Multi-Step Forecasting of Guillain Barré Cases Using Deep Learning

[1]Anibal Flores, [2]Hugo Tito-Chura
[1]Universidad Nacional de Moquegua, Prolong. Ancash S/N, Moquegua, Peru
[2]Universidad Nacional de Moquegua, Ciudad Jardin S/N, Ilo, Peru
E-mail: afloresg@unam.edu.pe, etitoc@unam.edu.pe

Keywords: Guillain Barré Syndrome, forecasting, deep learning, linear interpolation, data augmentation

Received: June 4, 2024

This work proposes a new data augmentation technique based on linear interpolation for time series regression. Weekly Guillain Barré syndrome cases of Peru collected from 2019 to 2023 were used as a study case. The methodology includes the data splitting into training, validation, and testing subsets, data augmentation of training data, min/max normalization, training of models, forecasting from 1 to 5 steps, and the respective evaluation of results. For the experiments, five deep learning models were implemented, including Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), Bidirectional GRU (BiGRU), and LSTM with attention layer (LSTMA), and experiments were carried out with different prediction steps 1, 2, 3, 4, and 5. The results show that of 50 implemented models with the proposed data augmentation, 41 improved in terms of MAPE, the improvements range between 0.27% and 338.75%. On average, the model that improved the most was the Gated Recurrent Unit (GRU5) which used data augmentation with 5 synthetic items.

Povzetek: Raziskava uvaja novo tehniko dopolnjevanja podatkov za napoved več korakov Guillain-Barré sindroma z uporabo globokega učenja.

1 Introduction

Guillain-Barré syndrome (GBS) is a rare disorder[1]–[5] in which the body's immune system attacks the nerves. Usually, the first symptoms are weakness and tingling[6] in the hands and feet. These sensations can spread quickly and, over time, paralyze the entire body. The most severe form of Guillain-Barré syndrome can paralyze the human body and is considered a medical emergency. The exact cause of Guillain-Barré syndrome is unknown[7]–[9]; however, two-thirds of patients exhibit a flu-like syndrome caused by gastrointestinal and respiratory tract infections before four weeks of the onset of weakness [10].

According to the literature, few works have been found related to the use of machine learning and deep learning that address Guillain-Barré Syndrome, in three of the related works the diagnosis of Guillain Barré syndrome was addressed, including [11], [12] and [13], where the authors worked with a dataset of 129 patients and tried to diagnose the syndrome with 26 clinical features, using machine learning techniques such as SVM, JRip, C4.5, Random Forest and others, the best accuracy achieved was obtained by Random Forest with 0.9366 outperforming other models such as JRip, SVM, C4.5, C5.0, Boosting, Bagging and others. In [14] and [15] a dataset of 122 patients was used, and three classifiers were implemented including JRip, SVM, and C4.5 to predict the need for mechanical ventilation in Guillain-Barré Syndrome patients, the best accuracy was obtained by OneR with 0.9119.

In this work, GBS was not addressed as a classification problem, but as a regression problem. Time series regression or forecasting is used to predict future behavior [16] of the time series. Forecasting Guillain-Barré syndrome cases will allow health authorities to know the number of future cases of GBS and make the corresponding decisions.

Five deep learning models were implemented including Long Short-Term Memory (LSTM) [17], Bidirectional LSTM (BiLSTM) [18], Gated Recurrent Unit (GRU)[19], Bidirectional GRU (BiGRU), and LSTM with attention layer (LSTMA)[20]. Data augmentation was used to improve the preliminary results. Commonly data augmentation was used for deep learning models to solve underfitting [21], [22] and overfitting[23][24] problems. Underfitting is produced when a model is not well-tuned to the training set [25][26], and overfitting is when a model produces good predictions for data points in the training set but performs poorly on unknown samples[27].

In this work, a new data augmentation technique based on linear interpolation is proposed, which, unlike the literature data augmentation techniques for time series regression, such as time warping + jittering, linear, polynomial, spline interpolation, and others generate synthetic values between each pair of items in the time series. The proposal produces synthetic values as it is done in time series classification problems, more details are in the 2.2 section.

The main contributions of this work are listed below:

- A comparison of multi-step forecasting of five deep learning models for Guillain Barré syndrome cases.

- A new data augmentation technique based on linear interpolation for time series regression.
- A comparison of different data augmentation levels for time series regression.

The rest of the paper is structured in a Related Work section that describes the state-of-the-art works. A Methodology section that describes the steps for implementing the deep learning models including the data augmentation proposal. Then, a results and discussion section, where the achieved results are described, and discussed. Finally, the conclusion reached based on the achieved results is described.

2 Related works

2.1 Overview of forecasting models

Forecasting models have evolved. In the beginning, statistical models such as Simple Moving Average (SMA), Exponential Smoothing (ES), and Autoregressive Integrated Moving Average (ARIMA) gained great importance. The most well-known is ARIMA[28] introduced by Box-Jenkins in the 1970s.

Later, models based on machine learning appeared, such as K-Nearest Neighbors (KNN)[29], Support Vector Regression (SVR)[30], Random Forest (RF), XGBoosting (XGB), and Multilayer Perceptron (MLP)[31].

Currently, those based on Deep Learning have become very popular, especially those based on Recurrent Neural Networks such as LSTM[32], GRU, BiLSTM[33], BiGRU, RNNs with attention layers, and more recently those based on Transformers[34].

2.2 Related works on Guillain Barre forecasting

As mentioned above very few related works about Guillain Barré Syndrome can be found in the literature, some of them are displayed in Table 1.

