Deep Learning Models for Imbalanced Time-Series Clinical Data

Sergio Ramirez, Kshitij Parab, Sharvari Kalgutkar, Vansh Jain

I. Abstract

Imbalanced data is a known plague that burdens numerous deep-learning models. However, this problem becomes even more detrimental with predictive models and their ability to classify unseen data accurately. While there have been some attempts to resolve the imbalance issue, the problem becomes more complex when dealing with time-series data specifically. If you consider the medical field, this issue becomes more problematic, where a large percentage of clinical datasets are of the time-series type. Good accuracy is of even greater importance when lives are at stake. In our report, we will conduct an empirical study investigating the potential of combining universally accepted solutions with experimental designs to replicate or enhance classifier performance, ultimately identifying optimized approaches for clinical imbalanced time series classification.

II. Problem Statement

Since the main topic of this paper handles the issues present with an imbalanced clinical dataset, we must first understand Why might this rare chance be a problem in the medical field? Often you might see a disease diagnosis model, with healthy patients making the majority class and those with the disease considered the minority class. If someone with the disease were to be misclassified in the majority class of healthy patients, they might not receive the treatment needed to help them in time. As you can see, the problem with imbalanced data can affect the lives of people every day. Ideally, all datasets should have an equal distribution of targets amongst the total number of classes to avoid bias and encourage fairness. Before we jump directly into our study analyzing how specific solutions can be used in tandem to solve this, let us first take a look back at previous researchers' attempts to solve the imbalance issue.

III. Previous Works

To begin with, researchers Pei-Yuan Zhou and Andrew Wong propose in their paper[1] a method to discover patterns among traits that can be used to predict/classify clinical data despite the existence of imbalance meant. Their newly developed method is Pattern Discovery and Disentanglement for Clinical Data Analysis, or "cPPD" for short (Zhou et al.). How cPPD works in comparison to other existing pattern recognition methods is rather than attempting to discover high-order patterns over the entire dataset of features or Attribute-Values (AV) as described in the paper, cPPD creates small clusters of AV pairs. While these small clusters may not provide a great number of pattern candidates compared to the amount that would be obtained across the entire dataset, they provide much more valuable information in the form of reduced variance and correlation between smaller sets of AV pairs.

The reduced variance is beneficial for large clinical datasets, as it avoids the possibility of patterns being ambiguous; in other words, there will be less of a chance that a training or testing sample can be classified as two or more classes. As for pattern correlation amongst AV pairs, using smaller clusters helps solve the imbalanced dataset issue. Regular pattern recognition approaches consider the entire dataset, as previously mentioned, where most classes will have an advantage over the minority in being classified. Smaller clusters avoid this averaging over the entire dataset, allowing minority classes to be more accurately categorized due to their relations amongst more minor AV pairs. As an added benefit to using cPPD, its output provides interpretable results, unlike the usual "black box" models of other Machine Learning algorithms (Zhou et al.).

One specific challenge that researchers Zhou and Wong encountered in their paper [1] was that it could come in different forms that may not provide an apparent relation between the features and output classes/predictions when dealing with clinical data. Thankfully, for our purposes, we will be handling solely time-series data of patients, but in the case that an obvious relation can't be made, one solution we have to circumvent this is by maintaining consistent data input amongst all patients. Ensuring that all patient data we come across has the same amount of data (features) available, it allows for us to determine patterns much more easily across consistent records as opposed to broken or missing pieces of crucial features.

A. Mary Sowjanya and Owk Mrudula[2] propose a new SMOTE to solve the imbalance problem in the health dataset. The study focuses on the Breast Cancer Image Dataset and Novel Coronavirus 2019 time-series dataset. Instead of using the classical SMOTE, they propose two versions of modified SMOTE: Distance-based SMOTE or D-SMOTE and Bi-phasic SMOTE or BP-SMOTE.

Distance-based SMOTE generates new synthetic samples by

considering the distance between the samples of the minority class. It generates new examples near the minority class, resulting in a more effective oversampling technique for imbalanced datasets. However, D-SMOTE could introduce additional noise in the data while creating new examples; therefore, the authors propose BP-SMOTE, an improvement over D-SMOTE. BP-SMOTE is a two-phase algorithm, wherein the first phase, classical SMOTE, is applied to duplicate the minority class, and in the second phase, to create the final dataset greedy algorithm is used. There is only one criterion to be a part of the final dataset; the instance should perform well. BP-SMOTE only considers the examples close to the decision boundary, which increases generalization and avoids overfitting. It gives the minority class a more precise boundary so that the model can focus only on those examples closer to the boundary.

