An effective learning rate scheduler for stochastic gradient descent-based deep learning model in healthcare diagnosis system

K. Sathyabama* and K. Saruladha

Department of CSE,
Pondicherry Engineering College,
Puducherry, India
Email: sathii_manju@pec.edu
Email: charuladha@pec.edu
*Corresponding author

Abstract: This study develops an effective stochastic gradient descent (SGD) and time with exponential decay (TED)-based learning rate scheduler called SGD-TED model for deep learning-based healthcare diagnosis. The presented SGD-TED model involves pre-processing, classification, SGD-based parameter tuning and TED-based learning rate scheduling. Once the data is pre-processed, three DL models namely recurrent neural network (RNN), long short-term memory (LSTM) and gated recurrent unit (GRU) are used for diagnosis. Then, the hyperparameter tuning takes place by SGD and TED is applied to schedule the learning rate proficiently. The application of SGD-TED approach in the DL models considerably helps to increase the classification performance. The effectiveness of the SGD-TED model is assessed on three benchmark medical dataset and the experimental outcome ensured that the SGD-TED-LSTM model has resulted to a higher accuracy of 98.59%, 93.68% and 95.20% on the applied diabetes, EEG Eye State and sleep stage dataset.

Keywords: deep learning; stochastic gradient descent; SGD; learning rate scheduler; long short-term memory; LSTM; healthcare; time with exponential decay; TED; recurrent neural network; RNN; gated recurrent unit; GRU.

Reference to this paper should be made as follows: Sathyabama, K. and Saruladha, K. (2022) ‘An effective learning rate scheduler for stochastic gradient descent-based deep learning model in healthcare diagnosis system’, *Int. J. Electronic Healthcare*, Vol. 12, No. 1, pp.1–21.

Biographical notes: K. Sathyabama received her MTech (Information Security) in Computer Science and Engineering from Pondicherry Engineering College, Pondicherry, India. She is currently pursuing her PhD in Computer Science and Engineering at Pondicherry Engineering College, Pondicherry, India. Her research interests include data mining, machine learning and deep learning. She has published in three international journals and two international conference.

K. Saruladha completed her BE in Computer Science and Engineering at University of Madras at 1989 and MTech under Pondicherry University in the year 1997. She completed her PhD in the year 2012 in the area of Semantic similarity measures for ontology based information retrieval systems. She is recognised supervisor in Pondicherry University. She has been special session

organiser on information retrieval in international conferences. She has organised several faculty development programmes sponsored national funding agencies like AICTE,TEQIP. She has also served as resource person for Short Term Training Programmes sponsored by AICTE, TEQIP, QIP.

1 Introduction

Recently, healthcare sector is one of the most important fields in bio-medicinal data. For example, precision medicine aims to provide the correct treatment for the patient as soon as it is needed under different features of patient's information, like living habits, electronic health record (EHR), atmosphere and variations in molecular tests. The substantial availability of bio-medicinal data has the tendency to create high challenges in the healthcare studies. Besides, searching the relation among other data sets has turned into the fundamental problem of deploying secure medicinal gadgets based on database technique and machine learning (ML). For the purpose of attaining the high performance, a variety of conventional techniques are tried to link huge data sources for developing joint knowledge databases that may be used in prediction tasks and identification (Xu et al., 2014). Although the earlier techniques represent the crucial problems, prognostic gadgets related on ML techniques have exploited the medicinal sources (Bellazzi and Zupan, 2008).

Clearly, the whole utilisation of bio-medicinal data is treated as a most difficult features which likely to have irregularities, temporary dependencies, heterogeneities, sparsity and high dimensionalities (Hripcsak and Albers, 2013). Therefore, these challenges are the reason for further complexity along with various medicinal ontologies and it is mostly used for data generalisation like International Classification of Disease-9th version (ICD-9), unified medical language system (UMLS), systematised nomenclature of medicine-clinical terms (SNOMED-CT). Traditional data mining and statistical learning methods commonly required to carry out the feature engineering for obtaining proficient and robust features from those data, and then build prediction or clustering models on top of them. There are lots of challenges on both steps in a scenario of complicated data and lacking of sufficient domain knowledge. The recent developments in the deep learning (DL) approaches offer effective way of obtaining end-to-end learning models from complex healthcare data. Several aspects of DL find useful in healthcare domain namely improved performance, end to end learning scheme with integrated feature learning, ability of managing complex and multi-modality data, etc. Deep models permit the discovery of high-level features, improving performances over traditional models, increasing interpretability and providing additional understanding about the structure of the biological data.

The most important intention of bio-medicinal researches is to attain a medical expert for phenotypes requirements which is used in an ad hoc manner. However, supervised depiction of a feature space scale in a very bad fashion and quits the possibility to locate novel pattern. Then, learning techniques can able to explore illustrations which are mandatory for real data prediction (Domhan et al., 2015). DL methods are defined as representation-learning models with various stages, but nonlinear technique that change the representations at every stage into a representation at high abstraction stage (Schmidhuber, 2015; Leung and Haykin, 1991). DL techniques have the possibility in processing better efficiency particularly in the application of natural language processing operation, audio prediction and also computer vision.

According to various applications and continuous development in technical enrichment, DL model launches new prospects in the bio-medicinal data. The laborious work is utilised in DL parts for healthcare was defined already. For example, Google DeepMind has developed many methods to utilise the knowledge of the professionals in healthcare and has been utilised through DL intelligence to indicate issues on X-rays and computed tomography (CT) scanner (Enlitic Uses Deep Learning to Make Doctors Faster and More Accurate, 2016). Thus, DL technique was marginally applied to a wide series of medicinal issues that attains the advantage of these problems. There are numerous reasons allied to DL which might be relevant in healthcare such as, qualified performance, effective complexity management, end-to-end learning with integrated feature learning, multimodality data, etc. For the effective results, the DL research considered the difficulties concerned in healthcare data and that is essential for betterment of techniques and implementations. It stimulates DL as an interface with healthcare data flows and disease diagnosis models.

At the same time, smart medicinal informatics models, such as Philips' CareVue system, accumulate the details of patients in relational databases for data management purpose. Doctors often derive equivalent medicinal storages for an ICU patient which is intended for decision making processes. A famous and broadly used disease code model is referred as ICD and commonly planned by World Health Organisation (WHO). The extended technique is ICD-10 which is used with local medical adjustments in several fields, for example, ICD-10-AM for Australia. The major intention of ICD is to extend a limited hierarchical classifier technique that has been evolved to record health condition in different class labels. In the USA, the ninth version of ICD9 is utilised in diverse areas for classifier tasks. For example, an ICU patient will be merged with a file of ICD9 codes in medical record like disease observation, pathology, or clinical data management. From the observations of past data, caretaker is needed to give a better treatment for the patient. So, absolute and accurate disease prediction is an essential task.