Table 1: Related works

Authors	Year	Method	Dataset	Results
[14]	2017	C4.5	122	Accuracy: 0.8925
[13]	2018	Random Forest	129	Accuracy: 0.9366
[15]	2018	OneR	122	Accuracy: 0.9119
[12]	2020	JRip	122	ROC: 0.9507
[11]	2021	SVM	129	Accuracy: 0.6584

According to Table 1, most of the related works focused on the implementation of classification models using machine learning techniques for the GBS diagnosis and the need of mechanical ventilation. Hence the importance of implementing regression models for Guillain Barré Syndrome weekly cases using deep learning techniques such as LSTM, GRU, BiLSTM, BiGRU, and LSTM with attention layers.

3 Methodology

This section describes the methodology to implement the proposed data augmentation technique and the deep learning models. Figure 2 shows a graphical summary of this process.

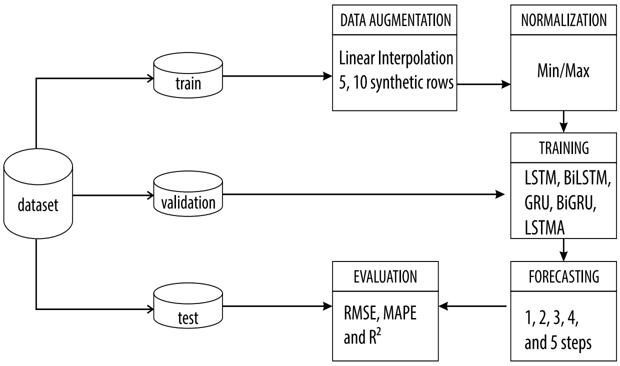


Figure 2: Methodology

3.1 Data collection

The dataset of Guillain Barré syndrome cases in Peru was downloaded from https://www.dge.gob.pe/sala-sgb/SALA-SGB.html. It contains cases from 2019 to 2024. But, for this study, data from 2019 to 2023 were used. Table 2 shows the characteristics of this dataset.

Table 2: Dataset characteristics

Count	Mean	Std	Min	Max
260	9.49	22.44	0.00	305.00

Data from 2019 to 2023 were used for this study, equivalent to 260 weeks. 208 weeks were used for training, 13 for validation, and 39 for testing.

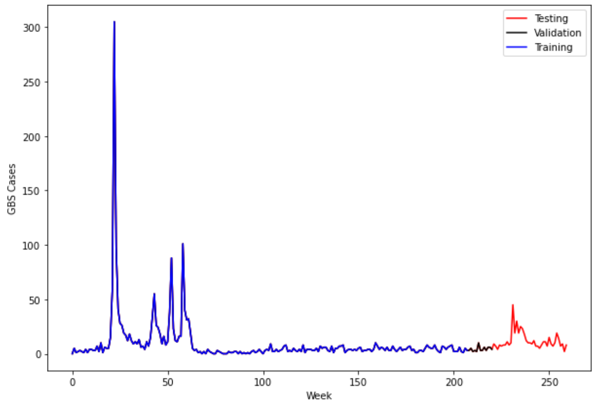


Figure 1: Dataset subsets

2.2 Data augmentation

Data augmentation was performed to enrich training data and improve the accuracy of the results of the deep learning-based models. In time series, data augmentation was mainly performed for classification tasks [35]–[37].

In [38], three data augmentation techniques for time series regression were described, time warping and jittering, linear and polynomial interpolation, they were used to insert *n* synthetics items between each pair of time series values.

In this work and the related work, linear interpolation for data augmentation was proposed, but the main difference lies in how the time series items are interpolated. While in [38] *ns* synthetic items were directly generated between each pair of time series items, in this work the training data was first organized in a 4 x 52 matrix, each row contains the data of one year, thus the first row contains the data for the year 2019, the second row for the year 2020 and so on. Then, linear interpolation was used to generate *ns* synthetic items between each week of the 2019 and 2020 years. Then between each week of the 2020 and 2021 years. Finally, between each week of the 2021 and 2022 years. In other words, the process generates new synthetic years instead of just new items. For implementing the deep learning models, data structured in a one-dimensional array is required, so the corresponding conversion from matrix to one-dimensional array was made.

Figure 3, shows the main difference between literature data augmentation techniques and the proposed data augmentation of this work. The black dots are the original values, and the red dots are the synthetic values.

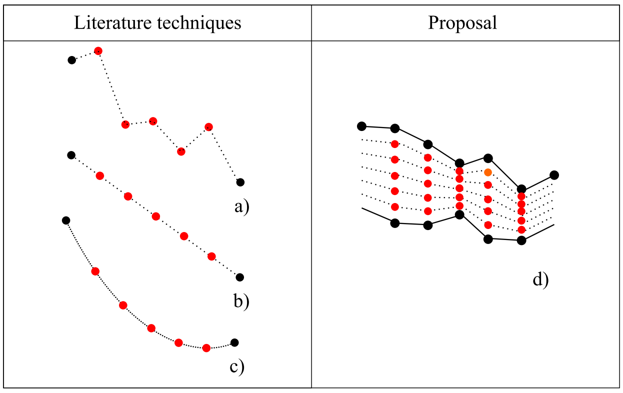


Figure 3: Literature data augmentation techniques vs Proposal. a) Time warping + jittering, b) Linear interpolation, c) Polynomial interpolation and d) Proposal

According to Figure 3, the main disadvantage of literature data augmentation techniques for time series regression is that original data frequencies and relations are altered, so the predictions preserve the new frequencies and relations, and to obtain the real predictions, the predictions that correspond to synthetic values must be discarded. This problem does not arise with the proposal, since the frequencies and relations in the data sequences are not altered.

