models using machine-learned features," *BMC Health Services Res.*, vol. 22, no. 1, p. 1415, Nov. 2022, doi: 10.1186/s12913-022-08748-y.

[5] A. Ashfaq, A. Sant'Anna, M. Lingman, and S. Nowaczyk, "Readmission prediction using deep learning on electronic health records," *J. Biomed. Informat.*, vol. 97, Sep. 2019, Art. no. 103256. https://www.sciencedirect.com/science/article/pii/S1532046419301753

[6] Y. Shang, K. Jiang, L. Wang, Z. Zhang, S. Zhou, Y. Liu, J. Dong, and H. Wu, "The 30-days hospital readmission risk in diabetic patients: Predictive modeling with machine learning classifiers," *BMC Med. Informat. Decis. Making*, vol. 21, no. S2, p. 57, Jul. 2021.

[7] I. B. Seraphim, V. Ravi, and A. Rajagopal, "Prediction of diabetes readmission using machine learning," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 6, p. 9, 2020.

[8] A. Hammoudeh, G. Al-Naymat, I. Ghannam, and N. Obied, "Predicting hospital readmission among diabetics using deep learning," *Proc. Comput. Sci.*, vol. 141, pp. 484–489, 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050918317873

[9] J. Benbassat and M. Taragin, "Hospital readmissions as a measure of quality of health care: Advantages and limitations," *Arch. International Med.*, vol. 160, no. 8, pp. 1074–1081, Apr. 2000, doi: 10.1001/archinte.160.8.1074.

[10] C. Fischer, H. F. Lingsma, P. J. Marang-van de Mheen, D. S. Kringos, N. S. Klazinga, and E. W. Steyerberg, "Is the readmission rate a valid quality indicator? A review of the evidence," *PLoS ONE*, vol. 9, no. 11, Nov. 2014, Art. no. e112282. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4224424/

[11] S. Upadhyay, A. L. Stephenson, and D. G. Smith, "Readmission rates and their impact on hospital financial performance: A study of Washington hospitals," *Inquiry, A J. Med. Care Org., Provision Financing*, vol. 56, Jul. 2019, Art. no. 0046958019860386. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6614936/

[12] C. Reid. (2019). *Diabetes Diagnosis and Readmission Risks Predictive Modelling: USA*. [Online]. Available: https://norma.ncirl.ie/4106/

[13] K. Teo, C. W. Yong, J. H. Chuah, Y. C. Hum, Y. K. Tee, K. Xia, and K. W. Lai, "Current trends in readmission prediction: An overview of approaches," *Arabian J. Sci. Eng.*, pp. 11–18, Aug. 2021, doi: 10.1007/s13369-021-06040-5.

[14] W. Li, M. S. Lipsky, E. S. Hon, W. Su, S. Su, Y. He, R. Holubkov, X. Sheng, and M. Hung, ''Predicting all-cause 90-day hospital readmission for dental patients using machine learning methods,'' *BDJ Open*, vol. 7, no. 1, pp. 1–7, Jan. 2021. [Online]. Available: https://www.nature.com/articles/s41405-021-00057-6

[15] P. Wolff, M. Graña, S. A. Ríos, and M. B. Yarza, ''Machine learning readmission risk modeling: A pediatric case study,'' *BioMed Res. Int.*, vol. 2019, Art. no. 8532892, Apr. 2019, doi: 10.1155/2019/8532892.

[16] K. Fawagreh, M. M. Gaber, and E. Elyan, ''Random forests: From early developments to recent advancements,'' *Syst. Sci. Control Eng.*, vol. 2, no. 1, pp. 602–609, Dec. 2014, doi: 10.1080/21642583.2014.956265.

[17] M. Awad and R. Khanna, ''Support vector machines for classification,'' in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, M. Awad and R. Khanna, Eds. New York, NY, USA: Apress, pp. 39–66, doi: 10.1007/978-1-4302-5990-9_3.

[18] A. Sharma, ''Guided parallelized stochastic gradient descent for delay compensation,'' *Appl. Soft Comput.*, vol. 102, Apr. 2021, Art. no. 107084.

[19] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data*. USA: AMLBook, 2012.