During the training process of DL, it is mainly needed to minimise the learning rate as the training process gets continued. It can be carried out by the use of pre-defined learning rate schedules or adaptive learning rate models. This paper follows the learning rate schedules type, which modifies the learning rate at the time of training through the minimisation of learning rate based on a predefined schedule. To achieve this goal, this paper designs a novel stochastic gradient descent (SGD) with time with exponential decay (TED)-based learning rate scheduler called SGD-TED model for DL-based healthcare diagnosis. The presented SGD-TED model undergoes four processes namely pre-processing, classification [recurrent neural network (RNN), long short-term memory (LSTM) and gated recurrent unit (GRU)], SGD-based parameter tuning and TED-based learning rate scheduling. When the data pre-processing gets completed, the hyperparameter tuning of the DL models takes place by SGD and the TED is applied to schedule the learning rate proficiently. The application of SGD-TED approach in the DL models considerably helps to increase the classification performance. A brief set of experimental validation takes place on benchmark medical dataset and the results are discussed under different dimensions.

2 Related works

In recent times, DL method is mainly applied for computing the accumulated EHRs, like structured models such as disease analysis, treatments; lab tests whereas unstructured modules like medical notes. The majority works of EHRs in medical domain is operated using deep structure, supervised, and predictive tasks. Specifically, a typical approach depicts that DL reaches best results than traditional ML methods with respect to specific parameters, such as area under the receiver operating characteristic curve, accuracy and F-score (Manning et al., 2008). Here, massive studies define the end-to-end supervised system, and only some modules are unsupervised networks. It accelerates latent patient depictions, which are evaluated using shallow classifiers namely, random forests (RF), logistic regression (LR).

Many works have been applied with DL model for diagnosing the disease according to the health state of a patient (Cheng et al., 2016) which applied a four-layer convolutional neural network (CNN) to predict the congestive heart disease (CHD) and chronic obstructive pulmonary disease and implied significant benefits over the baselines. RNNs with LSTM were employed in DeepCare system that examines the future medical records. In addition, developers have established an LSTM unit followed by a degrading effect to manage irregular events. Moreover, it is incorporated with clinical inventions to form the predications dynamically. Deep care is estimated for disease progression labelling, invention and future predictive risks prediction on diabetes as well as physical health. RNNs and GRU were utilised by Choi et al. (2016) to design doctor AI, an end-to-end method which applies patient records for disease diagnosis and preferred treatment has to be consumed. Such calculations demonstrate a vital recall measure when compared with shallow baselines and optimal normalisation with the help of consequent approach without any loss of accuracy.

Miotto et al. (2016) deployed a model to learn deep patient implications from EHRs by applying three-layer stacked denoising autoencoder (SDA). Furthermore, it is applied with a novel depiction on detecting the risk aspects with the help of RF classifier. The simulation outcome shows that, deep representation leads to generate a significant prediction than using actual EHRs such as principal component analysis (PCA), and k-means. Additionally, it demonstrates that, the achieved outcomes improve the addition of LR layer on top of AE to fine-tune the entire supervised system. Similarly, Liang et al. (2014) used RBMs to learn representations from EHRs that tends to develop novel methods and illustrated best prediction accuracy even under massive disease classes.

Moreover, DL was applied in frequent time signals, such as, lab results, which has to identify specific phenotypes automatically. Lipton et al. (2015) employed NNs with LSTM for pattern analysis from multivariate time sequence of clinical values. Specifically, the training has been offered for classifying massive cases; however irregular samples of medical values of patients in ICU. Thus, the obtained results states that improvement with respect to diverse robust baselines, like multilayer perceptron (MLP) is trained on hand-based features. Che et al. (2015) deployed SDAs normalised with a prior knowledge on the basis of ICD-9s that is widely applied in predicting medical patterns. Lasko et al. (2013) developed a two-layer SAE for the establishment of longitudinal series of serum uric acid values to classify the uric-acid signatures of gout and acute leukemia.

Razavian et al. (2016) evaluated CNNs as well as RNNs with LSTM units to detect the disease onset from lab test values, and depicted best working functions than LR and hand-crafted features, medical relied features. Neural language deep models were employed in EHRs, to learn incorporated presentations of medical objectives such as diseases, remedies, lab tests, and so on. It is utilised in predicting the existence of a disease. Tran et al. (2015) presented RBMs for learning abstractions of ICD-10 codes on physical health patients for detecting the risk of a suicide attempt. A deep structure depends upon RNNs that accomplishes best outcome for eliminating secured health data from medical notes to limit the automatic re-identification of free-text patient values. The examination of unknown patient re-admissions. Additionally, Wickramasinghe (2015) projected Deepr, dedicated structure is based on CNNs which detects and combines clinical motifs in longitudinal patient EHRs to solve the medical risks. Deepr functions perform better in predicting readmission within a specific time interval and able to predict applicable and interpretable clinical patterns.

3 The proposed SGD-TED model

The working process involved in the presented SGD-TED model is shown in Figure 1. As depicted, the medical data is initially collected and pre-processed into a proper format. Then, the SGD-TED model is applied to the DL models such as RNN, GRU and LSTM to find the appropriate hyper parameters and learning rate. Finally, a set of performance measures are used to investigate the classifier results of the SGD-TED model in terms of precision, recall, accuracy, F-score and kappa.

3.1 Pre-processing

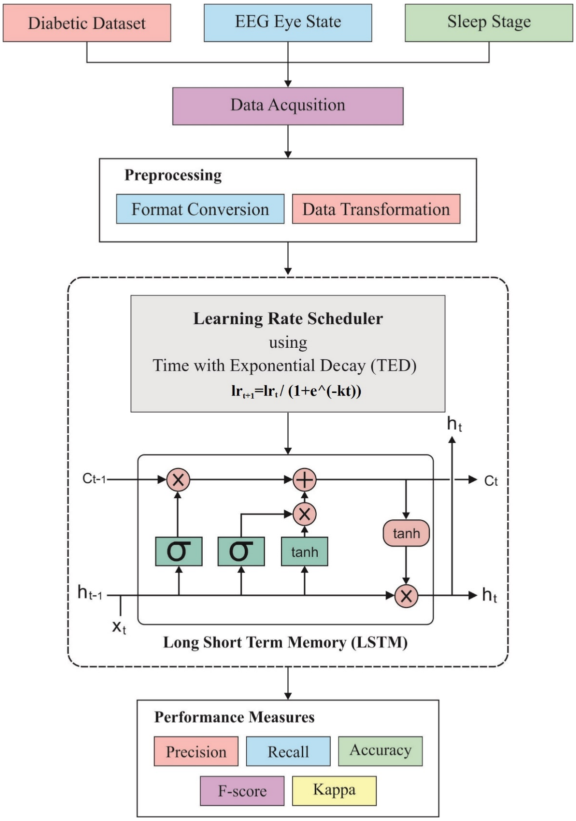
Data pre-processing includes the process of transforming the original input data into a comprehensible format. As the real-time data is incomplete and consist of missing values, there is a maximum possibility of having errors. Thus, data pre-processing is used for data conversion that is from actual point to proper format which has to be applicable for next processing. In this approach, pre-processing is carried out in two phases such as format conversion as well as data transformation. Initially, format conversion process takes place where the data is in any kind of format is changed into .csv format. Secondly, data transformation is computed and it has diverse sub-processes as shown in the following.