In this work, *ns*=5 and ns=*10* were used to generate synthetic items. Figure 4 shows five synthetic years between 2019 and 2020 year. And, as can be seen, the frequencies and ratios of the original data are not altered, and the synthetic data also retain similar characteristics as the original data used to generate it.

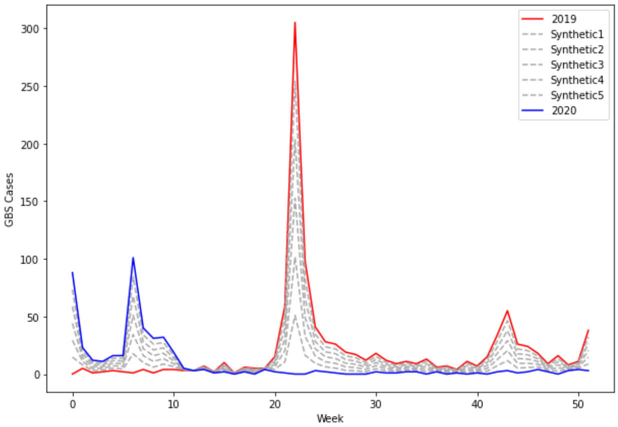


Figure 4: Data augmentation with 5 synthetic rows

The final length of training data including synthetic values can be estimated using equation (1).

$$length = (yrs + (yrs - 1) * ns) * wks \qquad (1)$$

Where *length* is the final length of training data including data augmentation, *yrs* are the training years, *ns* the number of synthetic years, and *wks* the number of weeks per each year. Table 3 shows training data including synthetic values.

Table 3: Training data including synthetic values

yrs	wks	ns	Total
4	52	5	988
4	52	10	1768

Figure 5 shows the algorithm of the proposal data augmentation technique.

According to Figure 5, the data augmentation algorithm requires as inputs *nr* (number of rows), *nc*, (number of columns), *nsyns* (number of synthetic items), and *data* (data matrix).

The first loop *i* allows to traverse the rows of the data matrix; the loop *j* allows to traverse the columns of each row of the data matrix; row *i* and row *i+1* are used to generate the synthetic items. In loop *j*, the distances between each pair of values in row *i* and row *i+1* are estimated. In loop *k*, the distances obtained in the prior loop are used to generate the linear synthetic items.

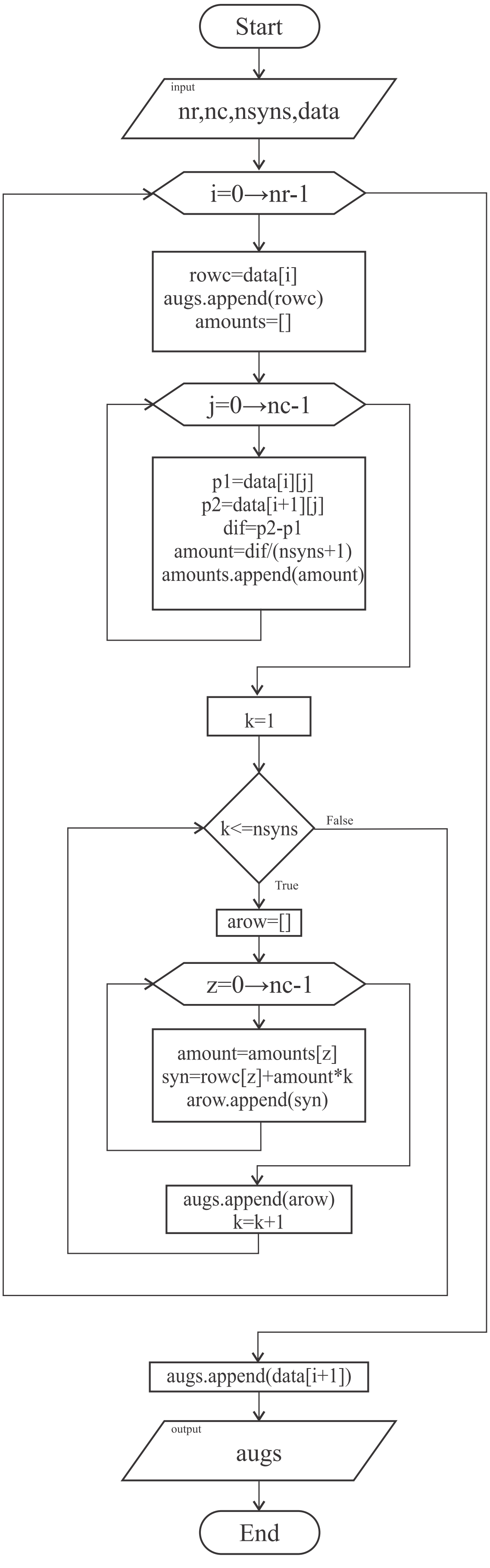


Figure 5: Algorithm of the proposed data augmentation

2.3 Normalization

Normalization consists of transforming the values of a set of data so that, they are in a specific range of values, typically between 0 and 1. This work used normalization to help models converge faster [39][40]. Data were normalized using the [0-1] scale, through the min/max normalization, according to equation (2).

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad (2)$$

Where x' is the normalized value, x is the value to be normalized, *min(x)* is the min value in the *x* vector, and *max(x)* is the max value in the *x* vector.

2.4 Model implementation

For deep learning models, data were structured in features and labels, with 52 features for each row, and depending on step size prediction, labels contain 1, 2, 3, 4, or 5 columns.

