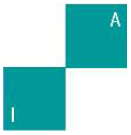
Inteligencia Artificial 26(71), 46-58
doi: 10.4114/intartif.vol26iss71pp46-58



INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Prediction of Research Project Execution using Data Augmentation and Deep Learning

Anibal Flores[1,A], Hugo Tito-Chura[1], Lissethe Zea-Rospigliosi[2]
[1]GIIA – Research Group in Artificial Intelligence, Universidad Nacional de Moquegua, Moquegua, Peru
[A] afloresg@unam.edu.pe
[2]Doctorate in Education, Universidad Nacional Jorge Basadre Grohmann, Tacna, Peru

Abstract This paper presents the results of seven deep learning models for prediction of research project execution in graduates from a public university in Peru. The deep learning models implemented are non-hybrid: Deep Neural Networks (DNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Networks (CNN) and, hybrid: CNN+GRU, CNN+ LSTM and LSTM+GRU. Since most of the dataset prediction features are of the nominal type (true or false), this paper proposes a simple novel data augmentation technique for this type of features. Taking as inspiration the input data type of a neural network, the proposal data augmentation technique considers nominal features as numeric, and obtain random values close to them to generate synthetic records. The results show that most of deep learning models with data augmentation significantly outperform models with just class balancing in terms of accuracy, precision, f1-score and specificity, being the main improvements of 17.39%, 80.00%, 25.00% and 20.00% respectively.

Keywords: Research project execution, data augmentation, deep learning, nominal features.

1 Introduction

In Peru, the number of university graduates has a positive trend that goes from 669,000 in 2012 to 833,000 in 2018 [1], but despite this, the proportion of people with completed university education does not exceed 13.0%. The regions with the highest proportion of graduates with respect to their population are Arequipa, Moquegua, Tacna and Puno with approximately 17.0%.

Although the number of university graduates has a positive trend, the proportion of those who manage to obtain their professional degree is worrying since it is less than 50%. This is due to multiple factors such as too complicated qualification requirements, lack of financial resources, lack of time, training, among others. Obtaining the professional degree is very important for graduates since this allows them to have greater opportunities to enter the workplace, where the professional degree is often a requirement [2].

The Universidad Nacional de Moquegua (UNAM) is no stranger to this problem and the ratio of graduated graduates reflects what the national reality indicates, and in the Systems Engineering and Informatics career this is more worrying since the ratio of graduated graduates is less than 20%.

Thus, considering that one of the modalities to obtain the professional degree is the realization or execution of a research project, in this study the implementation of different deep learning models using data augmentation is proposed to predict if a graduate will execute his research project. This is very important, since it will allow the timely identification of graduates who will not be able to execute their research project for the corresponding decision-making by the university authorities and, in this way improve the ratio of graduated graduates.

For the implementation of deep learning models, academic and socioeconomic data collected from 2017 to 2021 from the System Engineering and Informatics career of UNAM will be used. Being almost all the features of the nominal type containing true false values, except the enrol number feature which is numeric. Being a public

ISSN: 1137-3601 (print), 1988-3064 (on-line)

university, many students and graduates have limited financial resources and, in many cases do not have computer equipment and internet service. In the academic field, features related to their academic performance in the Thesis Seminar course are considered, such as has topic, changed the research topic, passed the course, study modality, enroll number among others. The Thesis Seminar corresponds to the last semester (X) of the study plan and its main objective is to ensure that the students complete the course with the research project formulated.

For experimentation seven deep learning models have been implemented, four non-hybrids and three hybrids. The non-hybrids are Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Also, the three hybrids are CNN+LSTM, CNN+GRU and LSTM+GRU.

One of the main difficulties for carrying out this work was the number of available records, which in total correspond to 77 students. Of which 54 (70%) are used for the training phase. And, when working with deep learning models, one of the main requirements to ensure good results is to have enough data for training in order to do not have underfitting or overfitting problems[3]. In this work, to avoid the problems described and improve the performance of deep learning models, a new data augmentation technique is proposed for classification models with nominal prediction features.

Data augmentation emerged in the field of computer vision [4] and initially consisted of performing simple operations on the original images such as rotation, scaling, cropping, mirroring to generate synthetic images [5], this allowed enriching the dataset and achieving better model training. Subsequently, it was taken to other fields such as Natural Language Processing (NLP) [6],[7], time series [8],[9] and structured data [10].

The proposal data augmentation technique considers each prediction feature as numeric instead of nominal, just as a neural network considers each prediction feature, with the aim of generating approximate synthetic values of 1 or 0, for which it uses two parameters: a) *snumber*, which corresponds to the number of synthetic records to be generated from a non-synthetic one, and a *max_distance*, which is a value between 0 and 0.5, it is used to generate random values close to 1 or 0 depending on the non-synthetic value, the corresponding algorithm is detailed in section III.

Regarding the limitations of this study, it is limited to the estimation of whether the graduate will be able to approve and execute his research project, but not if he will be able to complete and approve the respective report of the same, because to date the number of completed and approved research projects is only 15, and this is too small to implement machine or deep learning models. Therefore, 13 research projects in execution which are not completed yet were also considered. Thus, the target feature of the models for the class “1” has 28 records and for the class “0” has 49. Likewise, in this work, it is not intended to compare the proposal data augmentation technique with others of the state of the art.

Regarding the content organization of this work. Section 2 describes the related work that constitutes the starting point of the study. In section 3, the methodology addressed for the implementation of deep learning models is described. Section 4 describes the results achieved by the implemented models. Finally, in section 5, the conclusion reached and the future work that can be carried out based on the results of this work are shown.

2 Related Work

Most of the related work to this topic are focused on determining the factors that influence research project completion and non-completion, thus the works [11], [12] and [13] identified different factors such as motivation, ability to write scientific papers, guidance quality, commitment, diligence, communication skills, research experience among others.