In addition to SMOTE, to provide a further boost to the performance, the authors propose stacked ensembling. For this study, the authors have used three base learners: Decision Trees, Neural Network, and Naive Bayes, and one meta-learner: Stacked CNN for Breast Cancer and Stacked RNN for Covid-19 Dataset. The stacked CNN model is divided into two sub-models, the first of which runs for 1530 iterations and the second for 756. The results of both sub-models are combined using logistic regression to produce the final output after they have been trained. For the stacked RNN model, multiple vanilla RNNs are stacked together to form the model.

To compare the performance of the SMOTE algorithms, the authors apply four models: Logistic Regression, Decision Trees, Boosting, and Random Forest on each of pre-processed data, and BP-SMOTE outperforms all of them with an accuracy of 92% on Random Forest. The study also highlights the performance of the base model and meta-learner, where the meta-learner (98%) outperforms the highest base model accuracy (94%) by 4% and the lowest base model accuracy (69%) by 27%. Finally, on the Covid-19 dataset, stacked RNN and stacked CNN achieve an accuracy of 96%-97%, whereas the base model lies in the range of 83%-87%

One drawback that the study might face is the vanishing gradient in RNN. Stacking multiple RNNs could lead to a vanishing gradient and poor dataset results. This problem could be overcome by using LSTMs or advanced RNN models like Echo State Networks. Second, the study focuses on accuracy as a metric that could be better in healthcare and on imbalanced datasets. Instead of accuracy, recall, precision, and F1-score should be prioritized.

Qingsong Wen et al[3] comprehensively review the current state-of-the-art methods for time series data augmentation in deep learning. The paper introduces the concept of data augmentation and its importance in improving the accuracy and generalization of deep learning models, especially in the context of time series data.

The paper then delves into various data augmentation techniques such as interpolation, jittering, scaling, time warping, and frequency warping. The paper also discusses some combination techniques, where multiple data augmentation techniques are used to augment the time series data.

The paper also discusses the implementation of these techniques in various deep learning models:

1) Convolutional Neural Networks (CNNs): Jittering and Scaling can be easily applied to the input data before feeding it into the CNN. Additionally, frequency warping can be used to augment the input data in the frequency domain. The research shows that time-warping is less commonly used in CNNs due to its high computational cost.

2) Recurrent Neural Networks (RNNs): Jittering and Scaling can be applied to the input data, and time warping can introduce variability in the sequence. The research shows that time warping can be especially effective in RNNs, as it can help models learn to be more robust to variable-length sequences.

3) Autoencoders: Data augmentation techniques such as jittering and scaling can be applied to the input data, and time warping can introduce variability in the input sequence. Additionally, interpolation can generate synthetic data for training the autoencoder.

The authors note that the effectiveness of each data augmentation technique can depend on the specific deep learning model and the task at hand and suggest carefully evaluating the performance of each technique. Additionally, the authors highlight the importance of combining multiple data augmentation techniques to further enhance the performance of deep learning models on time series data. The research does not cater to combining the deep learning model with weighting for imbalanced datasets. The paper does not provide an efficient way of dealing with imbalanced time series datasets.

Researchers Lijue Liu et al.[4] propose a combined learning solution that integrates data-level and algorithmic solutions with ensemble learning. The dataset used by the study has a significant high-class imbalance ratio and focuses on binary classification problems. The study selected relevant features based on two significance tests and logistic regression results. A set of features Fs was selected using a t-test for continuous features and a Chi-square test for categorical features, with features having p¡0.05 being statistically significant. A Feature set was created by finding the union of Fs and features selected by logistic regression Fl.

The significance of the features in the Feature set was ensured by ranking the features according to importance using Random Forest and recursive feature selection. Random Forest ranked all the features in the dataset based on the Gini coefficient. The study also used the Support Vector Machine model for recursive feature elimination by using the order of elimination to create feature rankings. The optimacy of the selected feature set created by statistical tests and logistic regression was determined based on the rankings obtained.

The SVM model was selected as the base learner for the ensemble model based on its suitable attributes, like the ability to handle high-dimension data and the ability of its cost function to set higher penalties to the class with higher

weights. Thus, SVM would ensure low error tolerance for the positive patient (minority) class. The study compares SVM models with different weight values to find the best positive and negative class weights. The authors favor higher sensitivity over specificity while performing cost-sensitive analysis because while handling clinical datasets, the detection of most patients having the disease is of more significance than the misclassification of a few non-patients.

The proposed ensemble algorithm consists of T weak learners where T is calculated as the ratio of majority class samples to minority class samples. A weak learner is trained by initializing SVM with the weights obtained by cost-sensitive analysis and is trained over a balanced dataset obtained by combining the undersample majority class with the minority class. The meta-learner is trained by combining all weak learners using bagging.

