Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 4720539, 11 pages
https://doi.org/10.1155/2022/4720539

Research Article

Impact of Hyperparameters on Deep Learning Model for Customer Churn Prediction in Telecommunication Sector

Anouar Dalli

Ecole Nationale des Sciences Appliquées de Safi (ENSAS), Université Cadi Ayyad, Marrakesh, Morocco

Correspondence should be addressed to Anouar Dalli; anouar_dalli@yahoo.fr

Received 21 October 2021; Accepted 21 January 2022; Published 9 February 2022

Academic Editor: Venkatesan Rajinikanth

In this paper, in order to predict a customer churn in the telecommunication sector, we have analysed several published articles that had used machine learning (ML) techniques. Significant predictive performance had been seen by utilising deep learning techniques. However, we have seen a tremendous lack of empirically derived heuristic information where we had to influence the hyperparameters consequently. Here, we had demonstrated three experimental findings, where a Relu activation function was embedded and utilised successfully in the hidden layers of the deep network. We can also see that the output layer had the service ability of a sigmoid function, in which we had seen a significant performance of the neural network model and obviously it was improved. Furthermore, we had also seen that the model's performance was noticed to be even better, but it was only considered better though when the batch size in the model was taken less than the test dataset's size, respectively. In terms of accuracy, the RemsProp optimizer beat out the other algorithms such as stochastic gradient descent (SGD). RemsProp was seen even better from the Adadelta algorithm, the Adam algorithm, the AdaGrad algorithm, and AdaMax algorithm as well.

1. Introduction

Customer churn is defined as when customers stop doing business with a company [1]. This phenomenon directly affects banks, insurance companies, video game companies, and telecommunication companies [2].

In recent years, the telecommunication industries undergo enormous changes such as new services, technological enhancement, and increased competition [3]. Therefore, predicting customer churn in the telecommunication sector becomes an essential parameter for industry actors to protect their loyal customers [4].

Recently, there is a tremendous increase to apply ML techniques to predict customer churn in different industries [5, 6]. Using these techniques with data obtained from the telecommunication operator's customers generates regular models that can demonstrate causality and correlation, addressing customer churn challenges in telecoms.

In this study, we had collected various published articles that were using ML algorithms and techniques for customer churn prediction in the widely used telecommunication sector. The ML techniques used for this problem can be classified into three categories:

(1) Traditional ML techniques

(2) Ensemble learning

(3) Deep learning

The traditional ML techniques include random forest [7–9], logistic regression [8, 10], support vector machines [11–13], and Naive Bayes [14].

In Ensemble learning, we have seen the principle that the combination of two classifiers is even better than a single classifier, and this improves the predictive performance of the proposed model [15]. The ensemble learning uses a voting system to obtain the results from the classifiers, while these classifiers can be based on traditional ML algorithms [16, 17].

Deep learning has recently been widely used to predict churn; however, here, the process in which we have to select training hyperparameters for the churn modelling is considered as time consumption, which makes it challenging for researchers and practitioners for further improvement on the research side [18, 19]. As a result, only a few studies have looked at the effect of hyperparameters on deep learning models' effectiveness in predicting the churn rate in the telecommunication industry. As a result, there seems to be a little empirical basis to understand how hyperparameters affect the performance of the deep learning model when it is used to predict the churn rate. So, empirically derived heuristic knowledge to aid hyperparameter selection is still insufficient.

In this research, we will investigate how hyperparameters affect the performance of deep learning models in predicting churn in the telecommunication industry.

In the current study, our contribution can be summarized as follows:

(i) The impact of different combinations of activation functions on the performance of the NN model

(ii) The impact of several batch sizes that was used on the NN model performance, respectively

(iii) The impact of the different optimizers on the performance of the NN model

The structure of this paper is as follows: Section 2 presents a brief literature review of the published articles that had used all the different ML techniques and algorithms to successfully predict the customer churns in the widely considered and used telecommunication industry. Then, Section 3 elaborates us with the methodology of this study which would include the different choices of the dataset, such as preprocessing carried out on the considered dataset as well as the hyperparameters considered, which might be having a detail. Section 4 would enlighten us with all the experimental results been used and captured, and the model results have been analysed throughout in Section 5. Finally, this article ends with conclusions and suggestions for future work.

2. Literature Review

2.1. Traditional ML Techniques. The Bayes' theorem is used to create the Naïve Bayes (NB) classifier. The classifier principle is the assumption that the existence of a feature for a class is unrelated to the existence of other one [20]. For example, customers might be unsubscribing as if they do not have a contract and have not subscribed to any service.

The principle of the logistic regression (LR) algorithm works on the premise of linking the happening or non-happening of an event at the level of independent variables. In our case, the "customer churn" event is a binary variable that is produced from the explanatory variables, namely, the demographic variables of the customer and his data concerning the services [21].

Formulated on statistical learning theory, the support vector machine (SVM) algorithm can optimally separate two classes of objects (for example, retained and churners' clients) and separate simply across the peer group of a multivariate hyperplane. Recently, various researchers used the SVMs because of remarkable benefits such as good generalization ability and a small number of control parameters. However, this strategy is difficult to explain in terms of input properties, and it may not function well in cases with overlapping classes [12].

Random forest (RF) is based on the assembly of decision trees. It is intuitive to understand, quick to train, and produces some interesting results. Therefore, it is a popular algorithm. The basic idea of this algorithm is rather than having a complex estimator capable of doing everything; the RF uses several simple estimators (several decision trees). Each decision tree has a fragmented view of the problem. Then, all these estimators are brought together to obtain the global vision of the problem. It is the assembly of all these estimators which make the prediction extremely efficient [16].