Regarding the prediction of research project execution with machine learning and deep learning techniques, no work has been identified. The most similar work is related to thesis completion and non-completion through machine learning techniques [14]. In this work, the authors carry out a study to identify factors for master thesis completion and non-completion through learning analytics and machine learning techniques, for which 755 master thesis projects are used and several machine learning models are implemented, such as Naive Bayes, Logistic Regression, Deep Learning, Decision Tree, Random Forest and Gradient Boosted Trees. The results show that the best model is Gradient Boosted Trees with 72% accuracy.

In the other hand, regarding the deep learning techniques chosen for this work, all of them have been used in other works. Deep Neural Networks (DNNs) are commonly used for classification tasks with structured data in many works such as [15], [16] and [17], this is probably the simplest and most widely used deep learning technique for structured data classification. Recurrent Neural Networks such as LSTM and GRU are commonly used for sequential data such as Time Series and NLP tasks, but can also be used for classification as was done in [18], [19] and others, in these works RNNs presented better results than machine techniques such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Trees, Naïve Bayes, Logistic Regression and others. Finally, CNNs

are commonly used in Computer Vision with 2D and 3D layers, but they can also work with 1D layers for structured data. In works such as [20] and [21] CNNs are implemented with accuracies above 90%.

Regarding hybrid models, CNN+LSTM model has been used in different works for classification and regression tasks, for example [22], [23] and [24] implemented this model obtaining better results than standalone models. Regarding CNN+GRU, in similar way to CNN+LSTM, it has been used in several works for regression and classification such as [25], [26] and [27]. Finally, regarding LSTM+GRU, both standalone models are recurrent neural networks with similar architecture being GRU simpler than LSTM, they have been used for regression and classification in [28], [29] and [30], showing the superiority of hybrid model regarding standalone models.

3 Methodology

Graphically Figure 1 summarizes the methodology followed in this work for the implementation of the different deep learning models for prediction of research project execution.

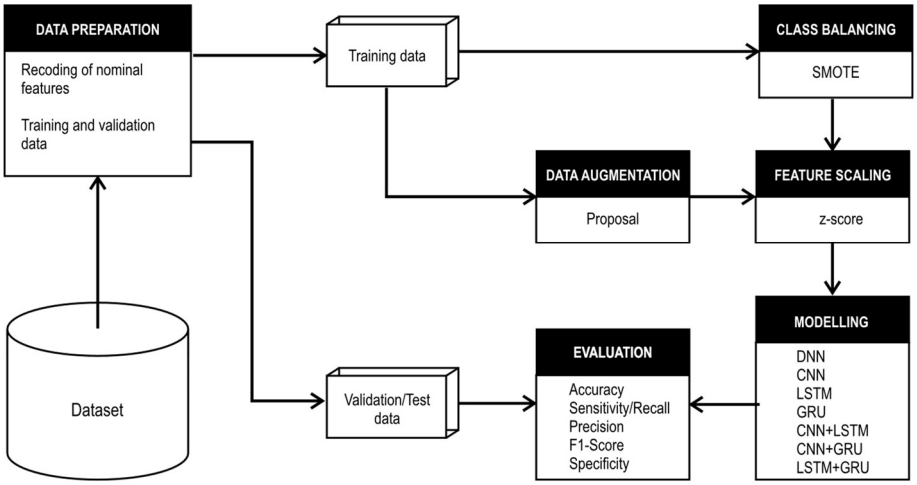


Figure 1. Methodology

3.1 Data Preparation

The dataset for this study is made up of data from 77 graduates of the System Engineering and Informatics career at UNAM enrolled in the Thesis Seminar course during the years 2017 and 2021 where most of them are male (57) and female (20). Also, it is important to indicate that the graduates who were students between the years 2017 and 2019 took the course face-to-face, while between the years 2020 and 2021, the course was developed virtually due to the COVID-19 pandemic.

The first data preparation activity was to recode the nominal variables. Thus, Gender was recoded from "F" to 0 and "M" to 1. Origin had more than five (5) different values, so it was decided to consider just two (2) ones; all graduates from Ilo city (0) and graduates from other cities or regions (1), because most of graduates were from Ilo city. Other variables that were recoded from "Yes" to 1 and "Not" to 0 were Income, Cell, PC, Laptop, Internet, Tablet, Breakfast, Insurance, Sport and Failed_courses. Modality was recoded from "face-to-face" to 0 and "virtual" to 1. Table 1 shows all the prediction and target features of the dataset.

Table 1: Features

N°	Feature	Description	Type
1	Gender	It corresponds to the student gender: male (1) or female (0)	Nominal
2	Origin	It's the place where the student comes from: Ilo (0) or other (1).	Nominal
3	Income	The student has some income: Yes (1) and Not (0)	Nominal
4	Cell	The student has cell phone: Yes (1) and Not (0)	Nominal
5	PC	The student has personal computer: Yes (1) and Not (0)	Nominal
6	Laptop	The student has laptop: Yes (1) and Not (0)	Nominal
7	Internet	The student has internet service: Yes (1) and Not (0)	Nominal

8	Tablet	The student has tablet: Yes (1) and Not (0)	Nominal
9	Breakfast	The student has breakfast before going to class: Yes (1) and Not (0)	Nominal
10	Insurance	The student has some kind of insurance: Yes (1) and Not (0)	Nominal
11	Sport	The student practice some sport: Yes (1) and Not (0).	Nominal
12	Failed_courses	The student failed courses in previous semesters: Yes (1) and Not (0)	Nominal
13	Previous_Semester	The student had successful academic performance in the previous semester: Yes (1) and Not (0)	Nominal
14	Enroll_Number	The times the student was enrolled in the Thesis Seminar course.	Numeric
15	Modality	The modality the student was enrolled the Thesis Seminar course. Face-to-face (0) and virtual (1)	Nominal
16	Passed	The student passed de Thesis Seminar course. Yes (1) and Not (0)	Nominal
17	Has_Topic	Has the student research topic? Yes (1) and Not (0)	Nominal
18	Changed_Topic	The topic was changed during the course development. Yes (1) and Not (0)	Nominal
19	Target	It corresponds to the target feature for the classification models. (1) The student is executing or has finished the research project. (0) the student is not executing a research project.	Nominal

Once the features of the dataset have been recoded, the training and test/validation data are generated. For this, 70% (54 records) are randomly considered for model training and, 30% for test/validation (23 records).

