Outliers removal using BDR on neural network and deep learning

Yue Zhou

Research School of Computer Science
Australian National University
Canberra 2601 Australia
E-mail: u6682532@anu.edu.au

Abstract. The real-world dataset is always including a lot of outliers and noise data which would harm the classification model. To study whether outliers in the training set certainly affected the performance of the NN and RNN model. This paper implemented a simple neural network (NN) model and deep learning with the bimodal distribution removal (BDR) technique. BDR is a method for outlier detection (e.g. LMS, LTS). Our results suggest that with an empirical hyperparameter setting, BDR network model has a slight improvement on classification testing accuracy as well as a significant speedup of running time on large runs of training process than regular neural network model. However, BDR has no apparent impact on recurrent neural network (RNN), RNN has a better performance than NN with BDR.

Keywords: Dataset Pre-process; Bimodal Distribution Removal; Marking Prediction; Neural Network; Recurrent Neural Network

1. Introduction

Real-world dataset usually contains a certain amount of noise and random data. The dirty data points would have a negative impact on the characteristics of original data, skew the main pattern and sometimes mislead researchers to get a not so good result from it. A raw dataset is never a useful input for building a neural network model[1, 2]. In order to obtain a robust and high-performance NN model and RNN model, outlier detection and removal should be considered. The objective of this paper is to implement an outlier removal algorithm, BDR for the training set and compare the result with the result of non-removal outliers' model, then conduct evaluations on all models. Finally, to find out whether BDR is a boost to the performance of NN and RNN on the classification task.

1.1 Outliers Background

Dataset quality is always considered in machine learning and data mining fields. Throughout investigating and analyzing several real-world datasets, most of them are containing large gap between outliers and regular data clusters. The raw dataset pre-processing techniques are undergoing crucial transformative changes. One of the essential steps of pre-processing is removing outliers or noise data. Most data scientists or neural network specialists are likely to run into outliers. Hawkins [3] stated that outlier is a kind of observation locates far away from other observations; a different pattern most likely generates it. Huber[4] defines outliers in mathematic way that observations' values are outside the range of $\mu \pm 1.5\hat{\sigma}$, the $\hat{\sigma}$ is the estimated variance of the dataset. Aggarwal and Yu[5] make two definitions of outliers. One is outliers can be seen as noise data locating outside groups of well-formed clusters. The other one is locating outside clusters as well as separated from noise data. In this paper, we consider the noise and outliers are the same.

Outliers are produced and arise in different situations. Hodge and Austin[6] have a summary of outliers producing conditions. Commonly the outliers are recorded by objective reasons such as the mechanical faults and varies of system normal behaviour, as well as by subjective reasons including human error, records wrong digits. Furthermore, some data points are normal in several years ago but become outliers gradually because of the change of the system environment. Along with these points, outliers are inevitable in real datasets.

Outliers in the original dataset would affect the model accuracy and the estimated parameters, principally in statistical analysis field[7]. Studying the influence of outliers on the neural network model is an active topic. Khamis Azme, et al[8] conduct several experiments to investigate the impact of outliers on model performance. They tried several experiments with two methods, percentage-outliers and magnitude-outliers, to process the same dataset. They found that in training dataset when the percentage-outliers is lower than 15%, the outliers would have a minor impact on model accuracy. The model's performance decreases along with the percentage-outliers and magnitude-outliers both increases. When the dataset only contains small Gaussian noise, if the dataset has any outliers, the performance is poor. However, the LMLS method is more robust and stable with real data.

1.2 Outlier Detection Techniques

Liano[9] discusses two accessible error functions. The two different approaches are applied to study how outliers affect the neural network models. He states that Mean squared error (MSE) are usually selected measure in neural network modelling. MSE derives to least mean squares (LMS) measure and least mean log squares (LMLS), the latter bounds the value of influence function which can control the loss produced by outliers. He made a comparison between the two methods and

found that the LMS method works. Numerous outlier detection and removal mechanisms are developed in recent years. Slade and Gedeon[10] compared four outlier detection methods, including "Absolute Criterion" method which applies to minimize the lower absolute value of error, but that limits it we need to figure out how many outliers existed in the dataset in advance and easily get swayed. "Least Median Squares" method focuses on minimizing median of error, but it needs to do plenty of calculation to get converged. "Least Trimmed Squares" method applies to only minimize lowest mean square errors, this method has same shortage with "Absolute Criterion" and is also not good at processing real-world dirty dataset.
Slade and Gedeon[10] mainly introduced a new outlier detection method called BDR. This method is implemented and used in this paper. The frequency distributions of errors in the training set during training are approximately bimodal. The low error peak contained well-learned patterns of the network, while the high error peak still contained outliers. Patterns with error also located between the two error peaks. The error distribution indicated that the network could determine outliers itself. They concluded that BDR overcomes all other three methods' disadvantages and has a better performance on classification results.