IV. Proposed Solutions

As previously mentioned, there have been attempts to combat the effects of an imbalanced dataset on classification models. Some that we've noted before have been in the form of SMOTE and its variants[2], a sampling technique that when combined with deep learning ensemble methods has shown to improve the accuracy despite the presence of imbalanced data. Others come in the form of using a specific model architecture like the Echo State Network [4], a variant of the RNN model that uses a reservoir with fixed hidden layer weights, and only updates the weights of the output layers to enforce the desired outcome. This procedure of the Echo State Network has also proven to provide greater accuracy with imbalanced datasets, but has yet to be lost to time in favor of more developed modern models today. Transformers architecture uses attention mechanisms to capture the sequence's dependencies between all time steps. Because of attention mechanisms, transformers give more weight to the most crucial time steps in the sequence while ignoring the irrelevant ones. This makes it easier for the model to capture long-term dependencies between distant time steps which is critical when working with time series data.

For this paper, an empirical study is performed to investigate the potential of combining universally accepted solutions with experimental designs to replicate or enhance classifier performance, ultimately identifying optimized approaches for clinical imbalanced time series classification. The study aims to discuss results based on experiments that use a combination of SMOTE and its variant BorderLine SMOTE, Simple RNN, Modified RNN consisting of Echo State Cell, Transformer, and Feature Ranking using Random Forest.

While not implemented in this report, we initially had a novel approach that would give class weights to individual observations of time-series data across all patients but normalized them in accordance to how many days of observation each patient. The name for this proposed approach was called Weights Relative to Data (WRD). We had hoped that by taking a look into these possible imbalances of observational data, we would observe whether or not the amount of time-series data each of our patients had recorded could be a factor contributing to the overall imbalance problem. However, we deemed that WRD did not follow suit with our empirical study of the model and synthetic data combinations that we intended to observe, and as a result, felt it best to drop our WRD approach.

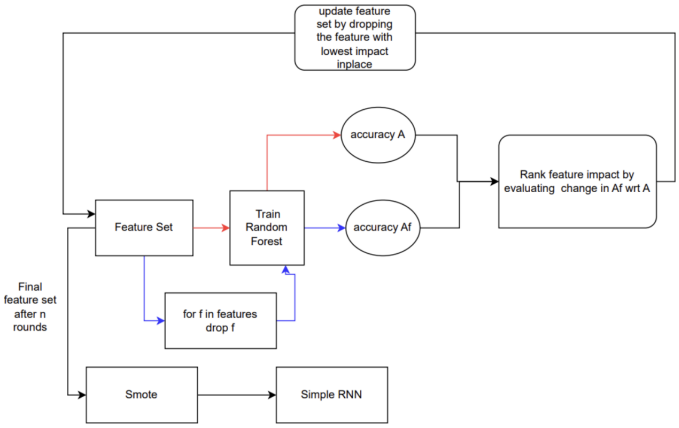

Fig. 1. Random Forest Feature Ranking Flowchart

V. Experiment Settings

A. Dataset

The Tappy Keystroke Dataset [5] was put together by PhysioNet and included the keystroke records of 200 individuals worldwide, both with and without Parkinson's. A program called Tappy, was designed to log each user's interaction with their keyboards to detect changes in the routines and examine the early-stage impacts of Parkinson's disease. There are two folders in the dataset:

- User Data: This section contains text files with the personal information of individuals, including their gender, age at diagnosis, and whether they have Parkinson's disease.
- Keystroke Data: The text files in this subdirectory contain information about keystrokes for each individual. Each file includes information about the user's key, whether left- or right-handed, the direction to the previous key, their latency time (time between the previous key and current key), and their flight time. (time between the release of the previous key and the press of the current key)

Table I details the data imbalance of the Tappy dataset with 74.45% belonging to subjects having Parkinson's disease and 25.55% of the subjects belonging to the healthy class. This highlights the stark imbalance present in the dataset.

TABLE I
IMBALANCE IN THE DATASET

Parkinson's?	Amount	Percentage of Dataset
True	169	74.45%
False	58	25.55%

B. Pre-Processing

The Tappy Keystroke Dataset offers a vast amount of time-series samples for our models to utilize, with roughly 9 million available. However, given the size of this dataset, we lacked the resources to reliably train each of our proposed solution models on the entire dataset and instead opted to use a subset of the dataset, with 100,000 samples of observational data across all patients. We ensured that this smaller batch of data maintained the same imbalance ratio presented earlier, and furthermore considered the possibility of timestep data points having no correlation with their previous neighbors. Thankfully, no loss of information was seen, as Figure 2 further proves through Lag Observation across all feature values of subsequent timestep datapoints are shown to be fairly close to each other, advising us that datapoints are within a reasonable range of one another.