In this work, the deep learning-based models were implemented using Google Colaboratory and the Tensorflow 2.15.0 library.

The hyperparameters for the implemented deep learning models are shown in Table 3.

Table 4: Hyperparameters of the deep learning models

Model	**hyperparameters**
LSTM	[100, 50, 50, ps*], droprate: 20,20,20
BiLSTM	[100, 50, 50, ps*], droprate: 20,20,20
GRU	[100, 50, 50, ps*], droprate: 20,20,20
BiGRU	[100, 50, 50, ps*], droprate: 20,20,20
LSTMA	[100, ATT, ps*]

* ps is the prediction steps: 1, 2, 3, 4, or 5.

In this work, just LSTM and GRU were optimized using GridSearchCV[41]. BiLSTM, BiGRU, and LSTMA used similar architecture to that of the main models.

According to Table 4, the Recurrent Neural Networks-based models such as LSTM, BiLSTM, GRU, and BiGRU have four layers, the first contains 100 units, the second and third 50 units, and the output layer presents the units equivalent to the predictions steps that can be 1, 2, 3, 4 or 5. The LSTM with attention layer (LSTMA) is simpler than the previous ones since it only presents 3 layers, the first contains 100 units, the second is the attention layer, and the third is the output layer, it presents the units equivalent to the prediction steps, like the previous ones. In the first four models, drop rates of 20% were used to avoid overfitting after each layer except the output layer. LSTMA did not use drop rates in any layer.

Models with no data augmentation obtained their best results with 100 epochs, while models with data augmentation (5 and 10) obtained their best results with 50 epochs.

All models used a learning rate of 0.001, adam as optimizer, mse and mae as loss functions, 100 as batch size.

2.5 Evaluation

The evaluation of implemented models was performed through Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and R^2, they are estimated using equations (3), (4), and (5).

$$\boldsymbol{RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Pi-Oi)^2}{n}}} \quad (3)$$

$$\boldsymbol{MAPE \ = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{(O_i-P_i)}{O_i} * 100\right|} \quad (4)$$

$$R^2 = \sum_{i=1}^{n}(P_i - \bar{O})^2 / \sum_{i=1}^{n}(O_i - \bar{O})^2 \quad (5)$$

Where Pi is the i^{th} item of the predicted vector P, Oi is the i^{th} item of the observed vector, n the length of P and O vectors, and $\bar{O}$ is the mean of the items in the vector O.

The RMSE was used to evaluate the error of the predictions in terms of GBS cases. The MAPE was used for evaluating the prediction error in percentage terms. And, the R^2 coefficient was used to measure the level of correlation between the predictions and the observed data.

4 Results and Discussions

4.1 Results

In this section, the achieved results are described.

Table 5: RMSEs of implemented models

Model	Prediction steps				
	1	2	3	4	5
LSTM	11.18	11.31	44.36	50.57	48.56
LSTM5	9.46	11.26	11.64	11.98	12.31
LSTM10	8.92	10.59	11.37	11.06	11.58
BiLSTM	9.76	10.85	21.97	12.98	27.11
BiLSTM5	9.49	14.20	12.00	13.21	12.68
BiLSTM10	10.71	12.51	12.59	12.48	12.01
GRU	9.06	**8.90**	39.15	35.63	32.71
GRU5	**8.33**	9.15	9.51	**9.56**	10.30
GRU10	8.68	9.39	9.80	10.03	10.67
BiGRU	10.01	10.37	37.50	35.19	39.16
BiGRU5	9.79	11.07	**9.13**	9.77	**10.23**
BiGRU10	9.41	10.51	10.65	10.69	11.21
LSTMA	11.77	11.45	14.36	11.65	10.87
LSTMA5	11.60	11.87	11.63	11.54	11.54
LSTMA10	11.26	11.45	11.29	11.63	10.93

In terms of RMSE, according to Table 5, it can be seen that most of the models managed to improve with data augmentation. For one-step predictions the model that improves the most is LSTM, passing through 11.18 (LSTM), 9.46 (LSTM5), and 8.92 (LSTM10). For two-step predictions, the model that improves the most is also LSTM, passing through 11.31 (LSTM), 11.26 (LSTM5), and 10.59 (LSTM10). For three-step predictions, the model that improves the most is also LSTM, passing through 44.36 (LSTM), 11.64 (LSTM5), and 11.37 (LSTM10). For four-step predictions, the model that improves the most is also LSTM passing through 50.57 (LSTM), 11.98 (LSTM5), and 11.05 (LSTM10). Finally, for five-step predictions the model that improves the most is LSTM passing through 48.56, 12.31, and 11.58.

In terms of RMSE, the models with lower RMSE are GRU5, GRU, BiGRU5, GRU5, and BiGRU5 with 8.33, 8.90, 9.13, 9.56 and 10.23 for 1, 2, 3, 4 and five steps respectively.