[20] D. J. Rubin, ''Hospital readmission of patients with diabetes,'' *Current Diabetes Rep.*, vol. 15, no. 4, p. 17, Apr. 2015. [Online]. Available: http://link.springer.com/10.1007/s11892-015-0584-7

[21] (2022). *Diabetes*. [Online]. Available: https://www.who.int/health-topics/diabetes

[22] S. E. Inzucchi and B. Lupsa. (2022). *Clinical Presentation, Diagnosis, and Initial Evaluation of Diabetes Mellitus in Adults*. [Online]. Available: https://www.uptodate.com/contents/clinical-presentation-diagnosis-and-initial-evaluation-of-diabetes-mellitus-in-adults

[23] V. Kumar, K. A. Abbas, and C. J. Aster, ''Endocrine system,'' in *Robbins Basic Pathology*, 10th ed. Canada: Elsevier, 2018, pp. 749–796. [Online]. Available: https://www.elsevier.com/books/robbins-basic-pathology/kumar/978-0-323-35317-5

[24] American Diabetes Association, ''Diagnosis and classification of diabetes mellitus,'' *Diabetes Care*, vol. 37, no. 1, pp. S81–S90, Dec. 2013, doi: 10.2337/dc14-S081.

[25] A. Naslafkih and F. Sestier, ''Diabetes mellitus related morbidity, risk of hospitalization and disability,'' *J. Insurance Med.*, vol. 35, no. 2, pp. 102–113, 2003.

[26] P. Zimmet, K. G. Alberti, D. J. Magliano, and P. H. Bennett, ''Diabetes mellitus statistics on prevalence and mortality: Facts and fallacies,'' *Nature Rev. Endocrinology*, vol. 12, no. 10, pp. 616–622, Oct. 2016.

[27] J. R. Petrie and J. G. Boyle, ''Diabetes mellitus,'' in *Davidson's Principles and Practice of Medicine*, 24th ed. Amsterdam, The Netherlands: Elsevier, 2022, pp. 703–753. [Online]. Available: https://www.elsevier.com/books/davidson's-principles-and-practice-of-medicine/978-0-7020-8347-1

[28] E. J. Comino, M. F. Harris, M. F. Islam, D. T. Tran, B. Jalaludin, L. Jorm, J. Flack, and M. Haas, ''Impact of diabetes on hospital admission and length of stay among a general population aged 45 year or more: A record linkage study,'' *BMC Health Services Res.*, vol. 15, no. 1, p. 12, Jan. 2015.

[29] O. Ben-Assuli and R. Padman, ''Analysing repeated hospital readmissions using data mining techniques,'' *Health Syst.*, vol. 7, no. 2, pp. 120–134, 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6452839/

[30] P. Michailidis, A. Dimitriadou, T. Papadimitriou, and P. Gogas, ''Forecasting hospital readmissions with machine learning,'' *Healthcare*, vol. 10, no. 6, p. 981, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9222500/

[31] E. B. Uitvlugt, M. J. A. Janssen, C. E. H. Siegert, E. L. Kneepkens, B. J. F. van den Bemt, P. M. L. A. van den Bemt, and F. Karapinar-Çarkit, ''Medication-related hospital readmissions within 30 days of discharge: Prevalence, preventability, type of medication errors and risk factors,'' *Frontiers Pharmacol.*, vol. 12, Apr. 2021, Art. no. 567424. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fphar.2021.567424

[32] B. Zheng, J. Zhang, S. W. Yoon, S. S. Lam, M. Khasawneh, and S. Poranki, ''Predictive modeling of hospital readmissions using metaheuristics and data mining,'' *Exp. Syst. Appl.*, vol. 42, no. 20, pp. 7110–7120, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417415003085

[33] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, ''State-of-the-art in artificial neural network applications: A survey,'' *Heliyon*, vol. 4, no. 11, Nov. 2018, Art. no. e00938. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405844018332067

[34] Hecht-Nielsen, ''Theory of the backpropagation neural network,'' in *Proc. Int. Joint Conf. Neural Netw.*, vol. 1, 1989, pp. 593–605.

[35] J. Amrutha and A. S. Remya Ajai, ''Performance analysis of backpropagation algorithm of artificial neural networks in verilog,'' in *Proc. 3rd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2018, pp. 1547–1550. [Online]. Available: https://ieeexplore.ieee.org/document/9012614/

[36] E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala, and C. O. Aigbavboa, ''A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks,'' in *Proc. Int. Conf. Comput. Techn., Electron. Mech. Syst. (CTEMS)*, Dec. 2018, pp. 92–99. [Online]. Available: https://ieeexplore.ieee.org/document/8769211/

[37] A. Sharma, ''Guided stochastic gradient descent algorithm for inconsistent datasets,'' *Appl. Soft Comput.*, vol. 73, pp. 1068–1080, Dec. 2018.

[38] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*.

[39] J. Duchi, E. Hazan, and Y. Singer, ''Adaptive subgradient methods for online learning and stochastic optimization,'' *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011. [Online]. Available: http://jmlr.org/papers/v12/duchi11a.html

[40] M. D. Zeiler, ''ADADELTA: An adaptive learning rate method,'' 2012, *arXiv:1212.5701*.