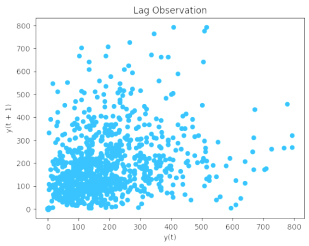

Fig. 2. Lag Observation of Reduced Dataset Across Timesteps

Furthermore, during preprocessing, specific columns were removed from the user part of the dataset because they indicated whether the user had Parkinson's disease. The UPDRS is a scale that is used to assess the severity of Parkinson's disease. The impact of Parkinson's disease on the patient's life is defined. Levodopa and Maob indicate whether the user is taking the medications to treat Parkinson's disease. The user data was combined with the Tappy Data, and the UserKey, Timestamp, and Gender fields were removed from the dataset. To generate the user's current age, feature engineering was used. The final dataset had seventeen features which were oversampled using the following methods to handle class imbalance.

- Synthetic Minority Oversampling Technique, or SMOTE, is an oversampling method that balances imbalanced data sets. SMOTE creates new minority-class examples from scratch rather than duplicating existing ones. First, a random point from the minority class is selected, and the point's k-nearest neighbors are determined. A new synthetic point is synthesized between two examples from the nearest neighbors.
- BorderLine SMOTE, also known as SVM SMOTE, was used as the data preprocessing technique for another experiment. The aim was to oversample the minority class of participants not having Parkinson's. BorderLine SMOTE also generates synthetic data similar to SMOTE; however, data is generated only along the borderline area. The borderline area is identified using SVM algorithms trained on the data. Thus, compared to SMOTE, Border-Line SMOTE generates the data without crowding in the training set.
- Feature Ranking using Random Forest, was used to identify seven features that have the most negligible impact on model accuracy, which are discarded. The remaining topmost features are used as input to oversampling techniques like SMOTE. Thus the feature set was reduced from 17 features to 10 features after applying Feature Ranking. By prioritizing the most critical features and discarding the least important ones, Random Forest can effectively help in dimensionality reduction without causing information loss. After feature selection, SMOTE is applied to the selected features, which can address the issue of imbalanced datasets by generating synthetic samples of the minority class. Fig. 1. describes the Feature Ranking algorithm used to identify the top 10 features. Initially, the Feature Set consisted of all 17 features. Accuracy A is obtained by training a Random Forest model on all features in the feature set. A is compared to the accuracy Af obtained by dropping a feature f from the feature set. The feature with the lowest impact is determined by evaluating the change in Af with respect to A and is dropped from the feature set. This process is continued until the top 10 features are obtained.

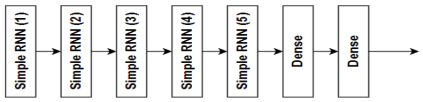

Fig. 3. Model 1: Simple RNN

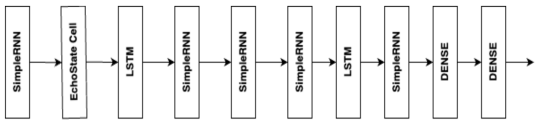

Fig. 4. Model 2: Simple RNN with EchoState and LSTM

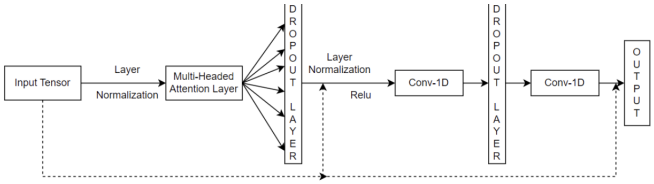

Fig. 5. Model 3: Transformer

C. Models

For the midterm study, we have used two models to develop a baseline for future experiments: the basic RNN and state of the art cells like Echo state and LSTM with simple RNN.

- SimpleRNN - The SimpleRNN model has seven layers: five RNNs and two dense layers. Each RNN and dense layer (except the final layer) has a Relu activation, while the final dense layer has a Sigmoid activation. The model's loss function is binary cross-entropy, and Adam is its optimizer. Fig 3. describes the architecture of the SimpleRNN model used which consist of five SimpleRNN layers and two Dense Layers.
- Modified SimpleRNN - The modified SimpleRNN is similar to SimpleRNN but has two additional LSTM layers and one echo state cell. All layers except the final dense layer have a Relu activation, while the final dense layer has a Sigmoid activation. The model's loss function is binary cross-entropy, and Adam is its optimizer. Fig 4. describes the architecture of Modified SimpleRNN which consist of SimpleRNN, EchoState Cell and LSTM layers.
- Transformer - When working with time series data, capturing long-term dependencies between distant time steps is critical. Unlike standard machine learning models, encoder blocks in transformers use attention mechanisms to construct contextual relationships between time steps. This allows the transformer to capture such long-term dependencies within the data. Transformers can weigh and combine information from numerous time steps by using attention mechanisms, giving more weight to those most relevant to producing correct predictions. Following the encoding process, a Global Average Pooling layer is used to reduce tensor dimensionality, which aids in the avoidance of overfitting and improves computational efficiency. Finally, Dense layers with mlp units are used to learn complicated patterns in time series data, allowing the model to generate accurate predictions based on the encoded data. Fig 5. details the structure of the Transformer model consisting of a Multi-Headed Attention Layer, Global Average Pooling layer, and Dense layers with mlp units.