3.2 Data Augmentation

The proposal data augmentation technique considers each prediction feature as numeric instead of nominal. The aim is to generate approximate synthetic values of 1 or 0 as appropriate, for which it uses two parameters:

a) *snumber*, which corresponds to the number of synthetic records to be generated from a non-synthetic one, in this work it is experimented with *snumber*=20. That is, for each non-synthetic record, 20 synthetic records are generated.
b) *max_distance*, which is a value between 0 and 0.5, it is used to generate random values close to 1 or 0 depending on the non-synthetic value. In this work, *max_distance* is set to 0.25.

Table 2 shows the algorithm implemented in Javascript language. The algorithm receives as input the dataset *d* in string format and the respective values for its *snumber* and *max_distance* parameters. Each record is split by "\n" and each feature is split by ",".

For each non-synthetic record, *snumber* synthetic records are generated considering a deviation according to *max_distance*. An example of the result of the proposal data augmentation technique can be seen in Figure 2, where a non-synthetic record with five synthetic records generated with *max_distance*=0.25 are shown. The non-synthetic record corresponds to the dataset of this work.

In case the dataset has features with other values than 0 and 1, the proposal data augmentation technique will apply the same logic for these features. This is the case of enroll_number feature which is numeric, but the technique with *max_distance*=0.25 works fine for it. It's important to highlight that for numeric features with smaller values *max_distance* should be analysed in order to generate right and useful values.

Table 2: Javascript code for proposal data augmentation technique

```
1  function augment(d, snumber, max_distance)
2  {
3      train=[];
4      dd=d.split("\n");
5      nd=dd.length;
6      for(i=0;i<nd;i++)
7      {
8          e=d[i].split(",");
9          ne=e.length-1;
10         train.push(e);
11         k=1;
```

```
12              rclass=e[ne];
13              while(k<=snumber)
14              {    aug=[];
15                   for(j=0;j<ne;j++)
16                   {    v=parseInt(e[j]);
17                        ini=(v-max_distance);
18                        end=(v+max_distance);
19                        if(v>0)
20                             sv=Math.random() * (v - ini) + ini;
21                        else
22                             sv=Math.random() * (end - v) + v;
23                        aug.push(sv);
24                   }
25                   aug.push(rclass)
26                   train.push(aug);
27                   k++;
28              }
29         }
30         return(train)
31   }
```

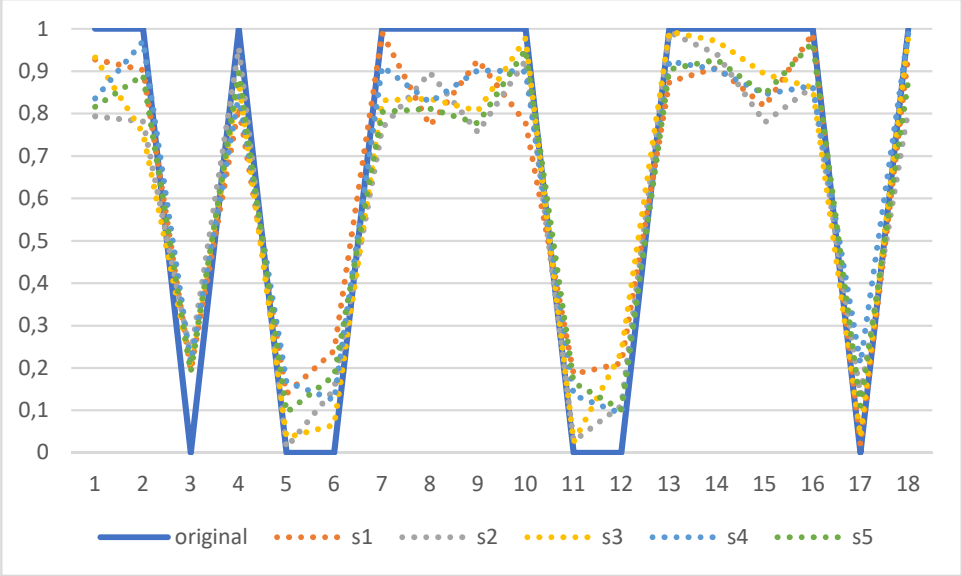


Figure 2. Five synthetic records generated from one non-synthetic record

After the application of the proposal data augmentation technique in training data, with *snumber*=20, 1080 synthetic records were generated and, being 54 non-synthetic records, in total there are 1134 records for training stage.

3.3 Class Balancing

Of the 54 data records for the model training phase, 30 correspond to the class of graduates who did not manage to execute their research project and 24 to those who did manage to execute their research project. Thus, in order to avoid the overfitting problem [3], the respective class balancing was carried out by applying the SMOTE [31] technique (Synthetic Minority Oversampling Technique) which is very used in several classification works. After this process both classes have 30 records.

It is important to highlight that training data with data augmentation was not balanced. SMOTE was just applied to training with no data augmentation.

3.4 Feature Selection

In this stage, correlation matrix was implemented, and according to Figure 3, correlations between prediction features and target feature can be seen in the last column and the last row. It can be appreciated that the correlations range between -0.4717 and 0.4682. So, for this study, it was decided to work with all prediction features for modelling phase.