- *Normalisation:* it processed the data scaling from the range of (–1.0 to 1.0 or 0.0 to 1.0).
- *Attribute selection:* a subset of parameters was chosen from the given collection of attributes for mining the task.
- *Discretisation:* actual values of mathematical attributes would be replaced by interval or conceptual levels.

3.2 Classification process

RNN, LSTM, and GRU are the popular DL models widely employed for classification process. The RNN is a layered NN where the output activation from one or multiple layers of the network is used. It considers that the inputs are not dependent on one another and find useful if the inputs are sequential. Though RNN is effective, it suffers from vanishing gradient problem that prevents them from the use of long term data. To resolve this problem, improved versions of RNN called LSTM and GRU are designed. They have the ability to remember long term dependencies. The architecture of both LSTM and GRU cells are mainly based on the fundamental RNN cell. The LSTM and GRU models are designed to retain data for longer duration with no need of dealing with the vanishing gradient problem. They have internal mechanism termed as gates which controls the data flows. These gates can learn which data in a sequence is important to keep or throw away. Thus, it can pass relevant information down the long chain of sequences to make predictions. Besides, the major difference between GRU and LSTM is that GRU's includes two gates namely reset and update whereas LSTM has three gates such as input, output, forget. Therefore, GRU is less complex than LSTM because it has few gates. If the dataset is small then GRU is preferred otherwise LSTM for the larger dataset. The detailed working of these three DL models is given in the following subsections.

Figure 1 Work flow of SGD-TED model (see online version for colours)



3.2.1 *RNN-based classification model*

RNN comes under the NN in which results from previous step is provided as input to the further step. For traditional NN, each input and output is independent; but, the data analysis is carried out with the help of existing data and without storing the previous information. Thus, RNN is applied for solving the issues using hidden layer. The standard feature of RNN is a hidden state which records data about the sequence. RNN contains a 'memory' which saves all data which has to be estimated. It uses same parameters for each input as it computes the same operation on hidden layers and generates best outcome. At last, the complexity of parameters was limited. Hence, NN process the given operation:

- RNN converts the independent activations for dependent actions by providing same weights and biases for every layers, and reduce the parameter complexity and memorise all existing results by giving the result as input to upcoming hidden layer.
- Therefore, three layers were integrated where weights and bias of all hidden layers are similar, which develops individual recurrent layer.

3.2.2 LSTM-based classification process

LSTM has been developed for handling the prolonged term dependency which does not develop vanishing gradient problem as the fact that the LSTM exploits memory cell state to process the data for longer duration. The cell state is highly applicable to process data even the information is not applied for prolonged time. The LSTM is composed of three gates namely, forget, update and reset gates. Both input and output gate manages the access CEC control. During the training process, the input gate is learned and allows novel data within the CEC. While the input gate is 0, no data is linked. Similarly, the output gate is learned and enables the data from CEC. When the gates are closed, data gets terminated inside a memory cell. It activates the error signals to flow over several time steps with no assumption of vanishing gradients.

The LSTM outperforms than RNN at the time of learning long-range dependency. This method is ineffective in data sequence. The LSTM condition is not arranged when the input stream is detached externally into sized sequences. In particular, the LSTM knows to reset the memory cell as it completes the sequence and enters into novel sequence. In order to resolve the issue, LSTM structure with forget gates has been deployed. The architectural diagrams of LSTM unit with forget gates are defined in the following:

- *Input:* the LSTM unit uses recent input vector denoted by x_t and shows the time step as h_{t-1}. The weighted inputs are summarised and passed by tanh activation which provides in z_t.
- *Input gate:* it reads x_t and h_{t-1}, determines the weighted sum, and applies sigmoid activation. The final outcome is enhanced with z_t, for providing input flow of a memory cell.
- *Forget gate:* it is worked by an LSTM unit that resets memory data when they are expired and irregular. It exists when the system begins to process a new series. The forget gate reads x_t and h_{t-1} and applies a sigmoid activation for weighted inputs. The results, f_t are enhanced with the help of a cell state at existing time step s_{t-1} that enables to forget the memory data which is unwanted.

- *Memory cell:* it is constrained with CEC, and a recurrent edge as well as unit weight. The current cell state s_t is estimated to forget irregular data from earlier time step and ensures relevant data from current input.
- *Output gate:* it applies a weighted sum of x_t and h_{t-1} and uses sigmoid activation to manage the data flow from LSTM unit.
- *Output:* the result of LSTM unit h_t, is determined by converting a cell state s_t by a tanh and enhance with output gate, o_t. The function of LSTM unit is represented as given in the following:

$$z_t = \tanh\left(W^z x_t + R^z h_{t-1} + b^z\right)(input) \tag{1}$$

$$i_t = \sigma\left(W^i x_t + R^i h_{t-1} + b^i\right)(inputgate) \tag{2}$$

$$f_t = \sigma\left(W^f x_t + R^f h_{t-1} + b^f\right)(forgetgate) \tag{3}$$

$$o_t = \sigma\left(W^o x_t + R^o h_{t-1} + b^o\right)(outputgate) \tag{4}$$

$$s_t = z_t \odot i_t + s_{t-1} \odot f_t(cellstate) \tag{5}$$

$$h_t = \tanh\left(s_t\right) \odot o_t(output) \tag{6}$$

3.2.3 GRU-based classification model

The vanishing-exploding gradients problems can be resolved under the application of RNN. The main approach is LSTM. A method with lower popularity but higher productive difference is termed as GRU. Dissimilar to LSTM, it contains three gates that does not sustain the internal cell state. The data stored in internal cell state in an LSTM recurrent unit is incorporated in hidden state of GRU. The gathered data is offered to the upcoming GRU. The gates of a GRU are defined as follows:

- *Update gate (z):* it process the existing knowledge which has to be induced to next processing. It is analogous to output gate in LSTM recurrent unit.
- *Reset gate (r):* it calculated the previous knowledge that has to be removed. It is analogous to the combination of input gate and forget gate from LSTM recurrent unit.
- *Current memory gate* $(\overline{h_t})$: it is highly used for GRU process. It is integrated into reset gate with input modulation gate is a sub part of input gate and used for developing nonlinearity into input and make input zero-mean. An alternative for creating a sub-part of reset gate is to restrict the impact of previous data on current data that is used for upcoming process.