D. Hyperparameters

Table II lists the Hyperparameters used to train the Simple RNN, Modified RNN and Transformer.

TABLE II
HYPERPARAMATERS FOR THE MODELS

Hyperparameter	Values
Batch Size	64
Learning Rate	1e-3
Epochs	15
Epochs for transformer	200
Optimizer	Adam

E. Experiments performed

As part of an empirical study, various experiments were conducted by combining universally accepted solutions. These experiments involved testing different combinations of solutions to determine their effectiveness in solving the imbalance problem in clinical datasets.The results of each of the below mention experiments are detailed in Section V.

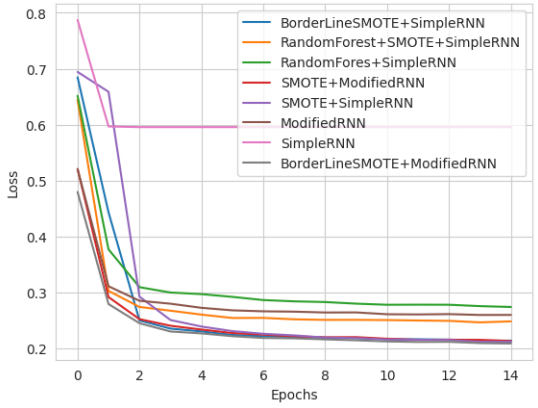

Fig. 6. Training Loss

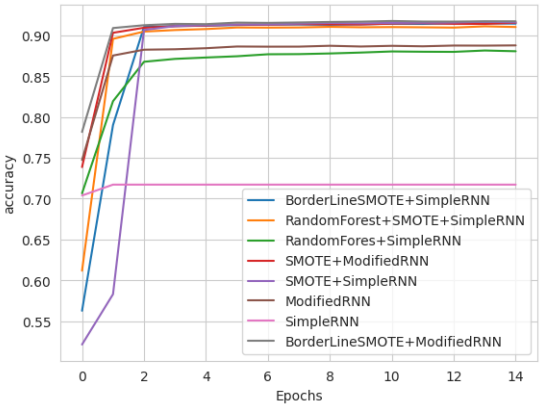

Fig. 7. Training Accuracy

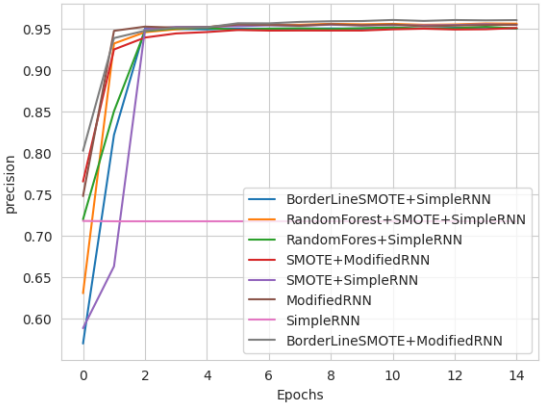

Fig. 8. Training Precision

- Simple RNN without SMOTE
- Simple RNN with SMOTE
- Simple RNN with BorderLine SMOTE
- Modified RNN with SMOTE
- Feature Ranking using Random Forest with SMOTE