The K-nearest neighbor (KNN) learning algorithm is used in supervised learning applications, for example, regression and classification. The class label for an event in this case is decided by how similar the event in the training set is to its neighbours. All of the events have been stored in this case, and the class label was determined by examining the KNN at the time of categorisation [22].

2.2. Ensemble Learning Techniques. Ensemble learning [23] is a process by which multiple models are strategically generated and integrated to solve computational intelligence problems. It uses multiple classifiers to achieve better prediction performance than a single classifier and has been applied to a variety of subjects in classification and regression. Some common ensemble learning techniques are AdaBoost, Bagging, and Random Subspace.

One of the most common boosting algorithms is named as "AdaBoost" algorithm. The main objective of the AdaBoost algorithm is to create a classification scheme with better-predicted results by focusing on difficult-to-classify data points. At the start of the AdaBoost algorithm, every pattern in the training set is assigned with the same weight. Weight values are increased for misclassified instances, and weight values are decreased for all correctly classified instances in the pattern weight values. Additionally, weight values for weak classifiers have been assigned accordingly. The sample distribution is modified in the AdaBoost algorithm, which eliminates some of the mistakes from previous iterations [24].

The Bootstrap aggregating or bagging algorithm is an ensemble learning technique. The bagging algorithm uses the bootstrap distribution, which would generate various base learners [17]. By collecting sampling training instances and replacement, a diversity among base learning algorithms is obtained in this technique. Aggregating the base learning algorithm prediction is trained on a varied training subset by majority voting or weighted majority voting. With small datasets, the bagging algorithm can produce promising results. Furthermore, it can improve the unstable base learning algorithm prediction performance.

The random subspace algorithm is an ensemble learning method that uses feature set manipulation to achieve diversity across the ensemble's base learning algorithms. The randomly selected modified subspace features are used to train base learning algorithms in this technique. Overfitting may be avoided using this approach, which improves prediction performance. The random subspace approach can give effective and efficient solutions for datasets with numerous duplicated features [15].

2.3. Deep Learning Techniques. Deep learning architectures have previously been employed in a variety of applications, including computer vision, pattern recognition, and natural language processing [18]. Multilevel feature representations may be learned using deep teaching methods. The designs focus on finding learning models that are built on numerous levels of hierarchical nonlinear information processing. Deep learning is a subfield of ML. However, due to an increase in its importance, it turns into a unique academic field.

Deep learning simulates the structure of the human brain in learning information using the neural network (NN). For a human being, we know that new things are frequently learned easily only by training the biological neurons in the brain, but here, training the mind with the help of examples is even more crucial. Because when a human memorizes something with the help of examples, it is stored in the long-term memory permanently. Now, in the neural network, we depict the same human mind's phenomenon where we input a considerable amount of data (x_i), and in which we use to train all the neurons, and then the interconnected network is adjusted accordingly to have a better response from it, or we can say to get more accurate and pure output predicted values. We should keep in mind that to get better output results, the researchers always recommend adjusting or updating the weights to the neural network accordingly (w_{ij}). These weights are adjusted for each bias and neuron to get a better result, which are the adder for each summation procedure (see Figure 1). Moreover, the complexity of the network is surely determined by the number of layers used in the neural network [25].

In addition, for the output (S_j), we must transform the activation function as (AF) which would then form the output as (O_j). Here, it will then lately transmit to the next layer of the neural network that had been designed. Nevertheless, from the literature, we had seen that there may be numerous types of activation functions used whose work is to bring the nonlinear property to the input signal. This would accommodate a several different number of instances. Therefore, it would then result in a highly adaptable network.

2.4. ML Techniques for Churn Prediction. Many researchers employed traditional ML methods to design a customer churn model such as logistic regression, decision tree, and support vector machine (SVM). According to the current studies, these techniques improve the prediction performance according to current studies. For example, Ebrah and Elnasir [24] applied the algorithms (Naive Bayes, support vector machine, and decision tree) to train the dataset to predict customer churn for a telecommunication operator and found that the support vector machine algorithm had better performance. The work of Brandusoiu and Toderean [13] also confirmed the effectiveness of these techniques using logistic regression and SVM. Moreover, Tamaddoni et al. [22] also carried out a comparative study of the precision of models based on traditional ML techniques to predict customer churn in the telecommunication industry. Khodabandehlou and Zivari Rahman [9] had also used SVM in their work. They had compared this algorithm to a few other models ultimately for the churn's prediction.

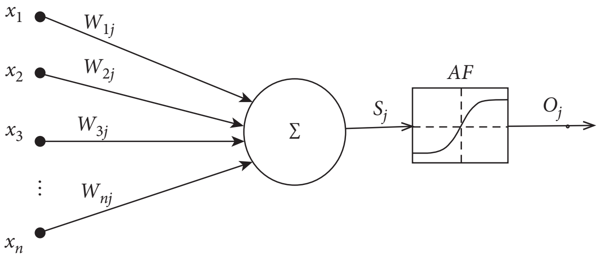

Figure 1: Neuron structure.