1.3 Dataset Description

It is interesting to study a final mark prediction project which is students' most curious field; they are keen on their marks. Teachers use students' assessment marks to apply classification model to find out students' learning situation better and work out a more proper final exam paper. Students learn from the classification results not only whether they need to catch up with others but know what level he/she is among classmates in advance. We use a student mark dataset from the undergraduate students in The University of New South Wales to evaluate our pre-processing mechanisms and Bimodal Distribution Removal algorithm. The dataset is consisting of 153 records. Each record contains 11 attributes of a student's assessment marks. The dataset does not have good quality overall. Thus, further data pre-processing steps would be conducted in the following section.

1.4 Structure of Following Work

In summary, this paper is structured as follows:

1. We performed data preprocessing methods on original marking dataset.
2. We implemented a three-layer NN model and RNN model to run the final mark classification task.
3. We implemented the BDR technique and applied it to the training dataset.
4. We conducted several experiments to compare the models' classification results of applying this technique and not applying it. We also implement evaluation methods to examine each model's performance.

2 Method

In this section, We perform data wrangling on original marking dataset[11] and implement a simple neural network model. Then We implemented the outlier removal algorithm from Slade and Gedeon[10].

2.1 Pre-processing Dataset

The original dataset has a low quality and missing plenty of data points (325 missing values and 7 students without any marks in total) which would have a substantial negative impact on the classification task. Then data wrangling methods were applied before building input features for the NN model and RNN model. [12] introduced how to handle missing values. Considering all values are numerical marking value, so all missing values are replaced with mode value corresponding to their attribute's column. For example, the mark under column "lab2, missing marks are replaced with mode value calculated by all marks of "lab2". Replacing value can also be mean, median value, they have a similar impact on the dataset under a simple pre-processing principle. The noise data points are imported into a dataset using this method. Other better handling missing values are not considered in this paper, because the whole dataset is too small, and it is not worth to perform a computationally complex measure. The student and course information attributes are dropped directly, whereas they are not irrelevant to the following classification model. Then data normalization has been processed. Sola and Sevilla[13] conduct a backpropagation neural network to problems of estimation and identification. Base on the huge training data, they found that if normalized the input data with specific criteria before training, the modelling results are gaining a significant improvement on performance, besides the computation process is much faster, compared to without normalization. They state two advantages of performing normalization, reduction of estimation errors in a factor between 5 and 10, and the training process computation time is reduced in one order of magnitude under the same result. Considering the quantity and quality of dataset, min-max normalization method is applied in this paper. It is easy to implement and boost to obtain a better result. The normalized value z is calculated by below formulation, where min and max are the minima and maximum values in x given its range.

$$z = \frac{x - min(x)}{max(x) - min(x)}. \quad (1)$$

Next step is manually labelling the final marks. The marks are divided into four categories. For mark above 75 which is a D is labelled as 0, a mark between 65 and 74 which is a C is labelled as 1, a mark between 50 and 54 which is a P is labelled as

2 and mark less than 50 which is a F is labeled as 3. After the category, the dataset has a relatively balanced data distribution where from P to D, the size of each category is 22, 31, 66 and 34. Then the dataset is divided into two parts, training set and testing set. Each set is randomly selected from 153 records. Foody et al.[14] discussed the size of the training set can greatly affect the artificial neural network performance based on the classification task, they concluded that when the size of the training set is relatively small, the accuracy of the classification model is better in general. In this paper, the training set size is 80%, and the testing set is 20%.