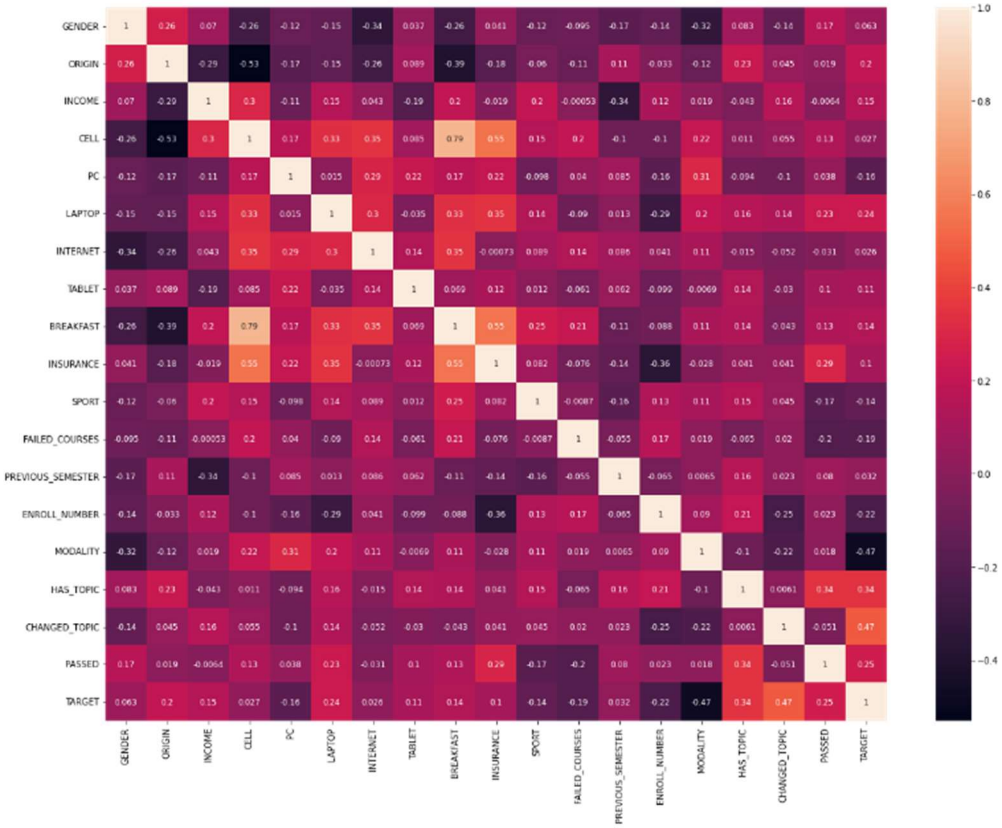


Figure 3. Correlation Matrix

3.5 Feature Scaling

When deep learning models are implemented, feature scaling is very important, so in this work, z-score normalization was considered, for this task equation (1) is used. According [32] feature scaling helps deep learning models to converge faster.

$$x' = \frac{x - \bar{x}}{\sigma} \tag{1}$$

Where:

x : original feature vector

$\bar{x}$: mean of feature vector

σ : standard deviation of feature vector

3.6 Modelling

For the implementation of deep learning models tensorflow 2.4.0 and keras 2.8.0 with Python and Google Colab were used. Table 3 shows the hyperparameters for all deep learning models implemented in this work. The last layer with 1 unit in all models is Dense, because prediction of research project execution is a binary classification problem.

Table 3: Hyperparameters for deep learning models

Model	Hyperparameters
DNN	Layers: 4, hidden units [23,23,23,1], drop_rate: 0.1, learning_rate: 0.001, epochs:100
LSTM	Layers: 3, hidden units [23,23,1], drop_rate: 0.1, learning_rate: 0.001, epochs:100
GRU	Layers: 3, hidden units [23,23,1], drop_rate: 0.1, learning_rate: 0.001, epochs:100
CNN	Layers: 4, hidden units [50,100,25,1], drop_rate: 0.1, learning_rate: 0.001, epochs:100
LSTM+GRU	Layers: 3, hidden units [23,23,1], drop_rate: 0.1, learning_rate: 0.001, epochs:100
CNN+LSTM*	Layers: 5, hidden units [50,100,23,23,1], drop_rate: 0.1, learning_rate: 0.001, epochs:100
CNN_GRU**	Layers: 5, hidden units [50,100,23,23,1], drop_rate: 0.1, learning_rate: 0.001, epochs:100

* The two first layers are Conv1D, the next two are LSTM.
** The two first layers are Conv1D, the next two are GRU

Learning rate was chosen from different experiments, it was tested with 0.1, 0.01 and 0.001, and the best results were obtained with 0.001. Architectures were chosen from other classification works.

3.7 Evaluation

For the evaluation of models, metrics such as Accuracy, Sensitivity/Recall, Precision, F1-Score and Specificity were considered, which are estimated through equations (2), (3), (4), (5) and (6) respectively.

$$accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{2}$$

$$sensitivity/recall = \frac{(TP)}{(TP+FN)} \tag{3}$$

$$precision = \frac{(TP)}{(TP+FP)} \tag{4}$$

$$f1-score = \frac{(2*sensitivity*precision)}{(sensitivity+preci\quad)} \tag{5}$$

$$specificity = \frac{(TN)}{(FN+TN)} \tag{6}$$

Where:
TP : True Positives
TN : True Negatives
FP : False Positives
FN : False Negatives

4 Results and Discussion

In this section, the results achieved after the implementation of the different deep learning models are described.

4.1 Results

The results achieved by the implemented models in terms of accuracy, sensitivity, precision, f1-score and specificity are shown in Table 4.

Table 4: Metrics for deep learning models

Model	Accuracy		Sensitivity		Precision		F1-Score		Specificity	
	No DA	DA	No DA	DA	No DA	DA	No DA	DA	No DA	DA
DNN	0.7391	0.9130	0.3333	0.3333	0.2000	1.0000	0.2500	0.5000	0.8000	1.0000
LSTM	0.8696	0.9130	0.6667	0.6667	0.5000	0.6667	0.5714	0.6667	0.9000	0.9500

GRU	0.8696	0.8696	0.3333	0.3333	0.5000	0.5000	0.4000	0.4000	0.9500	0.9500
CNN	0.8696	0.8695	0.3333	0.3333	0.5000	0.5000	0.4000	0.4000	0.9500	0.9500
LSTM+GRU	0.8695	0.9130	0.3333	0.3333	0.5000	1.0000	0.4000	0.5000	0.9500	1.0000
CNN+LSTM	0.8261	0.9130	0.6667	0.3333	0.4000	1.0000	0.5000	0.5000	0.8500	1.0000
CNN+GRU	0.8261	0.8261	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.9000	0.9000