The ensemble learning can be seen that has also been used as a part to predict potential churn rates only by using standard ensemble learners, as explained in [26]. Baumann et al. [27] had also offered a tremendous work where the combination of a widely used support vector machine (SVM) method and adaptive boosting known as "AdaBoost" has widely been used to an operator's dataset. This would then detect clients with a strong probability of leaving the operator and moving out to another operator's network. Here, we would mention that this association has wonderfully been able to finally achieve a remarkable and tremendous precision of classification to be precise. In the work of Vafeiadis et al. [7], the authors had successfully compared boosted versions where they had used the AdaBoost algorithm and unboosted versions of the very traditional classifiers such as SVM classifies and decision trees classifier, which would later find the very best performance result among them. Now, to get in-depth details, if we open the University of California's ML repository, a "Churn Prediction Dataset" would be seen as available; they had successfully succeeded in achieving an accuracy of 96.86% at once for their published work. In that published work, the Boosted SVM (or SVM polynomial kernel using AdaBoost) was considered the best technique ever used. The conclusion was provided by the authors that when the amplification technique was applied to the work over the previously used traditional classification methods, it had given a greater precision value. The work of Xiao et al. [28] used ensemble learning to build a prediction model and applied it to operator Orange and Cell2Cell datasets. They combined basic classifiers such as Random forest and KNN to get a majority vote and could predict the future instance more accurately.

Recently, there has been a great interest in deep learning in several areas, including customer churn prediction in recent years [28, 29]. For example, in the work of Zhang et al. [25], it has been shown that NN gives better results than traditional ML techniques such as logistic regression, decision tree, and ensemble learning. Sharma and Panigrahi [30], for the customer churn problem in the telecommunication field, used the NN to obtain an accuracy of 92.35% without using any preprocessing or sampling method. Azeem et al. [31] also built a model based on deep learning. This model was able to predict customer attrition with better probability than a traditional ML model. When Alboukaey et al. [32] evaluated the classical ML model and the deep learning model for the customer churn problem in the telecommunication industry, they came to a similar conclusion.

Thus, we can confirm that deep learning represents a robust and effective solution to the problem of customer churn's predictions in the telecommunication industry. Also, this technique produces significant results on several datasets. However, while the usage of NNs has been established, we have noticed that a very minimum research work has been investigated on the impact of different hyperparameter's selection, also on how to tune the NN performance for a better network. The same finding as mentioned before has also been published and confirmed in some other relevant papers that had also used the same deep learning methods to predict customer's churn in the business of the telecommunication field, and it can be seen in [33, 34]. Thus, we can gladly inscribe that the approaches which are used to configure the hyperparameters in a neural network where NNs are used for modelling the churn rate seem to be still lacking on a larger scale in the telecommunication sector, and this work has widely been explored in this article, respectively.

3. Method

3.1. Dataset. Usually, the customer churn datasets are widely collected from all different telecommunication operators. Now, we surely know in advance that these types of datasets do have a very different number of subscribers (users) and attributes (functionality) when used in the network. If we tear down it into subsections, it mostly contains personal customer data, billing-plan info, and call durations.

We know that all those datasets used in different studies are surely not available publicly most of the time. Or mostly, the company also does not mention its name, while the generic term is labelled such as "Turkish Telecommunication Company Dataset," where some anonymous keywords such as "anonymous (Anon)" and some widely publicly datasets are used.

We have considered an open-source database [35] dataset of our customer's churn which contains 21 characteristics and 3,333 observations. Here, the "Churn" feature is either the churn or the non-churn rate of those customers based on the other aspects. Approximately, 14.5 is the percent of "Churn," whereas 84.5 percent is "non-churn." The dataset's features are listed in Table 1. In this experiment, there were 2,666 (or 80%) and 667 (or 20%) occurrences in the training and test data, respectively.

Table 1 describes 21 parameters (1 dependent variable and 20 independent variables) of the dataset.

3.2. Data Preprocessing. In a data preprocessing technique, all missing values and all the outliers and the non-numeric feature values are considered as common features. Likewise, the disparate feature scales, all the imbalanced classes, and all the irrelevant features are also considered as common features in real-world churn datasets, as explained in [34]. Moreover, the preprocessing in a network on the data is therefore a very important step considered in improving the accuracy of customers' churn prediction. We have provided six methods to preprocessing data before sending it to train data in this article.

(i) Missing value: depending on the degree of functionality that is missing, we have devised several solutions for missing values. Now, we already know that all those features were removed from the dataset which had more than 95% missing values in the data that we had considered. Here, all those identified missing values have been injected as the mean and column mode successfully for the numeric and the categorical features for the remaining dataset, respectively.

(ii) Outliers: data points that are notably different from other data points are thought to skew predictive performance [36].

(iii) Category variable: now, we know that because of many advanced ML algorithms used nowadays, needs a binary input where a categorical variable has safely been transformed to binary variables by using the dummy variable approach. This can produce (n–1) dummy variables at once, where (n) is the number of distinct values in categorical variables.

(iv) Standardization: the overall number of calls, for example, may be thousands, while the total call time in minutes could be millions. Therefore, the MAX-MIN scaling approach [36] was used to uniformly standardise the data in this investigation, as shown in the following equation:

$$x_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}, \quad (1)$$

where min (x) is the less or minimum value in x, and max (x) is the more or maximum value in x.

(v) Class imbalance: typically, data tests and the churn variable class are unbalanced, i.e., the churner ratio is substantially lower than the non-churner ratio. To balance these two classes, the training dataset was subjected to the synthetic minority oversampling method (SMOTE) [37], which increased the size of the degraded or minor class; in this case, the churn, as the size of the majority class, which was not churned, was raised by random churn sampling.

Table 1: Descriptions of the dataset parameters.