Table 6: MAPEs of implemented models

Model	Prediction steps				
	1	2	3	4	5
LSTM	65.72	66.69	366.66	397.32	402.41
LSTM5	53.26	50.71	66.27	69.63	77.82
LSTM10	42.87	56.34	65.51	58.57	65.91
BiLSTM	55.31	55.19	186.81	100.45	229.63
BiLSTM5	54.21	92.42	76.62	94.33	84.72
BiLSTM10	58.00	80.39	83.49	76.54	67.57
GRU	46.51	54.84	349.78	314.92	272.63
GRU5	39.67	42.96	48.97	47.41	51.59
GRU10	41.97	47.34	52.70	51.84	57.03
BiGRU	58.32	51.42	311.60	263.75	325.90
BiGRU5	50.71	62.29	45.46	47.41	49.70
BiGRU10	43.01	52.88	55.42	55.42	57.93
LSTMA	64.99	60.41	99.81	63.78	53.62
LSTMA5	63.63	65.45	63.56	62.40	62.67
LSTMA10	59.39	60.41	59.34	63.51	54.23

In terms of MAPE, according to Table 6, it can be seen that most of the models managed to improve with data augmentation. For one-step predictions the model that improves the most is LSTM, passing through 65.72% (LSTM), 53.26% (LSTM5), and 42.87% (LSTM10). For two-step predictions, the model that improves the most is GRU, passing through 54.84% (GRU), 42.96% (GRU5), and 47.34% (GRU10). For three-step predictions, the model that improves the most is also GRU, passing through 349.78% (GRU), 48.97% (GRU5), and 52.7% (GRU10). For four-step predictions, the model that improves the most is also GRU passing through 314.92% (GRU), 47.41% (GRU5), and 51.84 %(GRU10). Finally, for five-step predictions the model that improves the most is LSTM passing through 402.41%, 77.82%, and 65.91%.

In terms of MAPE, the models with lower MAPE are GRU5, GRU5, BiGRU5, GRU5, and GRU5 con 39.67%, 42.96%, 45.46%, 47.41%, and 51.59% for 1, 2, 3, 4 and 5 steps respectively.

Table 7: R^2s of implemented models

Model	Prediction steps				
	1	2	3	4	5
LSTM	0.00	0.00	0.01	0.01	0.01
LSTM5	**0.26**	0.04	0.04	0.03	0.06
LSTM10	0.20	0.10	0.06	0.03	**0.07**
BiLSTM	0.00	0.00	0.01	0.00	0.02
BiLSTM5	**0.26**	0.01	0.10	0.09	0.06
BiLSTM10	0.11	0.07	0.06	0.02	0.00
GRU	0.08	0.03	0.04	0.04	0.03

GRU5	0.23	0.19	0.15	0.14	0.06
GRU10	0.23	**0.24**	**0.20**	**0.15**	**0.07**
BiGRU	0.00	0.00	0.00	0.00	0.01
BiGRU5	0.13	0.12	0.14	0.10	0.05
BiGRU10	0.10	0.09	0.04	0.04	0.01
LSTMA	0.05	0.05	0.00	0.05	0.01
LSTMA5	0.05	0.06	0.05	0.06	0.05
LSTMA10	0.04	0.05	0.07	0.09	0.06

In terms of R^2, according to Table 7, it can be seen that all models managed to improve with data augmentation. For one-step predictions the model that improves the most is LSTM, passing through 0.00 (LSTM), 0.26 (LSTM5), and 0.20 (LSTM10). For two-step predictions, the model that improves the most is GRU, passing through 0.03 (GRU), 0.19 (GRU5), and 0.24 (GRU10). For three-step predictions, the model that improves the most is also GRU, passing through 0.04 (GRU), 0.15 (GRU5) and 0.20 (GRU10). For four-step predictions, the model that improves the most is also GRU passing through 0.04 (GRU), 0.14 (GRU5) and 0.15 (GRU10). Finally, for five-step predictions, the model that improves the most is LSTM passing through 0.01, 0.06, and 0.07 followed very closely by GRU with 0.03, 0.06, and 0.07.

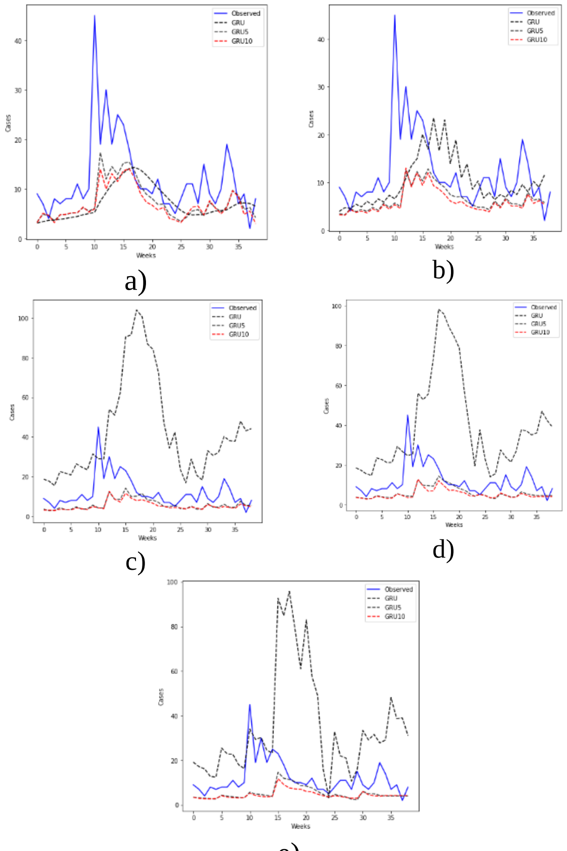
In terms of R^2, the models with higher R^2 are LSTM5, GRU10, GRU10, GRU10, and LSTM10 con 0.26, 0.24, 0.2, 0.15 and 0.07 for 1, 2, 3, 4 and 5 steps respectively.

Figure 6 graphically shows the predictions of GRU model which was the best model on average for 1, 2, 3, 4, and 5-step predictions.

It can be seen that for one-step predictions (Figure 6 a), the three versions of GRU present similar predictions; however, the versions with data augmentation GRU5 (MAPE 39.67%), and GRU10 (MAPE 41.97%) were slightly superior to the version with no data augmentation GRU (MAPE 46.51%), thus the data augmentation allowed obtaining greater detail in the predictions, that is, a less smooth curve.