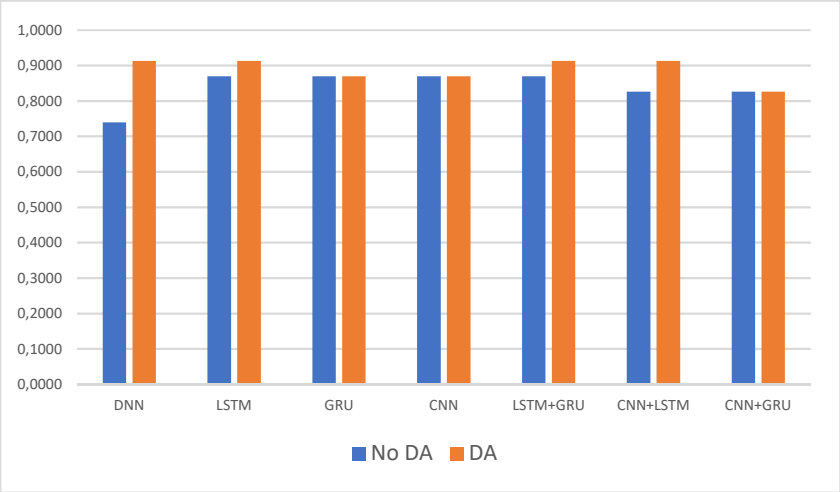


Figure 4. Accuracy

According to Table 4 and Figure 4, it can be seen that in terms of Accuracy, the proposal data augmentation technique allows improving four of the seven implemented models, the models that improve with data augmentation are DNN (17.39%), CNN+LSTM (8.7%), CNN+LSTM and LSTM, LSTM+GRU (4.35%). The other models present similar results in terms of accuracy.

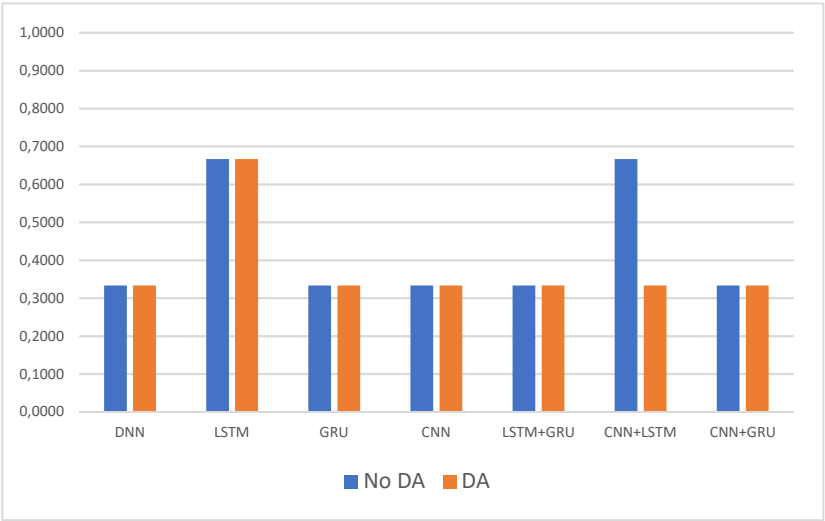


Figure 5. Sensitivity

In terms of Sensitivity, according to Table 4 and Figure 5, it can be seen that, the proposal data augmentation technique does not allow to improve any model with respect to the use of class balancing, both present the same sensitivity. Just one model with data augmentation (CNN+LSTM) decreases its sensitivity by 33.33%.

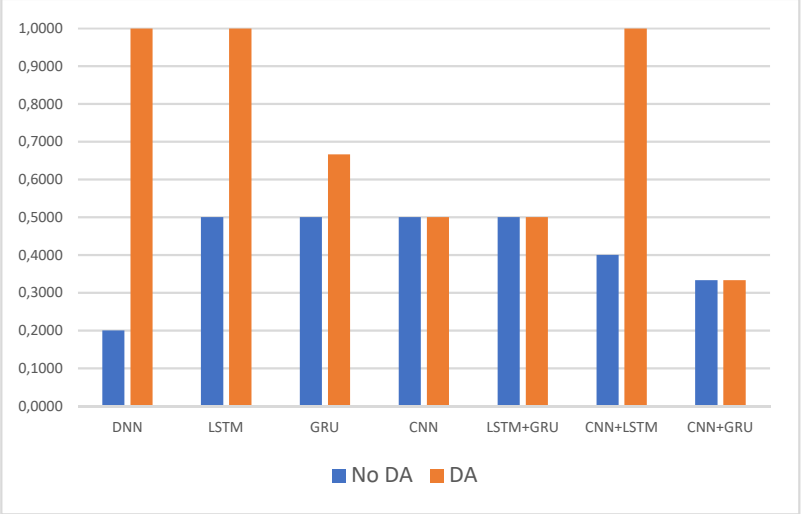


Figure 6. Precision

According to Table 4 and Figure 6, it can be seen that in terms of Precision, with the proposal data augmentation technique, four of the seven implemented models manage to improved (DNN, LSTM, GRU and CNN+LSTM), the other three (CNN, CNN+GRU and LSTM+GRU) keep the same precision.

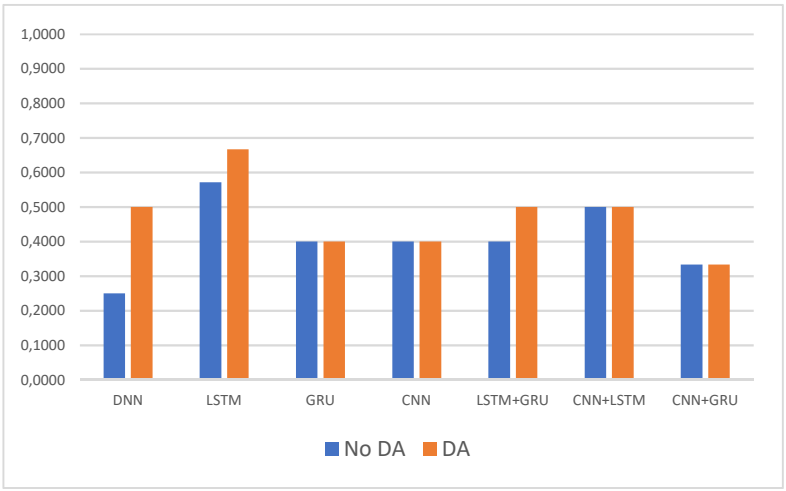


Figure 7. F1-Score

Regarding the F1-Score, according to Table 4 and Figure 7, in 3 of seven implemented models the proposal data augmentation technique allows to improve the F1-scores, the improvements range between 9.52% and 25.00%. The other four models keep the same results (GRU, CNN, CNN+LSTM and CNN+GRU)