Feature name	Description
Id_state	State
Acc_length	Account used days
Code_Area	Phone area code
Num_Phone	Customer phone number
Plan_international	Whether the customer starts an international business
Plan_vmail	Whether the customer starts the voice mail service
Num_v mail_messages	Number of customer vmail messages
Tot_day_minutes	Total minutes of talk during the day
Tot_day_calls	Number of calls in the day
Tot_day_charge	Call charges during the day
Tot_eve_minutes	Total minutes of talk last night
Tot_eve_calls	Number of calls last night
Tot_eve_charge	Charges for calls last night
Tot_night_minutes	Night total call minutes
Tot_night_calls	Total number of calls in the evening
Tot_night_charge	The total charge for calls at night
Tot_intl_minutes	Total minutes of international business calls
Tot_intl_calls	Total number of international business calls
Tot_intl_charge	Total charge of international business calls
Calls_custmr_service	The number of calls for customer service
Churn	Customer churn

(vi) Feature selection: when we consider the feature selection process, there, the final step in data preparation is to finally choose the best features. However, we would eliminate the highly correlated features since they would typically increase the computational workload without extra efforts and without improving prediction ability at all.

Following these steps, ten standardised values were kept and utilised as input to the NN. The last column (Churn) on the other side of the NN was utilised to train the model, categorising them as churners and non-churners.

3.3. Performance Evaluation. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions is summarized with count values by each class, as shown in Table 2. The confusion matrix shows how our classification model is confused when it makes predictions. It gives us insight not only into the errors made by a classifier but, more importantly, the types of errors.

We can explain TP, FP, TN, and FN as follows:

(i) True positive (TP): if the predicted "churn customer" is actually "churn customer," the prediction is TP

(ii) False positive (FP): if the predicted "churn customer" is real news, the prediction is FP

(iii) True negative (TN): if the predicted "no-churn customer" is actually "no-churn customer," the prediction is TN

(iv) False negative (FN): if the predicted "no-churn customer" is actually "churn customer," the prediction is FN

Classification accuracy is the number of correct predictions made as a ratio of all predictions made, as shown in the following equation [33]:

$$\text{Accuracy} = \frac{TN + TP}{TN + TP + FN + TP}. \quad (2)$$

It is the most common evaluation metric for classification problems. We have used this metric to evaluate our model in order to study the impact of the different hyperparameters.

3.4. NN Implementation. LeCun et al. [38] developed best practices for training the NN model. The approach that we will follow to train this model is mainly based on this work. The steps are summarized, as shown in Figure 2.

The structure of the NN is made up of a 10-node input layer. In the first hidden layer, every node is linked to every other node. This hidden layer has six nodes. In the neural network model used in a network, the second hidden layer is then safely connected to the single considered output layer which would then produce a binary output value.

As a result, the NN architecture is considered to have a 10-6-6-1 architecture for its model. The pre-processed data was surely inserted at the time of the input layer. Later, the produced data have then additionally been scaled and had sent to the layers in the batches to produce the result data for the output. The batch size is a hyperparameter that controls how many samples are processed by the NN at each epoch.

The activation function of each layer's node is another hyperparameter that regulates the NN's nonlinearity. These functions limit the output to a certain range or a certain threshold.

Table 2: Confusion matrix.

Confusion matrix		Actual classes	
		Churn customer	No churn customer
Predicted classes	Churn customer	TP	TN
	No churn customer	FP	FN

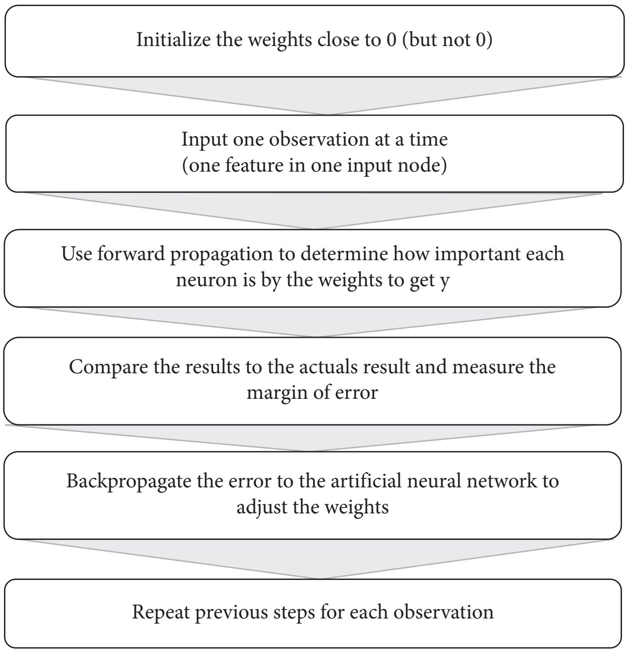
Initialize the weights close to 0 (but not 0)
Input one observation at a time
(one feature in one input node)
Use forward propagation to determine how important each neuron is by the weights to get y
Compare the results to the actuals result and measure the margin of error
Backpropagate the error to the artificial neural network to adjust the weights
Repeat previous steps for each observation

Figure 2: NN implementation steps.

We evaluated the influence of hyperparameters on a deep learning model for customer churn prediction in the telecommunication sector using the methodology of Domingos et al. [39].

3.5. Activation Function. There are three main functions of activations widely used in NN; the description is presented below:

The most used function was sigmoid (value between 0 and 1). This function provides probabilities that could be useful in classification problems with a binary output [40].