[41] N. Qian, ''On the momentum term in gradient descent learning algorithms,'' *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608098001166

[42] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, ''Large scale distributed deep networks,'' in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, vol. 1. Lake Tahoe, NV, USA: Curran Associates, 2012, pp. 1223–1231. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999271

[43] S. Zheng, Q. Meng, T. Wang, W. Chen, N. Yu, Z. Ma, and T. Liu, ''Asynchronous stochastic gradient descent with delay compensation,'' in *Proc. Int. Conf. Mach. Learn.*, Aug. 2017, pp. 4120–4129. [Online]. Available: http://proceedings.mlr.press/v70/zheng17b.html

[44] S. Ruder, ''An overview of gradient descent optimization algorithms,'' 2016, *arXiv:1609.04747*.

[45] O. K. Oyedotun, K. Papadopoulos, and D. Aouada, ''A new perspective for understanding generalization gap of deep neural networks trained with large batch sizes,'' 2022, *arXiv:2210.12184*.

[46] P. Sharma, A. K. Sharma, D. Kumar, and A. Sharma, ''A strategic weight refinement maneuver for convolutional neural networks,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–7.

[47] B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore, ''Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records,'' *BioMed Res. Int.*, vol. 2014, pp. 1–11, Apr. 2014. [Online]. Available: http://www.hindawi.com/journals/bmri/2014/781670/

[48] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ''SMOTE: Synthetic minority over-sampling technique,'' *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[49] L. Wang, ''Imbalanced credit risk prediction based on SMOTE and multi-kernel FCM improved by particle swarm optimization,'' *Appl. Soft Comput.*, vol. 114, Jan. 2022, Art. no. 108153. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621010176

[50] M. Waqar, H. Dawood, H. Dawood, N. Majeed, A. Banjar, and R. Alharbey, ''An efficient SMOTE-based deep learning model for heart attack prediction,'' *Sci. Program.*, vol. 2021, Mar. 2021, Art. no. 6621622, doi: 10.1155/2021/6621622.

[51] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, ''SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary,'' *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, Apr. 2018.

[52] A. Sharma, P. K. Singh, and R. Chandra, ''SMOTified-GAN for class imbalanced pattern classification problems,'' *IEEE Access*, vol. 10, pp. 30655–30665, 2022.

[53] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 448–456.

AVISHEK ANISHKAR RAM received the B.Sc. degree in computing science and information systems from The University of the South Pacific, Fiji, where he is currently pursuing the master's degree in computing science. From 2020 to 2021, he was a Software Developer with QIT Pacific, working on projects such as incident management systems and enterprise management systems. He is currently a Software Engineer. He is also a Systems Developer with the Department of Research and Education Enabling Technology, Fiji National University, where he works on automation projects and learning management systems. His research interests include artificial intelligence are in deep learning and optimization.

ZAIN ALI received the bachelor's degree in software engineering from USP, Fiji, in 2022. He is currently pursuing the master's degree in computer science on the application of a BERT-based CNN model with multi-domain data fusion in the analysis of Tweets to predict depression in users. He is a currently Software Engineer. He worked on several development projects, such as a price recommender systems and automation dashboards for multi-state agencies. His research interests include artificial intelligence, mainly in natural language processing and forecasting in finance.

VANDANA KRISHNA received the B.Sc. (Hons.), the Bachelor of Medicine, and the Bachelor of Surgery (Hons.) degrees from The University of the South Pacific, Fiji, the master's degree in emergency medicine from Fiji National University, Fiji, and the M.Sc. degree (Hons.) in medical education from the University of South Wales, U.K.

Since 2011, she has been practicing as a medical practitioner in Fiji. Since 2016, she also started her career in the academic field teaching clinical skills to MBBS students with Fiji National University. Since 2021, she has been involved in teaching postgraduate students of medical education with the University of South Wales.

NANDITA NISHIKA received the bachelor's degree in software engineering, in 2021. She is currently pursuing the master's degree in computing science on the enhancement of GAN model for imbalanced classification problems with The University of the South Pacific, Fiji. She is currently a Software Engineer. She is also a Software Analyst Programmer with Datec Fiji Ltd., with two years of experience in application development. Her research interests include artificial engineering, mostly in the economic segments.

ANURAGANAND SHARMA (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from The University of the South Pacific, Fiji, and the Ph.D. degree in artificial intelligence titled "Information discovery from constraints for evolutionary computational models" from the University of Canberra, Australia, in 2014.

From 2003 to 2005, he was a Software Developer with Colonial, Fiji. Since 2007, he has been an Assistant Lecturer to a Senior Lecturer of computer science with The University of the South Pacific. Currently, he is the DHoS (research) with the School of Information Technology, Engineering, Mathematics and Physics (STEMP). His research interests include deep learning with CNN, constraint optimization with meta-heuristic algorithms, and optimization and enhancement of SGD for gradient-based learning systems.