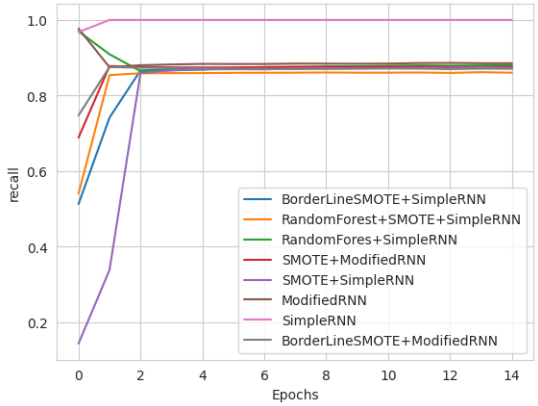

Fig. 9. Training Recall

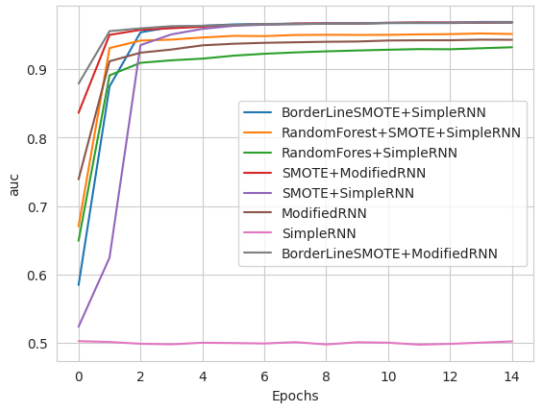

Fig. 10. Training AUC

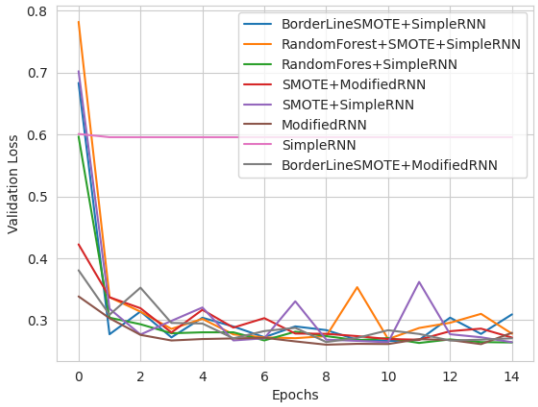

Fig. 11. Validation Loss

- Transformer with SMOTE
- Transformer without SMOTE
- Modified RNN without SMOTE
- Modified RNN with BorderLine SMOTE
- Feature Ranking using Random Forest without SMOTE

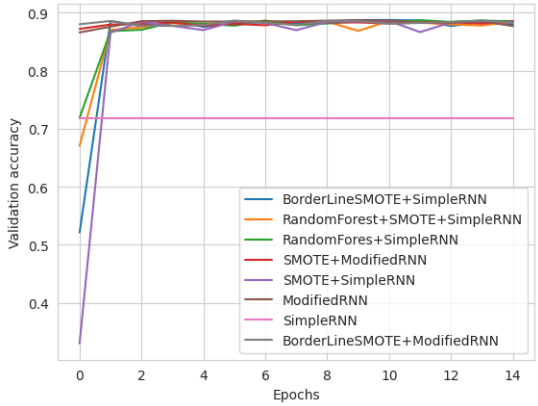

Fig. 12. Validation Accuracy

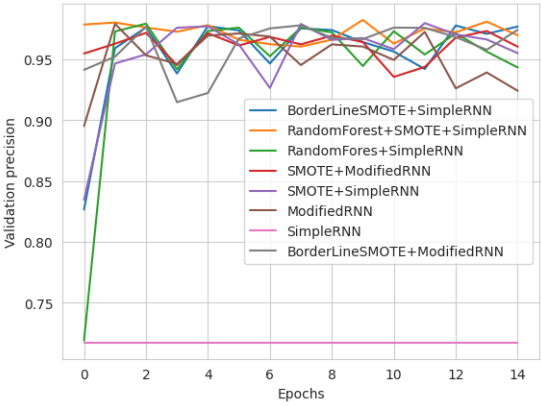

Fig. 13. Validation Precision

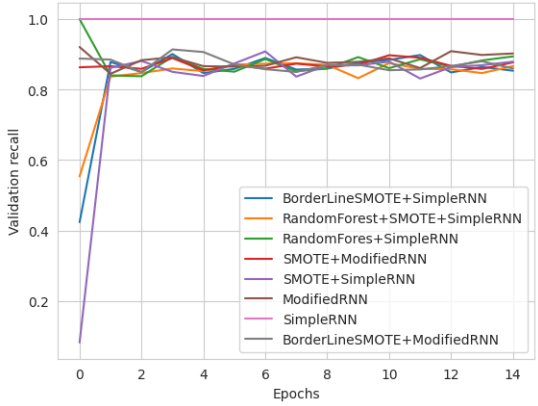

Fig. 14. Validation Recall

F. Evaluation Protocol

Since the project handles clinical datasets, it is crucial to identify patients with Parkinson's accurately compared to misclassifying a few samples not having Parkinson's. Hence, Recall and AUC are considered important metrics for model evaluation. Recall allows for us to observe the total amount of correctly classified patients with Parkinson's according