The basic operation of GRU is combined with RNN with few differences among two methods. The interior computation of GRU contains gates which change the present input and previous hidden state. The performance of GRU is defined in the following. At the initial phase, the input as recent input and previous hidden state are named as vectors. Followed by, the vales of three diverse gates are determined in the following:

- For every gate, compute the parameterised input and existing hidden state vectors by the computation of element-wise multiplication over the considered vector and corresponding weights.
- Apply the concerned activation function for gate element-wise on parameterised vectors. The list of gates is offered using the activation function.

3.3 Stochastic gradient descent

DL model has been trained with the application of SGD method. It is defined as an optimisation technique which evaluates the error gradient for recent state of an approach under the employment of training dataset, and upgrades the weights of the model by applying back-propagation (BP) of errors algorithm which is named as BP model. Consider a simple supervised learning procedure. Every instance z is a pair (x, y) which is enclosed with random input x as well as a scalar y. Suppose a loss function $l(\hat{y}, y)$ calculates the prediction cost $\hat{y}$ when the original answer is y and select a family F of functions $f_w(x)$ is parameterised using a weight vector w. Here, the function $f \in F$ which reduces the loss $Q(z, w) = l(f_w(x), y)$ averaged on the samples. Though the unknown distribution $dP(z)$ is maximised which embeds the Laws of Nature, then it settles for processing the maximum on a sample $z_1, \ldots, z_n$.

$$E(f) = \int l(f(x), y)dP(z)E_n(f) = \frac{1}{2}\sum_{i=1}^{n} l\left(f(x_i), y_i\right) \tag{7}$$

The empirical risk $E_n(f)$ determines the function of training set. The desired risk $E(f)$ calculates the generalisation task which is highly required on upcoming samples. The statistical learning principle involves in reducing the empirical risk than using expected risk while the selected family F is highly limited.

3.3.1 Gradient descent (GD)

It is projected to reduce the empirical risk $E_n(f_w)$ with the help of GD. Every iteration updates the weights w according to the gradient of $E_n(f_w)$.

$$w_{t+1} = w_t - \gamma\frac{1}{2}\sum_{i=1}^{n} \nabla wQ\left(z_i, w_t\right), \tag{8}$$

where γ defines the sufficiently selected learning rate. Using adequate regularity considerations, while initial estimate w_0 is nearby the optimum, and if the learning rate γ is sufficiently minimum, this method accomplishes linear convergence (Leung and Haykin, 1991), such as $-log\, \rho \sim t$, where ρ is the residual error. The massive solutions for optimisation models are developed by replacing scalar learning rate γ by a positive definite matrix Γ_t seeks inverse of Hessian of cost at optimal range:

$$w_{t+1} = w_t - \Gamma_t \frac{1}{2}\sum_{i=1}^{n} \nabla wQ\left(z_i, w_t\right), \tag{9}$$

The second order GD (2GD) is defined as a variant for popular Newton model. With the help of optimistic regularity considerations, have offered w_0 which is closer to the optimum, and 2GD attains quadratic convergence. While the cost is quadratic and scaling matrix Γ is accurate, the technique accomplishes the optimal range after each iteration. Else, the enough smoothness would be $-log\ log\ \rho \sim t$.

3.3.2 Stochastic gradient descent

The SGD method is assumed to be a drastic simplification model. Rather applying the gradient of $E_n(f_w)$ accurately, all iterations determine the gradient according to randomly selected instance z_t:

$$w_{t+1} = w_t - \gamma_t \nabla wQ(z_i, w_t) \tag{10}$$

The stochastic task $\{w_t, t = 1, \ldots\}$ is based on the samples that has been selected in random manner. At this point, the SGD optimised the desired risk, where the instances are arbitrarily obtained from the ground truth distribution.

3.4 Learning rate scheduler

While training the DL model, it is highly applicable in reducing learning rate (γ_t) when there is progress in training phase. The number of weights upgraded at the time of training is said to be a step size or 'learning rate'. In particular, learning rate is a configurable hyperparameter applied in training NN with minimum positive value, from 0.0 and 1.0. The learning rate should be fixed in an appropriate way to achieve better results. Smaller learning rates needs maximum training epochs that provides tiny modifications for the weights while learning rates tends to provide drastic changes and needs minimum training epochs. The process of tuning the learning rate is difficult. A high learning rate leads to the divergent training process whereas a low learning rate results in slow convergence. To achieve effective outcome, it is needed to simulate with different learning rate at the time of training. A technique used to schedule leaning rate is called as learning rate scheduler. Typical learning rate schedulers are time-based decay, step decay and exponential decay.

The SGD optimisation algorithm in the SGD class includes an argument known as decay. It is employed in the time-based learning rate decay schedule, as given below.

$$Learning\ Rate = Learning\ Rate * \frac{1}{1 + decay * iteration} \tag{11}$$

If the decay argument is 0 (the default), learning rate remains same. Once the decay value is defined, it would reduce the learning rate from the previous epoch by the given fixed amount. In this view, we have developed a new learning rate scheduler called TED by the integration of time decay and exponential decay to achieve maximum classification performance.

The time-based learning schedule can be mathematically represented as follows:

$$lr_{t+1} = \frac{lr_t}{1 + kt} \tag{12}$$

where k is a decay parameter, t is the iteration step and lr is the learning rate. The equation states that the SGD model takes decay and lr arguments and update the new learning rate by a decreasing factor in every individual epoch.

The exponential learning schedule can be mathematically defined by:

$$lr_t = lr_0 * e^{-kt} \tag{13}$$

where lr_0 is the initial learning rate, k is decay parameter and t is the iteration step. This function exponentially decaying the learning rate and fed into the learning rate scheduler. The proposed TED learning rate scheduler is the integration of time and exponential learning schedule functions, as defined below.

$$lr_{t+1} = \frac{lr_t}{1 + e^{-kt}} \tag{14}$$

The proposed SGD-TED learning rate scheduler exponentially decays the learning rate based on learning rate of previous time iteration. This leads to achieve minimum and maximum decay values. Therefore, the application of TED helps to achieve optimal learning rate and thereby classifier outcome gets increased.

4 Performance validation

The presented model is simulated using Python 3.6.5 tool. In addition, three different datasets namely diabetes, EEG Eye State and Framingham dataset are used. A set of measures used to examine the results are precision, recall, F-measure, accuracy, and kappa. These measures are defined in equations (15)–(19).

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

(18)

$$F\text{-}measure = \frac{2TP}{2TP + FP + FN} \tag{19}$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives respectively.