2.2 NN Model

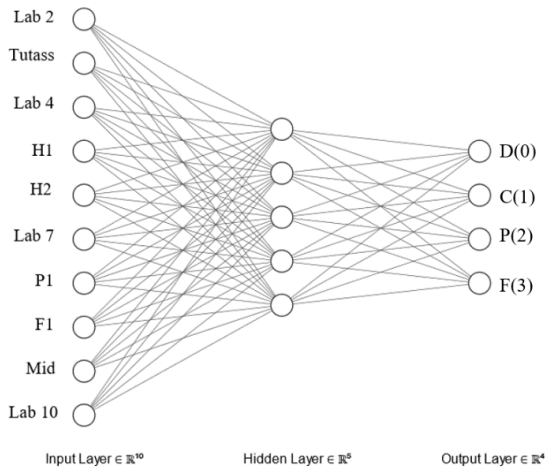

Fig. 1 Three-layer NN with 10 input neurons, 5 hidden neurons and 4 output neurons

The task of the three-layer neural network is predicting the final exam mark and classifies the overall final marks into four degrees. In Fig. 1, this model has 10 input features which are well pre-processed assessment marks, 5 hidden neurons and 4 output neurons corresponding to 4 grade degrees. The number of hidden neurons should not be too large to avoid computation and overfitting problem. Because the model's predicting output values are probability values between 0 and 1, the sigmoid function is a good choice as an active function.

$$g(z) = \frac{1}{1+e^{-z}}. \tag{2}$$

Then the difference between the desired output and the actual output should be measured. This model is a multiclass classification model, so the cross-entropy loss function below is applied. Where M represents the number of classes, y represents binary indicator if class label c is correct or not for observation o, p represents the predicted probability observation o is of class c.

$$L = -\sum_{c=1}^{M} y_{o,c}\, log\left(P_{o,c}\right) \tag{3}$$

For measuring the performance of the network on different categories, this paper runs the training process between 1 and 500 times to obtain the average training accuracy and testing accuracy, in case of randomness impacts the result. This main idea is similar to Cross-validation method. Running the model many times can well measure the performance of the model on the new dataset, it can also reduce the impact of overfitting. This method randomly selects out training set and testing set at each run, so it will nearly cover each record, which means each of them can be the input data and target data to get the most information from them.

2.3 RNN Model

The idea behind RNN is to use sequence information. In traditional NN, all inputs (including outputs) are assumed to be independent of each other. For many tasks, it is a terrible assumption. For example, in natural language processing task, if we want to predict the next word in a sequence, it is better to know which words precede it. The reason why RNN loops are because it performs the same operation for each element in the series and each operation depends on the previous calculation results. In other words, the RNN memorizes the information that has been calculated so far[15]. Numerous researchers are studying RNN in different fields. For example, information extraction from unstructured clinical text[16]; stock price prediction[17], network traffic prediction[18] and sentimental texts classification[19].

In this paper, we build a simple RNN model. In Fig. 2 The input, output features are all the same with the NN model. For the number of hidden neurons, we followed a rule of thumb, where N_i is the number of input neurons, N_o the number of output

neurons, N_s the number of samples in the training data, and α represents a scaling factor (usually between 2 - 10), so we decided hidden neurons are 5. Because the objective of the task using RNN is similar to using NN, we also chose sigmoid function as active function. The loss function is cross-entropy too.

$$N_h = \frac{N_s}{(\alpha \times (N_i + N_O))} \tag{4}$$

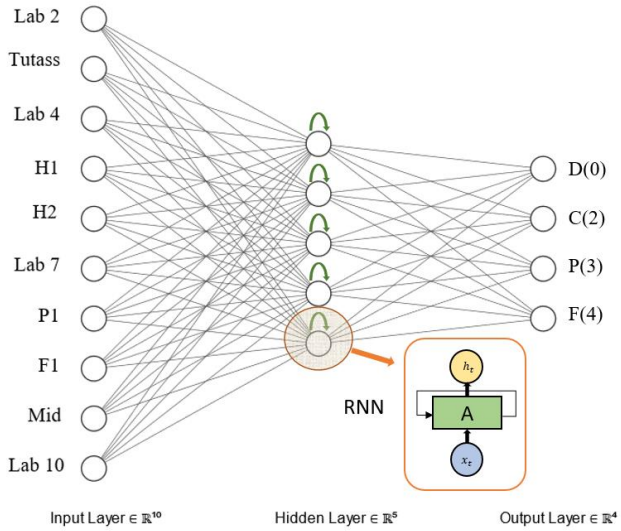

Fig. 2 RNN with 10 inputs, 1 hidden layer and 4 outputs

2.4 Outliers Removal

Next, the Bimodal distribution removal[10] is implemented in BIMODAL DISTRIBUTION REMOVAL (Algorithm 1)and applies during the training process. Note that the α is a parameter which can be set between 0 and 1, T is the original training set, T_1 is a new training set without outliers. The basic idea of BIMODAL DISTRIBUTION REMOVAL is to reduce the impact of larger error peak among the training set. This algorithm has a start condition which is until two error peaks have formed, and it is repeated every 50 epochs. Whereas each repeat has a new training set T_1, the outliers are removed gradually. The algorithm also has a halting condition because it removes outliers in the training set. Eventually, the size of the training set will become smaller and smaller, which might cause overfitting problem. Therefore, it stops when ν_{ts} is less than a constant; this paper sets it as 0.01. BDR has several advantages over other common use outlier handling methods, such as LMS and LTS. The essential benefit is to let the dataset be the "decision-maker". It identifies outliers, decides the number of outliers should be removed, makes full use of every data point, including outliers, controls the training process in a relatively short time and overcomes overfitting all by itself.