The sigmoid function is compatible with the problem of churn prediction because the model can be configured in such a way to have a threshold from which a client is declared “churner” or “non-churner.” The expression of the sigmoid is as follows, and its shape is given in detail in Figure 3:

$$\varphi(z) = \frac{1}{(1 + \exp(-z))}. \tag{3}$$

The Relu function (The rectified linear unit) has a real value, and its threshold is 0. This function replaces negative values with zero, as shown in Figure 4.

The Relu activation function can be used in the case when the input values are negative. The model is configured to manipulate the actual positive scaled values.

$$R(z) = \max(0, z). \tag{4}$$

The hyperbolic tangent (tanh) value is considered as a real number that lies between the range of −1 and 1, as shown in Figure 5. We can claim that this tangent value is considered as a very useful function value that can be used in the hidden layer in a neural network. Since the negative values are not often been scaled onto the used functions as to zero, the tanh expression is

$$\tanh(x) = 2\sigma(2x) - 1. \tag{5}$$

3.6. Optimizer. The optimizers serve to reduce the error function of the model. The internal parameters that can be learned are bias values and neural network weights used to calculate the output values. Optimizers have an indispensable role in decreasing losses experienced by the network formation process and in the neural network model during training.

The NN model is trained by updating the parameters of all the network layers iteratively, and the optimizer plays an important role here. In this respect, the different optimization techniques are explained below:

SGD is considered as a popular approach only because of its simplicity and ability to adapt linear algorithms and repressors. From multiple literature studies, we have seen that this method is much faster and can be learned online. Therefore, the SGD technique successfully conducts a

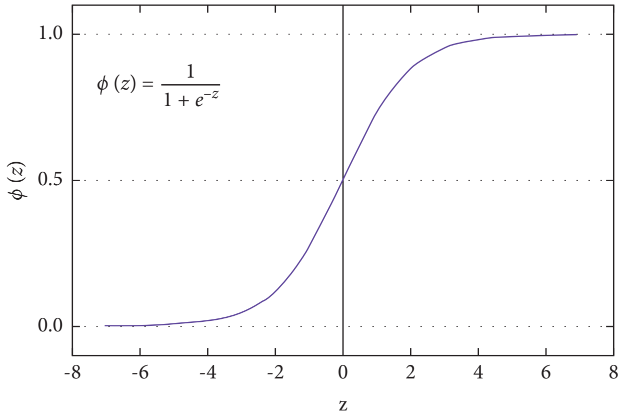

Figure 3: Sigmoid function.

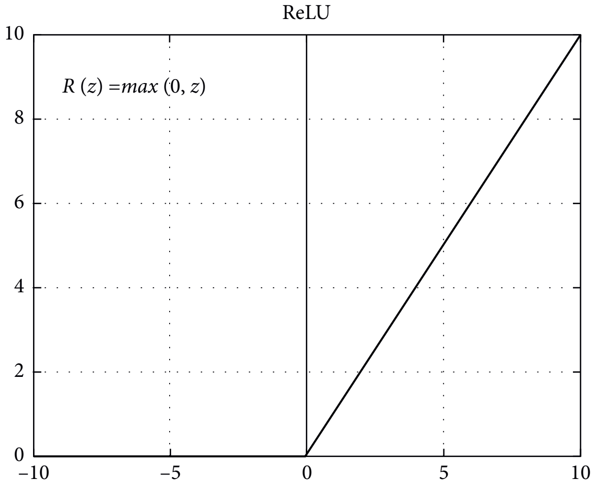

Figure 4: Rectifier function.

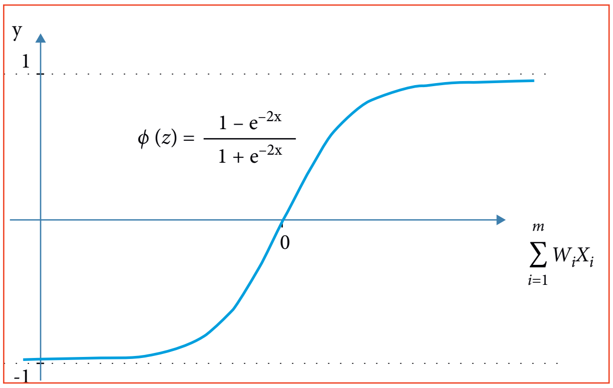

Figure 5: Tanh function.

parameter change in response for each training example considered, as explained in [41].

A description of the estimation of adaptive moments can be found in Zhang's work [42] (Adam). It is a first-order optimization approach that uses gradients, stochastic objective functions, and adaptive lower-order moment estimations.

The adaptive learning rate of the parameters is calculated using Adam. The prior gradients' exponential average is kept lowering in this approach.

By raising the denominator, the adaptive gradient algorithm (AdaGrad) [43] allows for a drop-in learning rate. We can also say it alternatively that the gradient-based optimization technique has been used which also adapts the learning rate that would then predict the parameters; it can surely provide us smaller updates been made to the data (low learning rates) for all those parameters that would consider to be relevant to it. This is because the AdaGrad technique employs very frequent features in the model and very greater updates or (high learning rates) considers for a parameter that would be relevant to the rare features, then it would surely be working fine with spare data in the network.

The authors in [44] had given us a new AdaGrad feature in the published study that simply tries to evenly reduce aggressive and monotonous learning, respectively. Thus, if we collect all the prior squared gradient information instead, the authors in [44] had suggested limiting the size of the window that accumulated the past gradient algorithms to a very specific size ratio to be used.