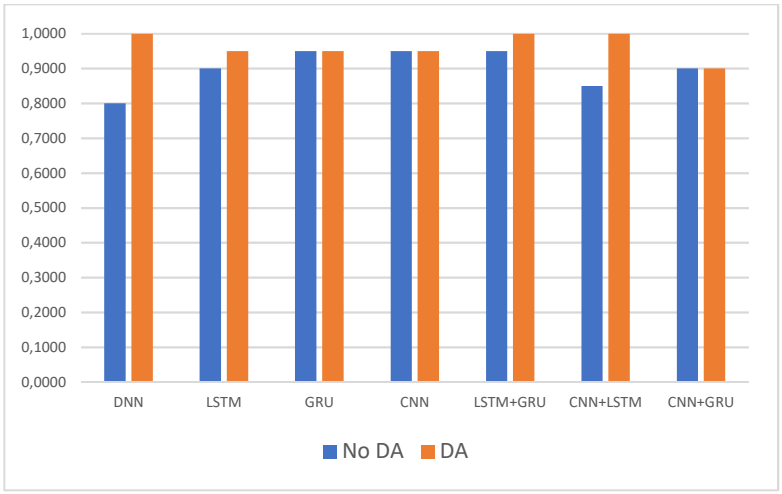


Figure 8. Specificity

Regarding the specificity, according to Table 4 and Figure 8, four of the seven implemented models (DNN, LSTM, LSTM+GRU and CNN+LSTM) manage to improve, the improvements range between 5% and 20%. The other three models keep the same specificity (GRU, CNN and CNN+GRU).

It is important to highlight that each model was trained for 100 epochs, and they were run at least a couple of times. For each epoch h5 files were generated with the respective model weights, from there the ones with the highest accuracy on validation data were chosen and saved.

Figure 9 shows the final evaluation on validation data of models which were trained with just class balancing (SMOTE) and, Figure 10 shows the final evaluation of models that used the proposal data augmentation technique. From Figure 9 and 10, Table 4 was generated.

These selected h5 files for all selected models and Coogle Colab files can be found at:
https://drive.google.com/drive/folders/1VRnhP52q85GNNlbgHUJ__dGNVMSwAIHP?usp=share_link

```
[36] from keras.models import load_model
     m1=load_model("dnn_7391.h5")
     m2=load_model("lstm_2_8696.h5")
     m3=load_model("gru_1_8696.h5")
     m4=load_model("cnn_1_8696.h5")
     m5=load_model("lstm_gru_1_8696.h5")
     m6=load_model("cnn_lstm_2_8261.h5")
     m7=load_model("cnn_gru_2_7826.h5")
```

```
[38] pred1=m1.evaluate(val_features, val_targets, verbose=1)
     pred2=m2.evaluate(val_features, val_targets, verbose=1)
     pred3=m3.evaluate(val_features, val_targets, verbose=1)
     pred4=m4.evaluate(val_features, val_targets, verbose=1)
     pred5=m5.evaluate(val_features, val_targets, verbose=1)
     pred6=m6.evaluate(val_features, val_targets, verbose=1)
     pred7=m7.evaluate(val_features, val_targets, verbose=1)

     1/1 [==============================] - 0s 46ms/step - loss: 2.1879 - fn: 2.0000 - fp: 4.0000 - tn: 16.0000 - tp: 1.0000 - precision: 0.2000 - recall: 0.3333 - acc: 0.7391
     1/1 [==============================] - 0s 39ms/step - loss: 0.6253 - fn: 1.0000 - fp: 2.0000 - tn: 18.0000 - tp: 2.0000 - precision: 0.5000 - recall: 0.6667 - acc: 0.8696
     1/1 [==============================] - 0s 39ms/step - loss: 0.3982 - fn: 2.0000 - fp: 1.0000 - tn: 19.0000 - tp: 1.0000 - precision: 0.5000 - recall: 0.3333 - acc: 0.8696
     1/1 [==============================] - 0s 37ms/step - loss: 0.4560 - fn: 2.0000 - fp: 1.0000 - tn: 19.0000 - tp: 1.0000 - precision: 0.5000 - recall: 0.3333 - acc: 0.8696
     1/1 [==============================] - 0s 39ms/step - loss: 0.5105 - fn: 2.0000 - fp: 2.0000 - tn: 18.0000 - tp: 1.0000 - precision: 0.3333 - recall: 0.3333 - acc: 0.8261
     1/1 [==============================] - 0s 40ms/step - loss: 0.5599 - fn: 1.0000 - fp: 3.0000 - tn: 17.0000 - tp: 2.0000 - precision: 0.4000 - recall: 0.6667 - acc: 0.8261
     1/1 [==============================] - 0s 42ms/step - loss: 0.6262 - fn: 2.0000 - fp: 2.0000 - tn: 18.0000 - tp: 1.0000 - precision: 0.3333 - recall: 0.3333 - acc: 0.8261
```