4.1 Dataset description

This segment reviewed the efficiency obtained through SGD-TED on varied datasets. A group of three datasets by diagnoses code like diabetes, EEG Eye State and Framingham has been used for result analysis of the SGD-TED (Strack et al., 2014; EEG Eye State Data Set, 2013; Ajmera, 2018). The earlier diabetes dataset has the whole 101,766 samples with a group of 49 features. Moreover, two class labels are available where a group of 78,363 samples appears in positive class type and the remaining 23,403 samples come in negative class type. The later EEG Eye State datasets consist of a sum of 14,980 samples along with a collection of 15 features. Further, a two class labels come under this dataset where a group of 82,527 samples falls in class one label and the remaining 6,723 samples comes in class two labels. Besides, the Framingham dataset comprises a set of 4,240 samples with the collections of 16 features. In addition, two class labels present where a collection of 3,596 samples fall beneath zero class and remaining 644 samples comes in class one.

4.2 Results analysis

In order to validate the effective performance of the presented model, a series of existing techniques such as SGD with exponential learning scheduler (SGDE) (Li and Arora, 2019), SGD with constant (SGDC) (Chee and Toulis, 2018), SGD with time-based learning scheduler (SGDT) (Lau, 2017), and SGD with step decay (SGDS) (Ge, 2019) models are used. The SGDE involves an exponential increase in learning rate schedule, i.e., learning rate increases by some $(1 + \alpha)$ factor in every epoch for some $\alpha > 0$. Next, the SGDC includes a default learning rate schedule in SGD optimiser. The SGDT has an argument called decay. This argument is used in the time-based learning rate decay schedule. Also, in SGDS, the learning rate gets dropped by a factor every few epochs.

A detailed comparative accuracy analysis of the SGD-TED with other models takes place and the results obtained on the three dataset are given in Figures 2–4.

Figure 2 Analysis of accuracy on proposed vs. existing learning rate scheduler on diabetes dataset (see online version for colours)

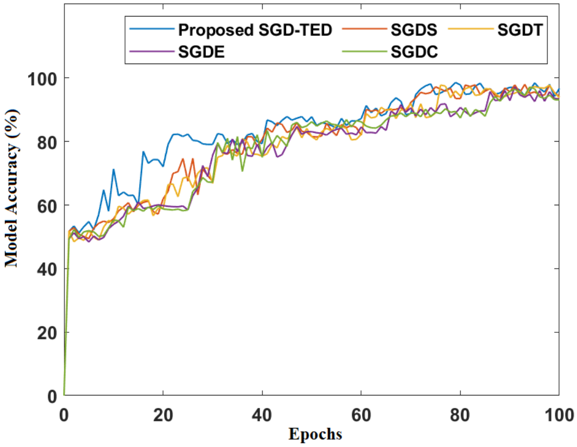


Figure 2 shows the accuracy analysis of the presented SGD-TED model compared to other learning rate scheduler models for SGD on the applied diabetes dataset. Under the varying number of epochs, the figure portrayed that the SGDE and SGDC models have exhibited reduced classifier outcome with the accuracy of 97.85% and 97.64%. At the same time, the SGDT and SGDS models have tried to show better accuracy over the earlier models with the accuracy of 97.91% and 98.10%, but not higher than the presented SGD-TED model. The presented model has exhibited higher accuracy of 98.59% and it gets increased with an increase in epoch count.

The accuracy analysis of the presented SGD-TED technique is represented in Figure 3 over previous learning rate schedule techniques for SGD on the EEG Eye State dataset. With a variation in epoch count, the figure revealed that the SGDE and SGDC techniques have shown comparatively lower classification results with the accuracy of 90.41% and 89.23% respectively. Simultaneously, the SGDT and SGDS techniques have offered betterment in accuracy values of 91.76% and 92.45%, except the projected SGD-TED technique. The proposed technique has demonstrated high accuracy rate of 93.68% and it continues to increase with a rise in number of epochs.

Figure 3 Analysis of accuracy on proposed vs. existing learning rate scheduler on EEG Eye State dataset (see online version for colours)

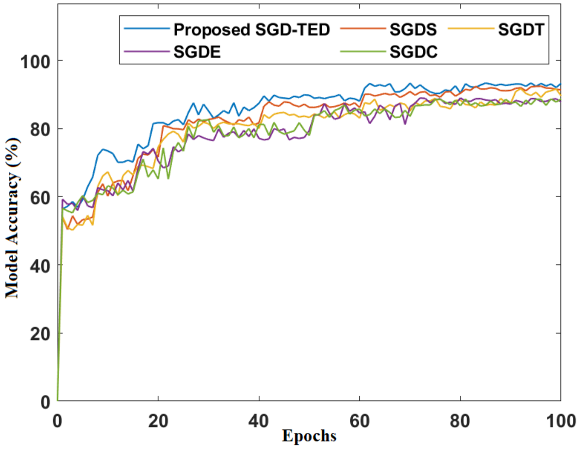


Figure 4 Analysis of accuracy on proposed vs. existing learning rate scheduler on sleep stage dataset (see online version for colours)

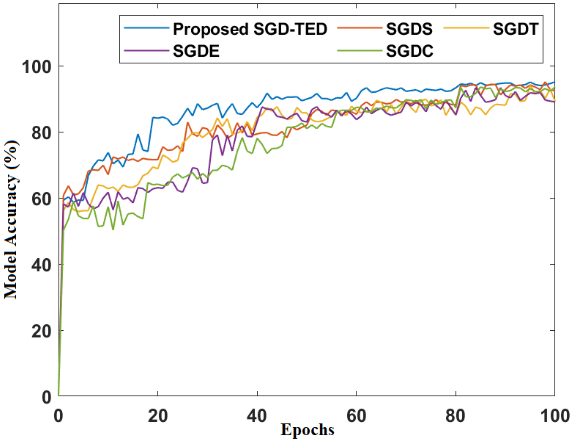


Figure 4 investigates the comparative accuracy analysis of the presented SGD-TED and existing learning rate schedule techniques for SGD on the sleep stage dataset. On analysing the results with various numbers of epochs, the figure pointed out that the SGDE and SGDC techniques have demonstrated ineffective performance with the accuracy of 93.90% and 93.68%. Concurrently, the SGDT and SGDS techniques have exhibited better performance to a certain extent over SGDE and SGDC models with the accuracy of 94.27% and 94.85%. However, a maximum accuracy rate of 95.20% has been attained by the presented model under the variation in number of epochs.