Also, when we talk about the root mean square propagation (RMSProp) technique, it is an AdaGrad modification in advance that would then regulate the learning rate's t a fast decrease level. It has widely been considered like the Adadelta technique. Although, the Adadelta technique definitely uses the RMSProp technique of parameter modifications which was performed in the rule of numerator's update. The AdaMax technique is a version of Adam based on the infinity standard [42], respectively.

Now, we can surely say that the SGD, the AdaGrad algorithm, and the Adadelta have all been used with the NN in and out in any case. Similarly, the Adam technique, the Adamax technique, and the RMSProp technique have also been used with NN as well.

4. Results

4.1. Impact of Activation Function. In the first experiment, we try different combinations of the activation function for the NN. This is performed during the training and testing phase. The goal is to investigate the impact of the configuration of alternative activation function on the extreme performance of the NN's model, which is the study's first question. Table 3 shows the results of the findings.

In Table 3, the churn prediction accuracy for the telecommunication industry varies depending on the configuration of the activation functions chosen.

4.2. Impact of Batch Size. It should be remembered that the sizes of the batch (number of lines) define the number of samples which propagate in the network during an epoch training phase.

In the study, the goal/target of the second experiment was considered to investigate the impact and influence of numerous different batch sizes in which we had the training of the NN. This was also been asked as the second question of

Table 3: Activation and output function results.

Hidden layer	Output layer	Accuracy (%)
Sigmoid	Sigmoid	79.8
Tanh	Tanh	79.85
Sigmoid	Tanh	83.6
Tanh	Sigmoid	85.5
Rectifier	Sigmoid	86.8
Rectifier	Tanh	84.8
Rectifier	Rectifier	84.75
Tanh	Rectifier	80.5
Sigmoid	Rectifier	82.9

our study initially. We gradually raised the batch size numbers to test how the NN model worked, and the results are displayed in Table 4.

Table 4: Impact batch sizes on NN performance.

Batch size	Accuracy (%)
3	85.2
7	85.75
10	84.9
12	84.2
15	84.3
20	84.1
25	84.2
30	84
35	84.35
40	84.2
50	84.25
70	84.45
90	83.8
110	84.25
140	85.15
180	83.85
230	83.2

4.3. Impact of Optimizers. In the third experiment, we focus on the third question of the study, i.e., the impact of optimizers on the performance of the NN model.

We divided the dataset into 10 parts during the training phase. To use the cross-validation of k-fold, the model is trained on nine flaps and tested in tenth.

Here, the used k-fold cross-validation technique would allow the considered model hence to be trained with much more perfect precise manner than merely testing on the test set, respectively. Table 5 shows the outcomes of training and testing the model at the same time.

When we consider some of the optimizers, then we are talking about some of the popular optimizers that may surely include firstly the stochastic gradient descent (SGD) algorithm, secondly the adaptive gradient algorithms (AdaGrad), and thirdly derivatives. We can also consider some of the other algorithms such as the Adadelta algorithm, root mean square propagation (RMSProp) algorithm, the Adam algorithm, and finally, the AdaMax algorithm.

Table 5: Impact of optimizer on NN performance.

Optimizer	Accuracy (%)
SGD	83.1
AdaGrad	83.75
Adadelta	85.65
AdaMax	84.5
Adam	84.5
RMSProp	86.45

4.4. Analysis and Discussion. According to the findings of the first experiment, the NN with Relu function in hidden layers and sigmoid function in the output layer gives the best results (86.9 percent). With this setting, the NN model can segregate data better.

We may infer that utilising the Relu function as a hidden layer activation function allows the model to categorise even negative values, which is the study's initial aim. It also showed that the model did not run out of values because they were outside of the analytical range. As a result, when the Relu function is applied in the hidden layers, the NN performs best.

On the one hand, the findings of the second experiment showed that the batch size has little influence on the NN's performance, especially as the batch size lowers. For batch sizes of 3 to 40, the NN's performance stabilised at an average of 84.52. The NN's performance, on the other hand, decreases as the batch size is greater. Moreover, Kandel and Castelli [45] claim that as the batch size approaches the test set, performance suffers noticeably due to the time constraints of processing each row individually.

Finally, the analysis of the third experiment revealed that when RMSProp is chosen as the training algorithm, we achieve better results (85.45 percent), while we achieve the lowest results (84.1 percent) when SGD is used as the training algorithm.

Importantly, the new research makes a theoretical and practical contribution. First, there are a few publications in the telecom sector that look at the effect of hyperparameter setup on the deep neural network churn prediction performance. Here, it would not be in vain to say that as a result, our study would surely contribute to the theoretical data mainly by setting the crucial foundation work for a better understanding of the influence work of hyperparameter configurations. It would also lay a foundation on their different values when we make a deep neural network architecture model that would then predict the customer's churn in the famous telecommunication sector that operates in our daily life. Its work is to facilitate us in comprehending the consequences of different widely used activation functions in the model when it is used for customer's churn prediction using the NN, while in contrast to the previous research. When NN is used to anticipate customer attrition, we investigated how different batch sizes and types of optimizers impact its performance.

On a practical level, this research lays the groundwork for generating meaningful heuristic knowledge that can help researchers apply ML techniques in the telecommunication industry to predict customer churn. This research contributes to the improvement of hyperparameter tuning efficiency when training NNs for churn modelling.

Despite the hopeful results, there are certain limitations to the study. To begin with, the dataset was a fake one obtained from a public repository. Because the data were acquired over a short period from a single telecommunication operator, this dataset could not be scaled up to additional operators. As a result, results should only be generalised with extreme caution. However, this constraint provides an opportunity to do additional research to evaluate the reproducibility of the results, with larger datasets collected over time from several telecommunication providers to generalise the results across the telecommunication sector.