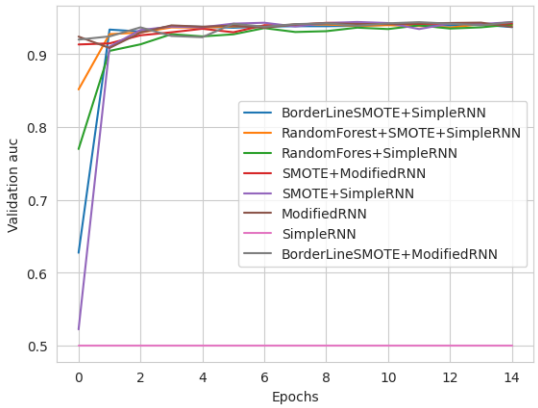

Fig. 15. Validation AUC

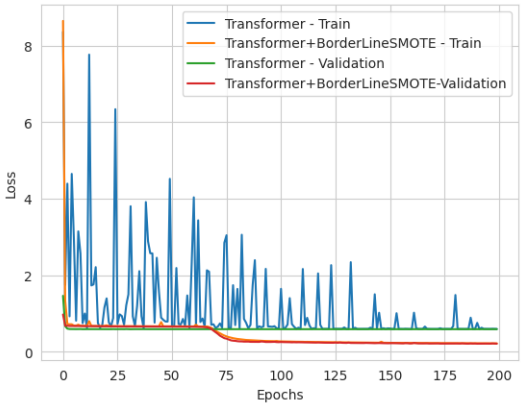

Fig. 16. Transformer Loss

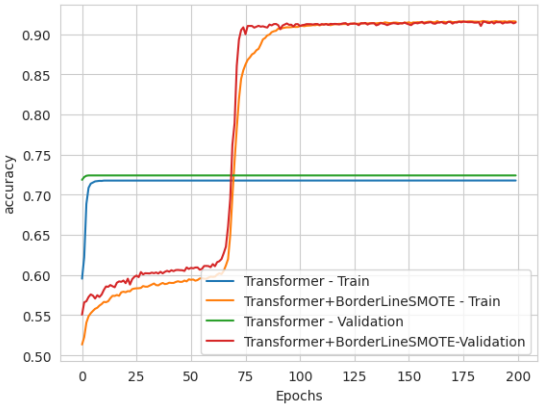

Fig. 17. Transformer Accuracy

to their data, while AUC score measures the probability of classes being classified amongst one another. In addition, the usual metrics of models that include Accuracy, Precision, and Loss are just as important to us as Recall and AUC. We wish to compare each of our implementations across every viable metric to get a concrete showcase of how ensembling of imbalance solutions can possibly improve classification performance.

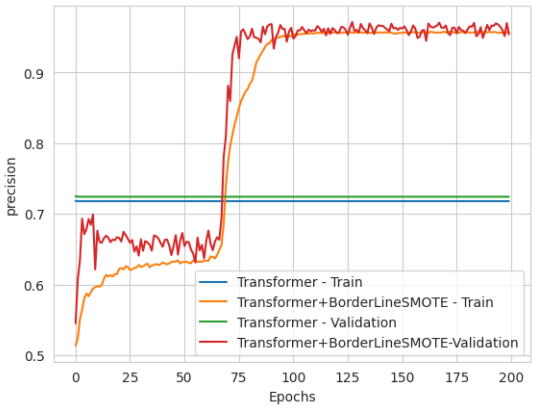

Fig. 18. Transformer Precision

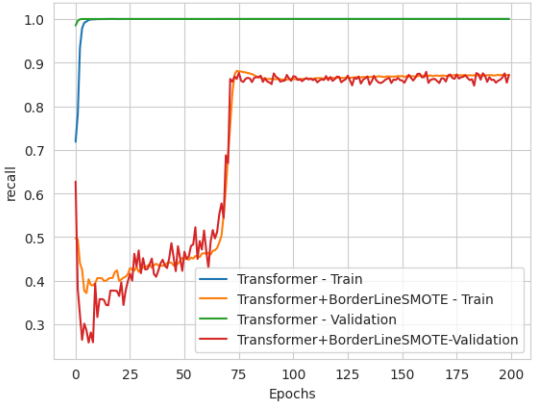

Fig. 19. Transformer Recall

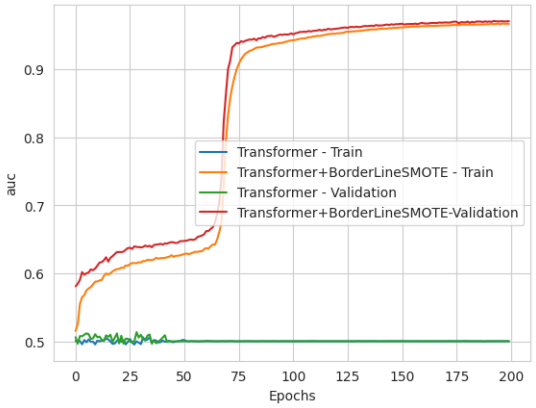

Fig. 20. Transformer AUC

VI. Results

This section details the results obtained from the different combinations of solutions tested mentioned in Section E under Experiments proposed. The study aims to provide insights into the most effective solutions for addressing the research problem by presenting these results.