Algorithm 1 BIMODAL DISTRIBUTION REMOVAL

Input: T
Output: T_1
if $\nu_{ts} < 0.1$ **then**
  $\overline{\delta_{ts}} \leftarrow Norm(T.var)$
  for $i \in T$ **do**
    if $e_i > \overline{\delta_{ts}}$ **then**
      sub ← i
    end if
  end for
  $\overline{\delta_{ss}} \leftarrow sub.mean$
  $\sigma_{ss} \leftarrow sub.std$
  for $i \in T$ **do**
    if $e_i < \overline{\delta_{ss}} + \alpha\sigma_{ss}$ **then**
      $T_1 \leftarrow i$
    end if
  end for
end if
return T_1

3 Results and Discussion

Firstly we implement two NN. Model A is implemented with BDR algorithm, and Model B is not implemented any outlier removal algorithm. Then we build two RNN model under a similar pattern as former. We run each model for specific runs and obtain all results in Table.1 and Table.2. A pair of NN model and a pair of RNN model has the same hyperparameters set correspondingly; the epoch number of NN is 500, the RNN is 200.

Table 1. Comparing two NN models with their training accuracy and testing accuracy

Rounds	Training set %		Testing set %	
	BDR	NN	BDR	NN
1	52.234	60.084	50.206	58.424
10	83.098	85.485	59.039	59.788
50	87.747	93.825	59.976	62.747
100	93.457	95.369	62.635	59.467
500	93.807	97.543	67.864	65.569

From Table.1, when the rounds are under 50, the testing accuracy of NN is higher than testing accuracy of BDR model, it shows that the performance of BDR method is influence by the dividing method of the original dataset. Runs under 50 represent that the training set and testing set may have bias and are not fully used. When the rounds increase to 100 and 500, the average testing accuracy values are more stable and reliable. The average testing accuracy of BDR model is 67.864% which is slightly larger than testing accuracy of NN 65.569%. 2.295% accuracy improvement, which proofs the outlier removal like BDR methods will boost the NN's performance. Compared with Choi and Gedeon[11] results, they had an average testing accuracy of 58.8% with extracted rules and average testing accuracy of NN 53.8%. The NN's performance of this paper is better than Choi and Gedeon's model, which is 11.769% improvement in testing accuracy. Due to the limitation of neural network optimizing methods and computation ability of computers then, the improvement is modern neural network model's achievement.

From Table.2, the average testing accuracy of RNN is higher than RNN with BDR. When the rounds are under 50, the difference is around 1%. When the rounds up to 100, the difference is rise to 3%-4%. The results indicated that the BDR technique might have a negative impact on the classification accuracy of the RNN model. This outcome might state that RNN is a state-of-art deep learning network; the BDR is no longer a necessary technique to boost it.

Table 2. Comparing two RNN models with their training accuracy and testing accuracy

Rounds	Training set %		Testing set %	
	BDR	RNN	BDR	RNN
1	57.401	59.152	57.476	58.061
10	86.192	93.181	62.594	63.972
50	94.694	97.261	67.722	68.776
100	93.017	98.079	70.764	73.530
500	95.756	98.858	71.229	75.325