Table 1 Performance analysis of proposed method with existing methods (Li and Arora, 2019; Chee and Toulis, 2018; Lau, 2017; Ge et al., 2019) for applied datasets

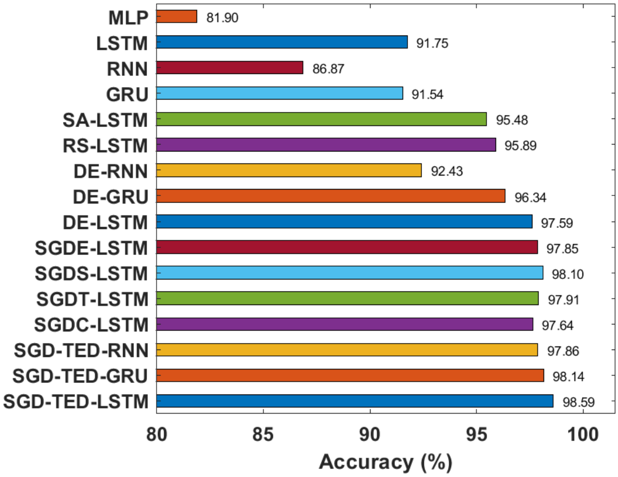
Methods	*Measures*	*Precision*	*Recall*	*F-measure*	*Accuracy*	*Kappa*
Diabetes	SGD-TED-LSTM	97.65	98.10	98.22	98.59	97.63
	SGDC-LSTM	96.43	96.10	97.12	97.64	95.81
	SGDT-LSTM	97.10	97.23	97.45	97.91	95.97
	SGDS-LSTM	97.34	97.83	98.02	98.10	96.87
	SGDE-LSTM	96.54	96.38	97.46	97.85	95.92
EEG Eye State	SGD-TED-LSTM	95.42	94.61	95.33	93.68	92.90
	SGDC-LSTM	86.55	86.80	88.31	89.23	86.05
	SGDT-LSTM	89.62	90.29	91.40	91.76	89.60
	SGDS-LSTM	90.43	91.30	91.65	92.45	90.84
	SGDE-LSTM	86.86	87.12	88.43	90.41	88.92
Sleep stage	SGD-TED-LSTM	96.31	95.83	95.87	95.20	94.09
	SGDC-LSTM	91.65	92.86	92.90	93.68	92.82
	SGDT-LSTM	92.44	91.40	92.47	94.27	93.21
	SGDS-LSTM	93.60	93.69	93.85	94.85	93.76
	SGDE-LSTM	91.86	92.91	92.97	93.90	93.10

Table 1 provides a comparative analysis of the results offered by distinct models on the applied three datasets. On the applied diabetes dataset, the SGDC-LSTM model has showed its inefficient classifier outcome by providing minimal precision of 96.43%, recall of 96.10%, F-measure of 97.12%, accuracy of 97.64% and kappa value of 95.81%. On the other hand, a slightly higher precision of 96.54%, recall of 96.38%, F-measure of 97.46%, accuracy of 97.85% and kappa value of 95.92% is offered by the SGDE-LSTM model. At the same time, the SGDT-LSTM model has showed better results to a certain extent with the precision of 97.10%, recall of 97.23%, F-measure of 97.45%, accuracy of 97.91% and kappa value of 95.97%. Also, the SGDS-LSTM model has demonstrated better results over the earlier methods with the high precision of 97.34%, recall of 97.83%, F-measure of 98.02%, accuracy of 98.10% and kappa value of 96.87%. At last, the SGD-TED-LSTM model has exhibited maximum classification performance with the higher precision of 97.65%, recall of 98.10%, F-measure of 98.22%, accuracy of 98.59% and kappa value of 97.63%.

The classification results analysis of the projected SGD-TED method on the given EEG Eye State dataset stated that the SGDC-LSTM approach has displayed its worst classifier result by providing lower precision of 86.55%, recall of 86.80%, F-measure of 88.31%, accuracy of 89.23% and kappa value of 86.05%. Besides, a better precision of 86.86%, recall of 87.12%, F-measure of 88.43%, accuracy of 90.41% and kappa value of 88.92% is provided by the SGDE-LSTM model. Simultaneously, the SGDT-LSTM model has depicted better results to a greater extent with the precision of 89.62%, recall of 90.29%, F-measure of 91.40%, accuracy of 91.76% and kappa value of 89.60%. Additionally, the SGDS-LSTM model has shown better results than previous models with the maximum precision of 90.43%, recall of 91.30%, F-measure of 91.65%, accuracy of 92.45% and kappa value of 90.84%. Finally, the SGD-TED-LSTM model has showcased higher classification performance with the best precision of 95.42%, recall of 94.61%, F-measure of 95.33%, accuracy of 93.68% and kappa value of 92.90%.

The classification results analysis of the projected SGD-TED model on the given Sleep State dataset implied that the SGDC-LSTM model has depicted its poor classifier outcome by providing least precision of 91.65%, recall of 92.86%, F-measure of 92.90%, accuracy of 93.68% and kappa value of 92.82%. Then, a moderate precision of 91.86%, recall of 92.91%, F-measure of 92.97%, accuracy of 93.90% and kappa value of 93.10% is offered by the SGDE-LSTM model. Meanwhile, the SGDT-LSTM technologies has shown measured results to a certain limit with the precision of 92.44%, recall of 91.40%, F-measure of 92.47%, accuracy of 94.27% and kappa value of 93.21%. Moreover, the SGDS-LSTM model has depicted manageable results over the traditional methods with the maximum precision of 93.60%, recall of 93.69%, F-measure of 93.85%, accuracy of 94.85% and kappa value of 93.76%. Consequently, the SGD-TED-LSTM model has shown best classification performance with the optimal precision of 96.31%, recall of 95.83%, F-measure of 95.87%, accuracy of 95.20% and kappa value of 94.09%.

Figure 5 Accuracy analysis on diabetes dataset (see online version for colours)



A comparative results analysis of the SGD-TED model with existing methods such as differential evolution (DE)-based LSTM (DE-LSTM), DE-GRU, DE-RNN, rough set-based LSTM (RS-LSTM), simulated annealing-based LSTM (SA-LSTM), GRU, RNN, LSTM, and MLP model is made on the diabetes dataset in Table 2 and Figure 5. The experimental indicated that the MLP and RNN models have failed to show effective results by achieving lower accuracy of 81.90% and 86.87% respectively. It is also noted that the GRU and LSTM models have reached to slightly higher and closer accuracy of 91.54% and 91.75% respectively. Along with that, it is noticed that the DE-RNN model has shown somewhat higher accuracy of 92.43%. Besides, the RS-LSTM and SA-LSTM models have portrayed moderate accuracy of 95.89% and 95.48% respectively. On continuing with, the SGD-TED-RNN, SGD-TED-RNN, SGDT-LSTM, SGDE-LSTM, DE-LSTM and DE-GRU models have attained acceptable performance with the accuracy of 97.86%, 97.64%, 97.91%, 97.85%, 97.59% and 96.34% respectively. Along with that, the SGD-TED-GRU and SGDS-LSTM models have exhibited competitive accuracy values of 98.14% and 98.10% respectively. However, the presented SGD-TED-LSTM model has shown superior performance with the higher accuracy of 98.59%.