The dataset used is also imbalanced. There are 483 churners and 2,850 non-churners in the sample. The predicted accuracy of the ML classifiers may surely be harmed by utilising the very well-known cross-validation method in the neural network therefore to obtain a valid representation of each category, respectively. Aside from these shortcomings, it is worth noting that the study met all three of its key objectives outlined in Section 1.

5. Conclusion

The effects of various hyperparameter configurations on the performance of a NN were investigated in this study. Three experiments were designed to confirm this. First, we investigate the impact of various monotonic activation function combinations is investigated in the hidden and output layers. Second, we investigate how different batch sizes affect a NN's performance. Third, we investigate the NN model's performance using various optimizers.

In this article, the data which we had mentioned, in which the first experiment demonstrated that by utilising a Relu function in the hidden layer of a neural network of the designed model and where a widely used sigmoid functioned applied on the output layer on the network, the NN churn model has surely given us the sign of working very well in the telecommunication industry though. Now, we had also noticed that the second experimental data had been demonstrated that the considered batch size in the model has a very significant successful influence on the NN's performance, as discussed in this article. While it is worth saying that the performance was dropping down as the batch size was trying to approach the size of the data in the test set, respectively, even though if we say that its architecture can surely prevent it from being actively trained while using RMSProp. Similarly, the last and third experimental data had demonstrated that the NN used in the model had performed ultimately best if the Adam algorithm was to be chosen as the training method in the research.

Furthermore, the current study has several significances been shown, which includes (i) the possibility of deeply expanding the customer's churn study to a level where we can add deep learning to it, that would then predict not only the customer's churn but would surely predict the loyalty as well in the explained model. Here, it would also be worth considering that this framework may definitely produce three different divided levels of loyalty categories such as not loyal, loyal, or very loyal. (ii) We can also say that the development of a used hybrid architecture which might surely be as based on the human's configurations though and would autonomously pick the perfect and very best hyperparameters to ultimately train, test, and improve its performance to the model. (iii) Finally, last but not least to say that we would also be doing the investigation of the impact of sinusoidal activation functions. Here, we are talking about the sine and spline that would have an effect on various types of activation functions in this situation to be considered as well.

Data Availability

The dataset used to support the findings of this study are available from the author upon request.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] M. Chandar and P. A. L. Krishna, "Modeling churn behavior of bank customers using predictive data mining techniques," *Proceedings of International Conference on Soft Computing Techniques and Engineering Application*, vol. 86, pp. 24–26, 2006.

[2] J. Ahn, J. Hwang, D. Kim, H. Choi, and S. Kang, "A survey on churn analysis in various business domains," *IEEE Access*, vol. 8, Article ID 220816, 2020.

[3] M. Almana and M. S. A. R. Alzahrani, "A Survey on data mining techniques in customer churn analysis for telecom industry," *International Journal of Engineering Research in Africa*, vol. 4, no. 5, pp. 165–171, 2014.

[4] A. Ahmed and D. M. Linen, "A review and analysis of churn prediction methods for customer retention in telecom industries," in *Proceedings of the 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1–7, Coimbatore, India, January 2017.

[5] S. A. Qureshii, A. S. Rehman, A. M. Qamar, A. Kamal, and A. Rehman, "Telecommunication Subscribers' Churn Prediction Model Using Machine Learning," in *Proceedings of the Eighth International Conference on Digital Information Management*, pp. 131–136, Islamabad, Pakistan, September 2013.

[6] V. Umayaparvathi and K. Iyakutti, "A survey on customer churn prediction in telecom industry: datasets, methods and metric," *International Research Journal of Engineering and Technology*, vol. 3, no. 4, pp. 65–70, 2016.

[7] T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," *Simulation Modelling Practice and Theory*, vol. 55, pp. 1–9, 2015.

[8] P. K. Dalvi, S. K. Khandge, A. Deomore, A. Bankar, and V. A. Kanade, "Analysis of customer churn prediction in telecom industry using decision trees and logistic regression," in *Proceedings of the 2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, pp. 1–4, Indore, India, March 2016.

[9] S. Khodabandehlou and M. Zivari Rahman, "Comparison of supervised machine learning techniques for customer churn prediction based on analysis of customer behavior," *Journal of Systems and Information Technology*, vol. 19, no. 1/2, pp. 65–93, 2017.

[10] A. Hemlat, A. Khunteta, and S. Srivastava, "Churn prediction in telecommunication using logistic regression and logit boost," *Procedia Computer Science*, vol. 167, pp. 101–112, 2020.

[11] J. Vijaya and E. Sivasankar, "Improved churn prediction based on supervised and unsupervised hybrid data mining system," *Information and Communication Technology for Sustainable Development*, Springer, vol. 9, Singapore, , 2018.

[12] S. Maldonado, Á. Flores, T. Verbraken, B. Baesens, and R. Weber, "Profit-Based feature selection using support vector machines, general framework and an application for customer retention," *Applied Soft Computing*, vol. 35, pp. 740–748, 2015.

[13] I. Brandusoiu and G. Toderean, "Churn prediction in the telecommunications sector using support vector machines," *Annals of the Oradea University: Fascicle Management and Technological Engineering*, vol. 1, pp. 19–22, 2013.

[14] S. Induja and M. Eswaramurthy, "Customers churn prediction and attribute selection in telecom industry using kernelized extreme learning machine and bat algorithms," *International Journal of Science and Research*, vol. 5, pp. 258–265, 2016.