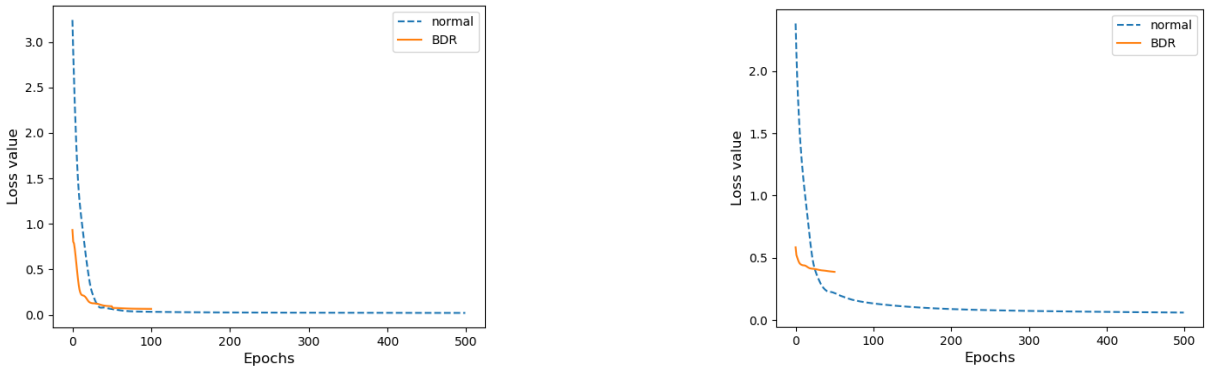

Fig. 3. The loss values under 500 rounds (left a) and 50 rounds (right b) of BDR and NN

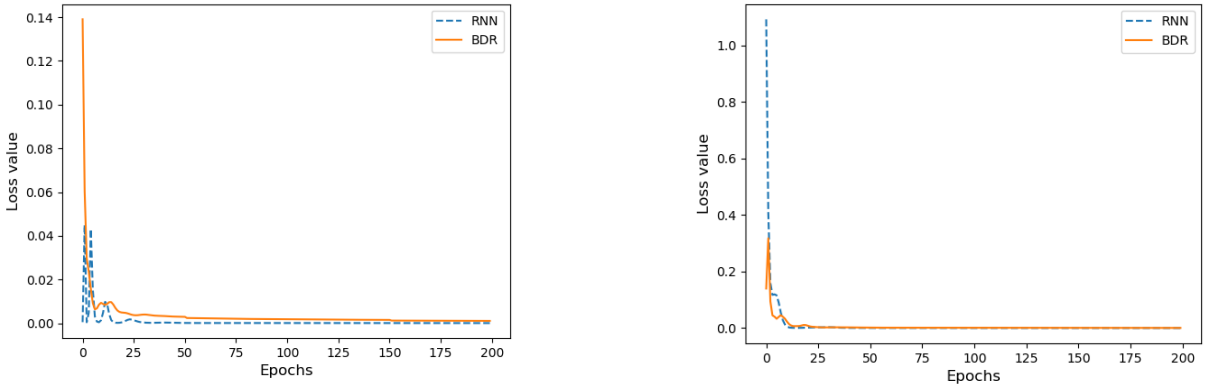

Fig. 4. The loss values under 50 rounds (left a) and 100 rounds (right b) of BDR and RNN

Fig.3a was the loss value at each epoch with a separate model, and both models had conducted 500 rounds of the training process. Each model obtains a local minimal loss value when the epoch is smaller than 50. Then both models have a decreasing smoothing curve. The BDR model stops earlier around 101 epochs, the early stop condition for BDR method occurred. A similar result in Fig.3b which is conducted 50 rounds. The loss value of the normal model is lower than BDR model at each curves' end. The corresponding situation was that the testing accuracy of BDR is lower than the normal model at runs 50 in Table 1. Fig.4a and Fig.4c showed that the loss value against epochs at 50 rounds and 100 rounds. When the epoch comes to larger than 50, the BDR and RNN model has a similar pattern which is a nearly smoothing horizontal line.

Then, Table.2 shows that BDR has a significant improvement in training process time when the number of runs is large. When it runs up to 500, BDR can save 55.76% training time. Training time results verified that BDR can mostly speed up training through removing outliers to optimize training set size at each epoch. At the same time, it has an early stop mechanism, terminates training at a proper time to improve the model's overall performance. Table.3 also verified the conclusion. During the massive rounds, the average run time of BDR is lower than RNN. Furthermore, the running time of RNN is much higher than the running time of NN. Due to RNN performed much more computations in the hidden layer during the training process.

Table 2. The training time of BDR and NN comparison

Rounds	BDR run time (s)	NN run time (s)
1	0.41	0.37
10	5.52	5.86
50	25.36	27.14
100	35.68	53.11
500	119.61	269.24

Table 3. The training time of BDR and RNN comparison

Rounds	BDR run time (s)	RNN run time (s)
1	0.75	0.70
10	8.02	7.50
50	34.41	36.77
100	61.67	80.43
500	129.469	407.21

4 Conclusion and Future Work

The classification results for NN are shown that when NN implements extra outlier removal measures such as BDR have a better performance on testing accuracy than NN. BDR testing accuracy is 2.295 % more than NN testing accuracy. Although the improvement is not distinct, Handling outliers is still a considerable optimization method for NN. Outliers among input dataset and target dataset distort data distribution and reduce the model classification performance. However, the classification results for RNN and BDR presented that BDR is not an ideal technique to improve the RNN performance; it might harm the classification accuracy of RNN on the contrary. The overall accuracy is improved from 67.864% (NN with BDR) to 75.325% (RNN), which is 11.04% improvement of performance.