Table 2 Comparative accuracy analysis of SGD-TED-LSTM with existing methods (Kaliyapillai and Krishnamurthy, 2020) on diabetes dataset

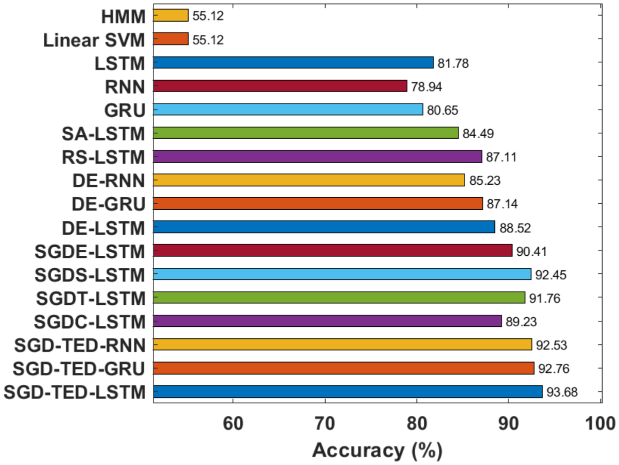
Classifiers	*Accuracy (%)*
SGD-TED-LSTM	*98.59*
SGD-TED-GRU	*98.14*
SGD-TED-RNN	*97.86*
SGD-TED-RNN	*97.64*
SGDT-LSTM	*97.91*
SGDS-LSTM	*98.10*
SGDE-LSTM	*97.85*
DE-LSTM	97.59
DE-GRU	96.34
DE-RNN	92.43
RS-LSTM	95.89
SA-LSTM	95.48
GRU	91.54
RNN	86.87
LSTM	91.75
MLP	81.90

The result analysis provided by the SGD-TED model with existing methods such as DE-LSTM, DE-GRU, DE-RNN, RS-LSTM, SA-LSTM, GRU, RNN, LSTM, linear support vector machine (SVM), and hidden Markov model (HMM) models take place with respect to accuracy on the EEG Eye State dataset is shown in Table 3 and Figure 6. The results implied that the HMM and Linear SVM methodologies have failed to showcase effective results by accomplishing minimum and closer accuracy of 55.12%. Also, it is pointed that the RNN, GRU and LSTM models have attained better and nearby accuracy of 78.94%, 80.65% and 81.78% correspondingly. In line with this, it is pointed that the SA-LSTM and DE-RNN approaches have shown reasonable accuracy of 84.49% and 85.23% respectively. On the other hand, the RS-LSTM, DE-GRU and DE-LSTM frameworks have implied gradual accuracy of 87.11%, 87.14% and 88.52% respectively. Along with that, the SGDC-LSTM, SGDE-LSTM, SGDT-LSTM and SGDS-LSTM schemes have reached considerable function with accuracy of 89.23%, 90.41%, 91.76% and 92.45% correspondingly. Similarly, the SGD-TED-GRU and SGD-TED-RNN models have showcased competing accuracy values of 92.76% and 92.53% correspondingly. Hence, the projected SGD-TED-LSTM model has depicted qualified performance with the maximum accuracy of 93.68%.

Table 3 Comparative accuracy analysis of SGD-TED-LSTM with existing methods (Kaliyapillai and Krishnamurthy, 2020) on EEG Eye State dataset

Classifiers	*Accuracy (%)*
SGD-TED-LSTM	*93.68*
SGD-TED-GRU	*92.76*
SGD-TED-RNN	*92.53*
SGDC-LSTM	*89.23*
SGDT-LSTM	*91.76*
SGDS-LSTM	*92.45*
SGDE-LSTM	*90.41*
DE-LSTM	88.52
DE-GRU	87.14
DE-RNN	85.23
RS-LSTM	87.11
SA-LSTM	84.49
GRU	80.65
RNN	78.94
LSTM	81.78
Linear SVM	55.12
HMM	55.12

Figure 6 Accuracy analysis on EEG Eye State dataset (see online version for colours)



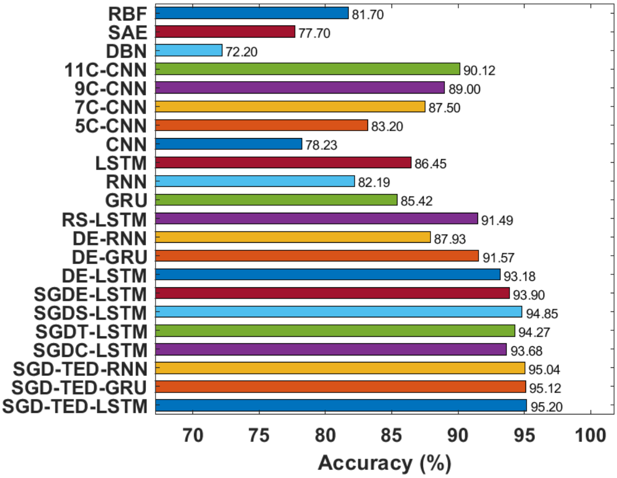
An accuracy analysis of the SGD-TED method is compared with the DE-LSTM, DE-GRU, DE-RNN, RS-LSTM, GRU, RNN, LSTM, CNN, 5 channel CNN (5C-CNN), 7 channel CNN (7C-CNN), 9 channel CNN (9C-CNN), 11 channel CNN (11C-CNN), deep belief networks (DBN), stacked autoencoders (SAE), and radial basis function (RBF) on the Sleep stage dataset is depicted in Table 4 and Figure 7. The figure portrayed that the DBN and SAE methodologies have failed to imply productive results by accomplishing minimal accuracy of 72.20% and 77.70% correspondingly. It is pointed that, the CNN, RBF and RNN approaches have attained a better and identical accuracy of 78.23%, 81.70% and 82.19% correspondingly. In line with this, it is notified that the 5C-CNN, GRU and LSTM models have reasonable accuracy of 83.20%, 85.42% and 86.45% respectively.

Table 4 Comparative accuracy analysis of SGD-TED-LSTM with existing methods (Kaliyapillai and Krishnamurthy, 2020) on sleep stage dataset

Classifiers	*Accuracy (%)*
SGD-TED-LSTM	*95.20*
SGD-TED-GRU	*95.12*
SGD-TED-RNN	*95.04*
SGDC-LSTM	*93.68*
SGDT-LSTM	*94.27*
SGDS-LSTM	*94.85*
SGDE-LSTM	*93.90*
DE-LSTM	93.18
DE-GRU	91.57
DE-RNN	87.93
RS-LSTM	91.49
GRU	85.42
RNN	82.19
LSTM	86.45
CNN	78.23
5C-CNN	83.20
7C-CNN	87.50
9C-CNN	89.00
11C-CNN	90.12
DBN	72.20
SAE	77.70
RBF	81.70

Next, the 7C-CNN, DE-RNN and 9C-CNN frameworks have implied gradual accuracy of 87.50%, 87.93% and 89%. Additionally, the 11C-CNN, RS-LSTM approaches have performed better result with considerable accuracy of 90.12% and 91.49% correspondingly. Similarly, the DE-GRU and DE-LSTM methodologies have productive performance with maximum accuracy of 91.57% and 93.18% respectively. On the same way, the SGDC-LSTM, SGDE-LSTM, SGDT-LSTM and SGDS-LSTM models have reached manageable function with the accuracy of 93.68%, 93.90%, 94.27% and 94.85% correspondingly. Similarly, the SGD-TED-GRU and SGD-TED-RNN methods have shown competing accuracy values of 95.12% and 95.04% respectively. Therefore, the proposed SGD-TED-LSTM model has displayed qualified performance with the higher accuracy of 95.20%. The above mentioned tables and figures indicated the effective learning rate determination and maximum classification outcome of the presented model on the applied medical dataset.