Figure 9. Evaluation of Deep Learning Models with No Data Augmentation

```
[34] from keras.models import load_model
     da1=load_model("da_dnn_1_9130.h5")
     da2=load_model("da_lstm_2_9130.h5")
     da3=load_model("da_gru_1_8695.h5")
     da4=load_model("da_cnn_1_8695.h5")
     da5=load_model("da_lstm_gru_1_9130.h5")
     da6=load_model("da_cnn_lstm_1_9130.h5")
     da7=load_model("da_cnn_gru_1_8261.h5")
```

```
[35] pred1=da1.evaluate(val_features, val_targets, verbose=1)
     pred2=da2.evaluate(val_features, val_targets, verbose=1)
     pred3=da3.evaluate(val_features, val_targets, verbose=1)
     pred4=da4.evaluate(val_features, val_targets, verbose=1)
     pred5=da5.evaluate(val_features, val_targets, verbose=1)
     pred6=da6.evaluate(val_features, val_targets, verbose=1)
     pred7=da7.evaluate(val_features, val_targets, verbose=1)

     1/1 [==============================] - 0s 386ms/step - loss: 0.5010 - fn: 2.0000 - fp: 0.0000e+00 - tn: 20.0000 - tp: 1.0000 - precision: 1.0000 - recall: 0.3333 - acc: 0.9130
     1/1 [==============================] - 1s 855ms/step - loss: 0.4551 - fn: 1.0000 - fp: 1.0000 - tn: 19.0000 - tp: 2.0000 - precision: 0.6667 - recall: 0.6667 - acc: 0.9130
     1/1 [==============================] - 1s 675ms/step - loss: 1.4455 - fn: 2.0000 - fp: 1.0000 - tn: 19.0000 - tp: 1.0000 - precision: 0.5000 - recall: 0.3333 - acc: 0.8696
     1/1 [==============================] - 0s 329ms/step - loss: 0.5693 - fn: 2.0000 - fp: 1.0000 - tn: 19.0000 - tp: 1.0000 - precision: 0.5000 - recall: 0.3333 - acc: 0.8696
     1/1 [==============================] - 1s 639ms/step - loss: 0.4051 - fn: 2.0000 - fp: 0.0000e+00 - tn: 20.0000 - tp: 1.0000 - precision: 1.0000 - recall: 0.3333 - acc: 0.9130
     1/1 [==============================] - 1s 864ms/step - loss: 0.4509 - fn: 2.0000 - fp: 0.0000e+00 - tn: 20.0000 - tp: 1.0000 - precision: 1.0000 - recall: 0.3333 - acc: 0.9130
     1/1 [==============================] - 1s 841ms/step - loss: 0.5458 - fn: 2.0000 - fp: 2.0000 - tn: 18.0000 - tp: 1.0000 - precision: 0.3333 - recall: 0.3333 - acc: 0.8261
```

Figure 10. Evaluation of Deep Learning Models with Data Augmentation

4.2 Discussion

In most of the implemented evaluation metrics such as accuracy, precision, f1-score and specificity, the proposal data augmentation technique allows to improve the results of most implemented models.

With the results obtained, it can be stated that with the proposal data augmentation technique, most of the deep learning models have as their main strength the ability to predict True Negatives (TN), that is, the ability to predict graduates who are not going to execute a research project. Implemented models reach a specificity between 90.00% and 100%. This is very important because, based on academic and socioeconomic features, it will be possible to detect in a timely manner with high accuracy the graduates who will have greater difficulties in executing a research project to obtain their professional degree.

Regarding the ability to predict True Positives (TP), despite the improvements in results with data augmentation, there is still a lot to be done, specifically with sensitivity and f1-score. The sensitivity and f1-score range between 33.33% and 66.67% which is quite poor. Also, sensitivity is the one metric that most models fail improving. In six of the seven implemented models, the values stay the same, and in one case (CNN+LSTM) it gets worse.

Something important to highlight is the difference between sensitivity and precision, both evaluate the ability to predict True Positives (TP), however, in this study, with data augmentation, precision presents better results than sensitivity, this is due to that with data augmentation the models manage to reduce the number of False Positives (FP) directly affecting the precision results, thus four models (DNN, LSTM, LSTM+GRU and CNN+LSTM) present perfect precision. In the case of sensitivity, unlike precision, it is based on False Negatives (FN), without data augmentation this value was already low and with data augmentation it remained without many changes.

5 Conclusion and Future Work

5.1 Conclusion

According to the obtained results, it can be affirmed that the proposal data augmentation technique allows to improve the performance of deep learning models for prediction of research project execution in graduates. Thus, of the seven implemented models, in most of them, a considerable improvement is achieved in terms of accuracy, precision, f1-score and specificity. Having models with a high ability to predict True Negatives (TN), it will allow in a timely manner to detect graduates with greater difficulties to execute research projects.

5.2 Future Work

According to the results, the main weakness of the implemented models is the ability to predict True Positives (TP), this can be improved through experimentation with other values for the parameters *snumber* and *max_distance* of the proposal data augmentation technique, where the number of synthetic records could be increased or also increase the distance between the values 0 and 1 with a lower *max_distance*.

In addition, the best prediction features can be selected using the correlation matrix as well as other state-of-the-art feature selection techniques such as Chi-Squared, Random Forest, XGBoost among others. Another hybrid and non-hybrid deep learning architectures could be implemented, as well some ensemble models.

Finally, another approach to improve the prediction ability of True Positives (TP) is feature generation which from the available features and a properly feature selection process, better prediction features can be found and build better deep learning models.

References

[1] SUNEDU, "II Informe Bienal sobre la Realidad Universitaria en el Perú," 2020. [Online]. Available: https://cdn.www.gob.pe/uploads/document/file/1230044/Informe Bienal.pdf.

[2] INEI, "Encuesta Nacional de Egresados Universitarios," 2015. [Online]. Available: https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1298/Libro.pdf.

[3] A. Flores, H. Tito-Chura, and H. Apaza-Alanoca, "Data Augmentation for Short-Term Time Series Prediction with Deep Learning," in *Lecture Notes in Networks and Systems*, 2021, vol. 284, doi: 10.1007/978-3-030-80126-7_36.

[4] M. Fang, N. Damer, F. Boutros, F. Kirchbuchner, and A. Kuijper, "The overlapping effect and fusion protocols of data augmentation techniques in iris PAD," in *Machine Vision and Applications*, 2022, vol. 33, no. 1, doi: 10.1007/s00138-021-01256-9.