In future, we will conduct advanced data pre-processing method on the mark dataset, including replacing missing values with more suitable values and drop empty records to obtain a more comprehensive "ready to train" dataset. A proper way to tune hyperparameters needs to be considered. In this paper, the parameters settings are mostly assigned as empirical values, such

as learning rate, epoch, and hidden neurons. If carefully tuned the hyperparameters, the NN and RNN model performance might obtain a good improvement. We will also experiment on different size of the real-world dataset to investigate whether the BDR algorithm works well on medium size and large size dataset too. Other advanced deep learning models (e.g. LSTM) would be studied to further improve the prediction accuracy. Finally, following the findings of this paper, it would be beneficial to analyze how other state-of-art outlier detection methods compared to BDR.

5 References

1. Beccali, M., et al. Influence of raw data analysis for the use of neural networks for win farms productivity prediction. in 2011 International Conference on Clean Electrical Power (ICCEP). 2011. IEEE.
2. Nelson, M., et al., Time series forecasting using neural networks: Should the data be deseasonalized first? 18(5): pp. 359-367 (1999).
3. Hawkins, D.M., Introduction, in Identification of Outliers, Springer, Dordrecht (1980).
4. Huber, P.J., Robust statistics. Vol. 523. John Wiley & Sons (2004).
5. Aggarwal, C.C. and P.S. Yu, Outlier Detection for High Dimensional Data, in Proceedings of the ACM SIGMOD Conference 2001. pp. 37-46 (2001).
6. Hodge, V.J. and J. Austin, A Survey of Outlier Detection Methodologies. The Artificial Intelligence Review. 22(2): pp. 85-126 (2004).
7. Rousseeuw, P.J. and A.M. Leroy, Robust regression and outlier detection. Vol. 589. John wiley & sons (2005).
8. Khamis, A., et al., The Effects of Outliers Data on Neural Network Performance. journal of Applied Sciences. 5(8): pp. 1394-1398 (2005).
9. Liano, K., Robust error measure for supervised neural network learning with outliers. IEEE Transactions on Neural Networks. 7(1): pp. 246-250 (1996).
10. Slade, p. and T.D. Gedeon, Bimodal distribution removal, in International Workshop on Artificial Neural Networks. Springer: Berlin, Heidelberg. pp. 249-254 (1993).
11. Choi, E.C.Y. and T.D. Gedeon, Comparison of extracted rules from multiple networks., in Proceedings of ICNN'95-International Conference on Neural Networks. IEEE. pp. 1812-1815 (1995).
12. Bennett, D.A.J.A. and N.Z.j.o.p. health, How can I deal with missing data in my study? 25(5): pp. 464-469 (2001).
13. Sola, J. and J. Sevilla, Importance of input data normalization for the application of neural networks to complex industrial problems. IEEE Transactions on Nuclear Science. 44(3): pp. 1464-1468 (1997).
14. Foody, G.M., M.B. McCulloch, and W.B. Yates, The effect of training set size and composition on artificial neural network classification. International Journal of Remote Sensing. 16(9): pp. 1707-1723 (1995).
15. Medsker, L.R., L.J.D. Jain, and Applications, Recurrent neural networks. 5 (2001).
16. Jagannatha, A.N. and H. Yu. Structured prediction models for RNN based sequence labeling in clinical text. in Proceedings of the conference on empirical methods in natural language processing. conference on empirical methods in natural language processing. 2016. NIH Public Access.
17. Selvin, S., et al. Stock price prediction using LSTM, RNN and CNN-sliding window model. in 2017 international conference on advances in computing, communications and informatics (icacci). 2017. IEEE.
18. Vinayakumar, R., K. Soman, and P. Poornachandran. Applying deep learning approaches for network traffic prediction. in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). 2017. IEEE.
19. Tang, D., B. Qin, and T. Liu. Document modeling with gated recurrent neural network for sentiment classification. in Proceedings of the 2015 conference on empirical methods in natural language processing. 2015.