For two-step predictions, similarly to one-step predictions, the GRU predictions (MAPE 54.84%) produced more error than the data augmentation versions GRU5 (MAPE 42.96%) and GRU10 (MAPE 47.34%) with a not very big difference.

For three-step predictions, the difference between the models with data augmentation GRU5 (MAPE 48.97%) and GRU10 (MAPE 52.7%) became quite noticeable in regards to the model with no data augmentation GRU (MAPE 349.78%).



a)

b)

c)

d)

e)

Figure 6: GRU predictions. a) 1-step, b) 2-step, c) 3-step, d) 4-step, and e) 5-step

For four-step and five-step predictions, something similar to three-step predictions occurred, the model with no data augmentation GRU presented a high error, greater than 250%, while the models with no data augmentation GRU5 and GRU10 remained more uniform.

As seen in Tables 4, 5, 6, and Figure 7, the greater the number of prediction steps, the performance of the implemented deep learning models is negatively affected. So, it can be stated that with the available data, predicting more than two steps was very complex for models with no data augmentation. For most of the implemented deep-learning models, the data augmentation allowed good improvements in the performance of most models, not being affected in the magnitude that the models with no data augmentation were, but there is still work to do for future studies, the errors still are high.

It is important to highlight that from all the implemented models, LSTM with attention layer (LSTMA) despite not presenting the best RMSE, MAPE, and R^2, was the only model that presented uniform results in the different prediction steps.

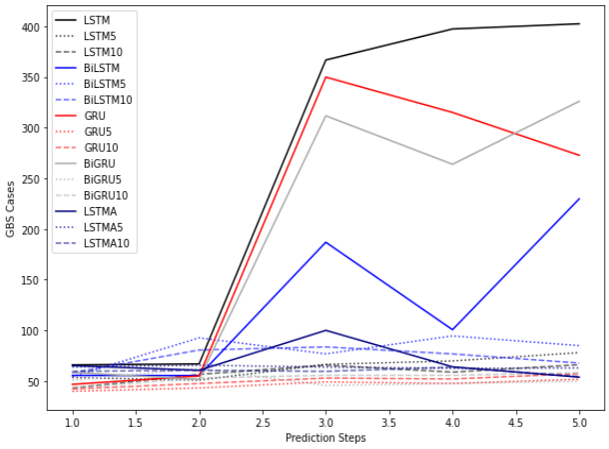


Figure 7: MAPEs of each prediction step

4.2 Discussions

Error analysis

According to Figure 6, it can be seen that the best model (GRU) has greater difficulty in predicting between weeks 9 and 19, where the largest number of cases occur. The data augmentation allows to reduce the error considerably, however, there is no significant difference between the data augmentation of 5 items regarding the 10 items augmentation. Also, according to Figure 1, testing data is not similar to training data, especially data between weeks 9 and 19, so the implemented models are not able to learn this type of data, producing high errors in the predictions. An alternative to solve this would be to modify the data augmentation strategy; instead of using the weeks sequentially, they could be combined randomly.

Computational cost in seconds

The models were implemented in Python language in Jupyter IDE with a processor AMD Ryzen 7 4700U with Radeon Graphics 2.0 GHz, and RAM 12GB.

Table 8: Computation cost in seconds

Model	Data augmentation		
	0	5	10
LSTM	35	897	1734
GRU	28	611	1621
BiLSTM	48	6450	10786
BiGRU	61	2855	5237
LSTM ATT	15	365	973

According to Table 8, computation cost increases considerably with data augmentation, however, the improvements in results justify the implementation, being five the optimal number of synthetic items.

Broader impact

The GRU model that presented the best performance in forecasting Guillain Barré Syndrome cases can be integrated into public health planning through a web application accessed through web environments from desktop machines or mobile devices. The architecture for this application can be seen in Figure 8. The Back End would run in Python with Flask framework and, the Front End with HTML5, JavaScript, and CSS.

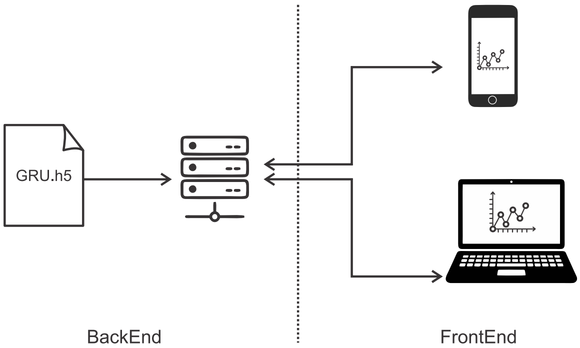


Figure. 8 Architecture of a web app

Limitations

One of the main limitations of this work is the amount of data obtained, which is limited to what is currently available in the repository of the Ministry of Health of Peru. Another limitation is the configuration of the hyperparameters of three of the implemented neural networks, BiLSTM, BiGRU, and LSTMA, they used similar architecture of the main models LSTM and GRU.

5 Conclusion and future work

5.1 Conclusion

According to the results achieved, it can be stated that the proposed data augmentation technique based on linear interpolation in terms of MAPE allowed improvement in most of the implemented models such as LSTM, BiLSTM, GRU, BiGRU and LSTMA, thus, from 50 implemented models with data augmentation, 41 of them were improved, being these improvements between 0.27% and 338.75%, so the proposal is a good alternative for data augmentation in Guillain Barré syndrome time series forecasting.