[5] E. Okafor, R. Smit, L. Schomaker, and M. Wiering, "Operational data augmentation in classifying single aerial images of animals," 2017, doi: 10.1109/INISTA.2017.8001185.

[6] S. Y. Feng *et al.*, "A Survey of Data Augmentation Approaches for NLP," 2021, doi: 10.18653/v1/2021.findings-acl.84.

[7] H. Shi, K. Livescu, and K. Gimpel, "Substructure Substitution: Structured Data Augmentation for NLP," 2021, doi: 10.18653/v1/2021.findings-acl.307.

[8] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," *PLoS ONE*, vol. 16, no. 7 July. 2021, doi: 10.1371/journal.pone.0254841.

[9] K. M. Rashid and J. Louis, "Times-series data augmentation and deep learning for construction equipment activity recognition," *Adv. Eng. Informatics*, vol. 42, 2019, doi: 10.1016/j.aei.2019.100944.

[10] J. Sivakumar, K. Ramamurthy, M. Radhakrishnan, and D. Won, "Synthetic sampling from small datasets: A modified mega-trend diffusion approach using k-nearest neighbors," *Knowledge-Based Syst.*, vol. 236, 2022, doi: 10.1016/j.knosys.2021.107687.

[11] S. A. Shahab, I. Tassaduq, A. Haque, A. Naheed, M. Zafar, and I. Zakria, "Study of the Factors that Influence the Completion of the Thesis of Master of Health Professions Education Graduates: A Qualitative Study," *J. Rawalpindi Med. Coll.*, vol. 25, no. 3, 2021, doi: 10.37939/jrmc.v25i3.1711.

[12] C. Maphosa and N. Wadesango, "Exploring Student-specific Factors Affecting PhD Theses Completion," *J. Commun.*, vol. 7, no. 1, 2016, doi: 10.1080/0976691x.2016.11884890.

[13] E. S. Ginting and A. H. Hutasoit, "FACTORS AFFECTING STUDENTS' THESIS COMPLETION ON DEPARTMENT OF MANAGEMENT STIE MIKROSKIL," *J. Tarb.*, vol. 27, no. 2, 2021, doi: 10.30829/tar.v27i2.843.

[14] J. Nouri, K. Larsson, and M. Saqr, "Identifying Factors for Master Thesis Completion and Non-completion Through Learning Analytics and Machine Learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11722 LNCS, doi: 10.1007/978-3-030-29736-7_3.

[15] P. Shah and T. Shah, "Identification of Breast Tumor Using Hybrid Approach of Independent Component Analysis and Deep Neural Network," *Int. J. Intell. Syst. Appl. Eng.*, vol. 9, no. 4, 2021, doi: 10.18201/IJISAE.2021473642.

[16] L. K. Tsou *et al.*, "Comparative study between deep learning and QSAR classifications for TNBC inhibitors and novel GPCR agonist discovery," *Sci. Rep.*, vol. 10, no. 1, 2020, doi: 10.1038/s41598-020-73681-1.

[17] A. Nabil, M. Seyam, and A. Abou-Elfetouh, "Prediction of Students' Academic Performance Based on Courses' Grades Using Deep Neural Networks," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3119596.

[18] M. A. Khan and Y. Kim, "Cardiac arrhythmia disease classification using LSTM deep learning approach," *Comput. Mater. Contin.*, vol. 67, no. 1, 2021, doi: 10.32604/cmc.2021.014682.

[19] M. Djerioui, Y. Brik, M. Ladjal, and B. Attallah, "Heart Disease prediction using MLP and LSTM models," 2020, doi: 10.1109/ICEE49691.2020.9249935.

[20] A. W. Salehi, P. Baglat, B. B. Sharma, G. Gupta, and A. Upadhya, "A CNN Model: Earlier Diagnosis and Classification of Alzheimer Disease using MRI," 2020, doi: 10.1109/ICOSEC49089.2020.9215402.

[21] A. Ashraf, S. Naz, S. H. Shirazi, I. Razzak, and M. Parsad, "Deep transfer learning for alzheimer neurological disorder detection," *Multimed. Tools Appl.*, vol. 80, no. 20, 2021, doi: 10.1007/s11042-020-10331-8.

[22] T. Li, M. Hua, and X. Wu, "A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM2.5)," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2971348.

[23] A. Tasdelen and B. Sen, "A hybrid CNN-LSTM model for pre-miRNA classification," *Sci. Rep.*, vol. 11, no. 1, 2021, doi: 10.1038/s41598-021-93656-0.

[24] J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Tree-Structured Regional CNN-LSTM Model for Dimensional Sentiment Analysis," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, 2020, doi: 10.1109/TASLP.2019.2959251.

[25] M. Pan *et al.*, "Water Level Prediction Model Based on GRU and CNN," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2982433.

[26] J. Yu, X. Zhang, L. Xu, J. Dong, and L. Zhangzhong, "A hybrid CNN-GRU model for predicting soil moisture in maize root zone," *Agric. Water Manag.*, vol. 245, 2021, doi: 10.1016/j.agwat.2020.106649.

[27] Q. Wang, Y. Li, Y. Wang, and J. Ren, "An automatic algorithm for software vulnerability classification based on CNN and GRU," *Multimed. Tools Appl.*, vol. 81, no. 5, 2022, doi: 10.1007/s11042-022-12049-1.

[28] S. Krishnan, P. Magalingam, and R. Ibrahim, "Hybrid deep learning model using recurrent neural network and gated recurrent unit for heart disease prediction," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, pp. 5467–5476, 2021, doi: 10.11591/ijece.v11i6.pp5467-5476.

[29] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," *Chaos, Solitons and Fractals*, vol. 140, 2020, doi: 10.1016/j.chaos.2020.110212.

[30] E. Ahmadzadeh, H. Kim, O. Jeong, N. Kim, and I. Moon, "A Deep Bidirectional LSTM-GRU Network Model for Automated Ciphertext Classification," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3140342.

[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, 2002, doi: 10.1613/jair.953.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning, ICML 2015*, 2015, vol. 1.