Figure 7 Accuracy analysis on sleep stage dataset (see online version for colours)



5 Conclusions

This paper has developed an effective SGD with learning rate scheduling model for DL-based healthcare diagnosis model. The integration of time and exponential decay approaches helps to achieve proper learning rate at minimal computation time. The presented SGD-TED model initially undergoes data pre-processing and then DL-based data classification process takes place. The SGD-TED model is applied to tune the parameters of the three DL models namely LSTM, RNN and GRU. The experimental results of the SGD-TED model are validated using three benchmark dataset. The simulation outcome indicated that the SGD-TED-LSTM model has resulted to a higher accuracy of 98.59%, 93.68% and 95.20% on the applied diabetes, EEG Eye State and sleep stage dataset. As a part of future scope, the presented model can be deployed in an internet of things (IoT) and cloud-based platform to assist physicians and doctors from remote areas.

Acknowledgements

This work was supported by AICTE Research funding under the Research Promotion scheme-2019-VIDE NO:F.No.8-35/RIFD/RPS. The authors acknowledge the support of

Dr. V. Akila of Department of Computer Science and Engineering, Pondicherry Engineering College in her assistance in improving the quality of the manuscript.

References

Ajmera, A. (2018) *Framingham Heart Study Dataset* [online] https://www.kaggle.com/amanajmera1/framingham-heart-study-dataset (accessed 14 June 2020).

Bellazzi, R. and Zupan, B. (2008) 'Predictive data mining in clinical medicine: current issues and guidelines', *International Journal of Medical Informatics*, Vol. 77, No. 2, pp.81–97.

Che, Z., Kale, D., Li, W., Bahadori, M.T. and Liu, Y. (2015) 'Deep computational phenotyping', in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August, pp.507–516.

Chee, J. and Toulis, P. (2018) 'Convergence diagnostics for stochastic gradient descent with constant learning rate', in *International Conference on Artificial Intelligence and Statistics*, PMLR, March, pp.1476–1485.

Cheng, Y., Wang, F., Zhang, P. and Hu, J. (2016) 'Risk prediction with electronic health records: a deep learning approach', in *Proceedings of the 2016 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, June, pp.432–440.

Choi, E., Bahadori, M.T., Schuetz, A., Stewart, W.F. and Sun, J. (2016) 'Doctor AI: predicting clinical events via recurrent neural networks', in *Machine Learning for Healthcare Conference*, PMLR, December, pp.301–318.

Domhan, T., Springenberg, J.T. and Hutter, F. (2015) 'Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves', in *Twenty-fourth International Joint Conference on Artificial Intelligence*, June.

EEG Eye State Data Set (2013) [online] https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State (accessed 14 June 2020).

Enlitic Uses Deep Learning to Make Doctors Faster and More Accurate (2016) [online] https://www.enlitic.com/index.html (accessed 29 November 2016).

Ge, R., Kakade, S.M., Kidambi, R. and Netrapalli, P. (2019) *The Step Decay Schedule: A Near Optimal, Geometrically Decaying Learning Rate Procedure for Least Squares*, arXiv preprint arXiv: 1904.12838.

Hripcsak, G. and Albers, D.J. (2013) 'Next-generation phenotyping of electronic health records', *Journal of the American Medical Informatics Association*, Vol. 20, No. 1, pp.117–121.

Kaliyapillai, S. and Krishnamurthy, S. (2020) 'Differential evolution based hyperparameters tuned deep learning models for disease diagnosis and classification', *Adv. Sci. Technol. Eng. Syst. J.*, Vol. 5, No. 5, pp.253–261.

Lasko, T.A., Denny, J.C. and Levy, M.A. (2013) 'Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data', *PloS One*, Vol. 8, No. 6, p.e66341.

Lau, S. (2017) *Learning Rate Schedules and Adaptive Learning Rate Methods for Deep Learning, towards Data Science* [online] https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1 (accessed June 14 2020).

Leung, H. and Haykin, S. (1991) 'The complex backpropagation algorithm', *IEEE Transactions on Signal Processing*, Vol. 39, No. 9, pp.2101–2104.

Li, Z. and Arora, S. (2019) *An Exponential Learning Rate Schedule for Deep Learning*, arXiv preprint arXiv: 1910.07454.

Liang, Z., Zhang, G., Huang, J.X. and Hu, Q.V. (2014) 'Deep learning for healthcare decision making with EMRs', in 2014 *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, November, pp. 556–559.

Lipton, Z.C., Kale, D.C., Elkan, C. and Wetzel, R. (2015) *Learning to Diagnose with LSTM Recurrent Neural Networks*, arXiv preprint arXiv: 1511.03677.

Manning, C.D., Raghavan, P. and Schütze, H. (2008) 'XML retrieval', in *Introduction to Information Retrieval*, Cambridge University Press, California.

Miotto, R., Li, L., Kidd, B.A. and Dudley, J.T. (2016) 'Deep patient: an unsupervised representation to predict the future of patients from the electronic health records', *Scientific Reports*, Vol. 6, No. 1, pp.1–10.

Razavian, N., Marcus, J. and Sontag, D. (2016) 'Multi-task prediction of disease onsets from longitudinal laboratory tests', in *Machine Learning for Healthcare Conference*, PMLR, December, pp.73–100.

Schmidhuber, J. (2015) 'Deep learning in neural networks: an overview', *Neural Networks*, Vol. 61, pp.85–117.

Strack, B., DeShazo, J.P., Gennings, C., Olmo, J.L., Ventura, S., Cios, K.J. and Clore, J.N. (2014) 'Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records', *BioMed Research International*, Article ID: 781670, Vol. 2014, 11p [online] https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008 (accessed 14 June 2020).

Tran, T., Nguyen, T.D., Phung, D. and Venkatesh, S. (2015) 'Learning vector representation of medical objects via EMR-driven nonnegative restricted Boltzmann machines (eNRBM)', *Journal of Biomedical Informatics*, Vol. 54, pp.96–105.

Wickramasinghe, N. (2017) *A Convolutional Net for Medical Records*, Engineering in Medicine and Biology Society, pp.1–9.

Xu, R., Li, L. and Wang, Q. (2014) 'dRiskKB: a large-scale disease-disease risk relationship knowledge base constructed from biomedical text', *BMC Bioinformatics*, Vol. 15, No. 1, pp.1–13.