5.2 Future work

The data augmentation proposal improved most of the implemented models; however, despite the improvements, the best MAPE achieved is 39.67% (GRU5) for one-step forecasting, which shows that there is still much room for improvement. Thus, some future improvements that can be made include the exploration of other deep learning architectures e.g. other models with attention layers such as CNN-LSTM, BiLSTM, GRU, BiGRU, models based on Transformers, ensemble models, and hybrid models.

In addition, multivariate models could be considered, including variables e.g. temperature, and precipitation.

Table 9: Description of abbreviations

Abbreviation	Explanation
GBS	Guillain Barre Syndrome
ARIMA	Autoregressive Integrated Moving Average
SMA	Simple Moving Average
SVR	Support Vector Regression
KNN	K Nearest Neighbors
RF	Random Forests
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
BiLSTM	Bidirectional Long Short-Term Memory
BiGRU	Bidirectional Gated Recurrent Unit
LSTMA	LSTM with attention mechanism
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
HTML	Hyper Text Markup Language

6 References

[1] S. A, A. K, A. H, A. M, and H. A, “COVID-19 and Guillain-Barre Syndrome Case Report,” J. Neurol. Neurobiol., vol. 6, no. 4, 2020. https://doi.org/10.16966/2379-7150.169

[2] Uwagboe UC and Isikhuemen ME, “Guillain-Barre syndrome in pregnancy: A rare disease with challenging management,” Ibom Med. J., vol. 16, no. 3, 2023. https://doi.org/10.61386/imj.v16i3.345

[3] J. A. Vega-Fernández, D. O. Suclupe-Campos, M. M. Coico-Vega, and F. R. Aguilar-Gamboa, “Viral etiology associated with Guillain-Barré Syndrome: seeking an answer to the idiopathic,” Rev. la Fac. Med. Humana, vol. 22, no. 3, 2022. https://doi.org/10.25176/rfmh.v22i3.2993

[4] T. Hasan, M. Khan, F. Khan, and G. Hamza, “Case of Guillain-Barré syndrome following COVID-19 vaccine,” BMJ Case Rep., vol. 14, no. 6, 2021. https:/doi.org/10.1136/bcr-2021-243629

[5] L. Súkeníková et al., “Autoreactive T cells target peripheral nerves in Guillain–Barré syndrome,” Nature, vol. 626, no. 7997, 2024. https://doi.org/10.1038/s41586-023-06916-6

[6] N. Khalayli, D. Haj Ali, and M. Kudsi, “Guillain–Barre syndrome complicating knee infectious arthritis,” Clin. Case Reports, vol. 11, no. 8, 2023. https://doi.org/10.1002/ccr3.7761

[7] M. Wajih Ullah, A. Qaseem, and A. Amray, “Post Vaccination Guillain Barre Syndrome: A Case Report,” Cureus, 2018. https://doi.org/10.7759/cureus.2511

[8] E. Vladislavov, “A post COVID-19 Guillain-Barre syndrome,” J. Neurol. Sci., vol. 429, 2021. https://doi.org/10.1016/j.jns.2021.119847

[9] O. K. Koloskova, R. V. Tkachuk, M. N. Garas, T. M. Bilous, L. I. Romanchuk, and D. I. Sorochan, “Guillain - Barré syndrome following asymptomatic COVID-19 in an adolescent: a clinical case,” Child’s Heal., vol. 17, no. 1, 2022. https://doi.org/10.22141/2224-0551.17.1.2022.1491

[10] Y. Zhang, C. Huang, W. Lu, and Q. Hu, “Case Report: Delayed Guillain-Barré syndrome following trauma: A case series and manage considerations,” Front. Surg., vol. 9, 2022. https://doi.org/10.3389/fsurg.2022.903334

[11] D. Alarcón-Narváez, J. Hernández-Torruco, B. Hernández-Ocaña, O. Chávez-Bosquez, J. Marchi, and J. J. Méndez-Castillo, “Toward a machine learning model for a primary diagnosis of Guillain-Barré syndrome subtypes,” Health Informatics J., vol. 27, no. 3, 2021. https://doi.org/10.1177/14604582211021471

[12] M. Torres-Vásquez, O. Chávez-Bosquez, B. Hernández-Ocaña, and J. Hernández-Torruco, “Classification of guillain-barre syndrome subtypes using sampling techniques with binary approach,” Symmetry (Basel)., vol. 12, no. 3, 2020. https://doi.org/10.3390/SYM12030482

[13] J. Canul-Reich, J. Hernández-Torruco, O. Chávez-Bosquez, and B. Hernández-Ocaña, “A predictive model for guillain-Barré syndrome based on ensemble methods,” Comput. Intell. Neurosci., vol. 2018, 2018. https://doi.org/10.1155/2018/1576927

[14] J. Hernández-Torruco, J. Canul-Reich, and O. Chávez-Bosquez, “Predicting the need of mechanical ventilation in guillain-barré patients using machine learning algorithms with relevant features,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2017, vol. 10062 LNAI. https://doi.org/10.1007/978-3-319-62428-0_20

[15] J. Hernández-Torruco, J. Canul-Reich, and D. L. Román, “Rule based classifiers for diagnosis of mechanical ventilation in Guillain-Barré syndrome,” in Advances in Intelligent Systems and Computing, 2018, vol. 620. https://doi.org/10.1007/978-3-319-62410-5_22

[16] V. Sohrabpour, P. Oghazi, R. Toorajipour, and A. Nazarpour, “Export sales forecasting using artificial intelligence,” Technol. Forecast. Soc. Change, vol. 163, 2021. https:/doi.org/10.1016/j.techfore.2020.120480

[17] J. S. Sepp Hochreiter, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, 1997. https://doi.org/10.1162/neco.1997.9.8.1735