[15] A. Lemmens and C. Croux, "Bagging and boosting classification trees to predict churn," *Journal of Marketing Research*, vol. 43, no. 2, pp. 276–286, 2006.

[16] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, pp. 1399–1404, 1999.

[17] K. Coussement and K. W. De Bock, "Customer churn prediction in the online gambling industry: the bene_cial effect of ensemble learning," *Journal of Business Research*, vol. 66, no. 9, pp. 1629–1636, 2013.

[18] A. Mishra and U. S. Reddy, "A novel approach for churn prediction using deep learning," in *Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1–4, Coimbatore, India, December 2017.

[19] M. Tim, G. Jeremy, and F. Art, "Data mining static code attributes to learn defect predictors," *IEEE Transactions on Software Engineering*, vol. 33, no. 9, pp. 635-636, 2007.

[20] K. Dahiya and S. Bhatia, "Customer churn analysis in telecom industry," in *Proceedings of the 4th International Conference on Reliability, Infocom Technologies and Optimization (ESCRITO) (Trends Future Directions)*, pp. 1–6, Noida, India, September 2015.

[21] M. Owczarczuk, "Churn models for prepaid customers in the cellular telecommunication industry using large datasets," *Expert Systems with Applications*, vol. 37, pp. 4710–4712, 2010.

[22] A. Tamaddoni, S. Stakhovych, and M. Ewing, "Comparing churn prediction techniques and assessing their performance: a contingent perspective," *Journal of Service Research*, vol. 19, no. 2, pp. 123–141, 2016.

[23] K. Gajowniczek, T. Zabkowski, and A. Orlowski, "Comparison of Decision Trees with Rényi and Tsallis Entropy Applied for Imbalanced Churn Dataset," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, pp. 39–44, Lodz, Poland, September 2015.

[24] K. Ebrah and S. Elnasir, "Churn prediction using machine learning and recommendations plans for telecoms," *Journal of Computer and Communications*, vol. 7, pp. 33–53, 2019.

[25] Y. Zhang, R. Liang, Y. Li, Y. Zheng, and M. Berry, "Behavior-based telecommunication churn prediction with neural network approach," in *Proceedings of the 2011 International Symposium on Computer Science and Society*, pp. 307–310, IEEE, Kota Kinabalu, Malaysia, July 2011.

[26] T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Proceedings of the International Workshop on Multiple Classifier Systems*, pp. 1–15, Springer, London, UK, 2015.

[27] A. Baumann, S. Lessmann, K. Coussement, and K. W. De Bock, "Maximize what matters: predicting customer churn with decision-centric ensemble selection," in *Proceedings of the 23rd European Conference on Information Systems (ECIS)*, AIS, Munster, Germany, May 2015.

[28] J. Xiao, Y. Xiao, A. Huang, D. Liu, and S. Wang, "Feature-selection-based dynamic transfer ensemble model for customer churn prediction," *Knowledge and Information Systems*, vol. 43, pp. 29–51, 2015.

[29] A. Afan and F. Yangyu, "Unsupervised feature learning and automatic modulation classification using deep learning model," *Elsevier Physical Communication*, vol. 25, pp. 75–84, 2017.

[30] A. Sharma and K. Panigrahi, "A neural network based approach for predicting customer churn in cellular network services," *International Journal of Computer Applications*, vol. 27, pp. 26–31, 2011.

[31] M. Azeem, M. Usman, and A. C. M. Fong, "A churn prediction model for prepaid customers in telecom using fuzzy classifiers," *Telecommunication Systems*, vol. 66, pp. 603–614, 2017.

[32] N. Alboukaey, A. Joukhadar, and N. Ghneim, "Dynamic behavior-based churn prediction in mobile telecom," *Expert Systems with Applications*, vol. 162, 2020.

[33] M. Ahmed, H. Afzal, A. Majeed, and B. Khan, "A survey of evolution in predictive models and impacting factors in customer churn," *Advances in Data Science and Adaptive Analysis*, vol. 9, no. 3, Article ID 1750007, 2017.

[34] A. Ahmed and D. M. Linen, "A review and analysis of churn prediction methods for customer retention in telecom industries," in *Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1–7, Coimbatore, India, January 2017.

[35] Crowd Analytix, "Dataset," 2012, https://www.crowdanalytix.com/contests/why-customer-churn.

[36] C. Aggarwal, *Outlier Analysis," in Data Mining*, Springer, Berlin, Germany, 2015.

[37] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–445, 2015.

[39] E. Domingos, B. Ojeme, and O. Daramola, "Experimental analysis of hyperparameters for deep learning-based churn

prediction in the banking sector," *Computation*, vol. 9, no. 34, 2021.

[40] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Neural Networks*, vol. 138, pp. 14–32, 2021.

[41] M. N. Halgamuge, E. Daminda, and A. Nirmalathas, "Best optimizer selection for predicting bushfire occurrences using deep learning," *Natural Hazards*, vol. 103, pp. 845–860, 2020.

[42] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," in *Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pp. 1–7, Banff, AB, Canada, June 2018.

[43] N. Zhang, D. Lei, and J. F. Zhao, "An Improved Adagrad Gradient Descent Optimization Algorithm," in *Proceedings of the 2018 Chinese Automation Congress (CAC)*, pp. 2359–2362, Xi'an, China, December 2018.

[44] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease classification: a comparative evaluation of convolutional neural networks and deep learning optimizers," *Plants*, vol. 9, 2020.

[45] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, 2020.