[18] H. Liu, Z. Duan, and C. Chen, “A hybrid multi-resolution multi-objective ensemble model and its application for forecasting of daily PM2.5 concentrations,” Inf. Sci. (Ny)., vol. 516, 2020. https://doi.org/10.1016/j.ins.2019.12.054

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Gated Recurrent Neural Networks on Sequence Modeling,” arXiv, 2014. https://arxiv.org/pdf/1412.3555

[20] V. H. Nguyen et al., “Deep learning models for forecasting dengue fever based on climate data in Vietnam,” PLoS Negl. Trop. Dis., vol. 16, no. 6, 2022. https://doi.org/10.1371/journal.pntd.0010509

[21] B.-C. Jo et al., “DAGAM: Data Augmentation with Generation And Modification,” arXiv, 2022. https://arxiv.org/abs/2204.02633

[22] M. S. H. Shovon, S. J. Mozumder, O. K. Pal, M. F. Mridha, N. Asai, and J. Shin, “PlantDet: A Robust Multi-Model Ensemble Method Based on Deep Learning For Plant Disease Detection,” IEEE Access, vol. 11, 2023. https://doi.org/10.1109/ACCESS.2023.3264835

[23] L. I. Venturott and P. M. Ciarelli, “Data Augmentation for improving Hate Speech Detection on Social Networks,” 2020. https://doi.org/10.1145/3428658.3431760

[24] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” J. Big Data, vol. 6, no. 1, 2019. https://doi.org/10.1186/s40537-019-0197-0

[25] E. Halawa, Y. Bindar, A. Purqon, and W. Srigutomo, “Prediction of Carbon Monoxide Concentration with Variation of Support Vector Regression Kernel Parameter Value,” J. Math. Fundam. Sci., vol. 54, no. 1, 2022. https://doi.org/10.5614/j.math.fund.sci.2022.54.1.3

[26] E. Walker, “Regression Modeling Strategies,” Technometrics, vol. 45, no. 2, 2003. https://doi.org/10.1198/tech.2003.s158

[27] L. Bottou and O. Bousquet, “The tradeoffs of large scale learning,” 2008. https://doi.org/10.7551/mitpress/8996.003.0015

[28] E. Dave, A. Leonardo, M. Jeanice, and N. Hanafiah, “Forecasting Indonesia Exports using a Hybrid Model ARIMA-LSTM,” in Procedia Computer Science, 2021, vol. 179. https://doi.org/10.1016/j.procs.2021.01.031

[29] F. Li and G. Jin, “Research on power energy load forecasting method based on KNN,” Int. J. Ambient Energy, vol. 43, no. 1, 2022, https://doi.org/10.1080/01430750.2019.1682041

[30] J. Li, X. Li, K. Wang, and G. Cui, “Atmospheric pm2.5 prediction based on multiple model adaptive unscented kalman filter,” Atmosphere (Basel)., vol. 12, no. 5, 2021. https://doi.org/10.3390/atmos12050607

[31] C. H. Cordova, M. N. L. Portocarrero, R. Salas, R. Torres, P. C. Rodrigues, and J. L. López-Gonzales, “Air quality assessment and pollution forecasting using artificial neural networks in Metropolitan Lima-Peru,” Sci. Rep., vol. 11, no. 1, 2021. https://doi.org/10.1038/s41598-021-03650-9

[32] M. El Ghazi and N. Aknin, “Optimizing Deep LSTM Model through Hyperparameter Tuning for Sensor-Based Human Activity Recognition in Smart Home,” Inform., vol. 47, no. 10, 2023. https://doi.org/10.31449/inf.v47i10.5268

[33] R. Ranjan and A. K. Daniel, “CoBiAt: A Sentiment Classification Model Using Hybrid ConvNet-Dual-LSTM with Attention Techniques,” Inform., vol. 47, no. 4, 2023. https://doi.org/10.31449/inf.v47i4.3911

[34] A. Y. Yldz, E. Koc, and A. Koc, “Multivariate Time Series Imputation With Transformers,” IEEE Signal Process. Lett., vol. 29, 2022. https://doi.org/10.1109/LSP.2022.3224880

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

[36] 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. https://doi.org/10.1371/journal.pone.0254841

[37] H. Wu and H. Liu, “Non-intrusive load transient identification based on multivariate LSTM neural network and time series data augmentation,” Sustain. Energy, Grids Networks, vol. 27, 2021. https://doi.org/10.1016/j.segan.2021.100490

[38] A. Flores, H. Tito-Chura, D. Centty-Villafuerte, and A. Ecos-Espino, “Comparison of Data Augmentation Techniques for pm2.5 Time Series Prediction,” Int. J. Intell. Syst. Appl. Eng., vol. 12, no. 3, pp. 2312–2319, 2024. https://ijisae.org/index.php/IJISAE/article/view/5699

[39] D. Li, T. Jiang, and M. Jiang, “Norm-in-Norm Loss with Faster Convergence and Better Performance for Image Quality Assessment,” 2020. https://doi.org/10.1145/3394171.3413804

[40] Q. Yan, B. Yang, W. Wang, B. Wang, P. Chen, and J. Zhang, “Apple leaf diseases recognition based on an improved convolutional neural network,” Sensors (Switzerland), vol. 20, no. 12, 2020. https://doi.org/10.3390/s20123535

[41] M. A. Haq and M. A. R. Khan, “Dnnbot: Deep neural network-based botnet detection and classification,” Comput. Mater. Contin., vol. 71, no. 1, 2022. https://doi.org/10.32604/cmc.2022.020938