TABLE III
TEST DATASET RESULTS

Method	Loss	Accuracy	Recall	Precision	AUC
1. SimpleRNN	0.59	0.71	1	0.71	0.49
2. SMOTE+SimpleRNN	0.26	0.88	0.87	0.95	0.94
3. BorderLineSMOTE+SimpleRNN	0.30	0.88	0.85	0.97	0.93
4. ModifiedRNN	0.28	0.87	0.90	0.92	0.93
5. SMOTE+ModifiedRNN	0.26	0.88	0.88	0.95	0.94
6. BorderLineSMOTE+ModifiedRNN	0.26	0.88	0.86	0.97	0.94
7. RandomForest+SimpleRNN	0.26	0.88	0.89	0.94	0.93
8. RandomForest+SMOTE+SimpleRNN	0.28	0.88	0.86	0.96	0.93
9. Transformer	0.86	0.71	0.99	0.71	0.50
10. Transformer+BorderLineSMOTE	0.25	0.88	0.87	0.96	0.95

Fig 6. and Fig. 11 show the plot Training and Validation Loss of 8 experiments, respectively. It can be observed that Simple RNN obtained the worst loss without SMOTE for both training and validation sets. On the other hand, BorderLine SMOTE with SimpleRNN, SMOTE with ModifiedRNN, SMOTE with SimpleRNN, and BorderLine SMOTE with ModifiedRNN give comparable results for Loss for the training set. All seven models, apart from Simple RNN without SMOTE, show comparable results for Loss for the validation set.

Fig 7. and Fig. 12 show the plot for Training and Validation Accuracy for eight experiments, respectively. It can be observed that Simple RNN without SMOTE shows the worst performance with an accuracy of less than 0.75% for both training and validation sets. Feature Ranking with Random forest+SMOTE+SimpleRNN, SMOTE with ModifiedRNN, and BorderLine SMOTE with Modified RNN achieve comparable accuracies of around 0.9 for both training and validation sets.

Fig 8. and Fig.13 show the plot for Training and Validation Precision for eight experiments, respectively. It can be observed that Simple RNN without SMOTE shows the worst performance with less than 75% precision for both training and validation sets. All other models show comparable performance with respect to precision for both training and validation sets.

Fig 9. and Fig 14. shows the plot of Training and validation recall for all eight combinations. Recall is an essential metric in imbalanced datasets, but it cannot be considered alone as a metric in some cases, as it only provides part of the picture.SimpleRNN has a precision (Fig. 10) and AUC (Fig. 11) of 0.72 and 0.5, respectively, and a recall of almost 1. This illustrates that the model is underperforming despite the high recall and consistently predicts the majority class. All the other seven combinations have a comparable recall of 0.85-0.87.

Fig 10. and Fig 15. highlight the AUC of the combinations, which shows the entire picture of false positive and true positive rates. SimpleRNN, the worst-performing model, achieves a maximum AUC of around 50% in training and validation. In contrast, combinations like SMOTE/BorderLine SMOTE with SimpleRNN and Modified RNN achieve the highest recall in both training and validation.

Fig 16. displays the Transformer's loss performance on the training and validation datasets. The model is significantly more stable with BorderLine SMOTE included than before. Additionally, by including BorderLine SMOTE, the model performs better on both datasets.

Fig 17, 18, 19, and 20. highlight the transformer model's accuracy, precision, recall, and AUC on both datasets. Without Borderline SMOTE, the model achieves a maximum accuracy, precision, and AUC of 70%,70%, and 50%, respectively, on both datasets. In contrast, with Borderline SMOTE, the model reaches an improved accuracy of around 91% and handles the imbalance by achieving an improved AUC of 95-97%. However, the model achieves a recall of around 87%, comparable to other simple models.

From Table III we can make the following conclusions. Combining different preprocessing techniques and models leads to better performance than baseline models. For example, SimpleRNN and Transformer without SMOTE and Borderline SMOTE perform worse compared to the addition of oversampling techniques, predicting only the majority class. This also indicates that using oversampling techniques, such as SMOTE and its variants, can tackle the imbalance issue, ensuring equal weightage to each class. Even though reducing less important features simplifies the model, leading to better performance, in this case, random forest feature selection contributed little towards improvement. However, it achieved comparable results with lower dimensions of input data. If we compare the results of SimpleRNN and ModifiedRNN without SMOTE, the modified outperforms SimpleRNN, achieving almost comparable to the other experiments. This shows that the EchoState cell efficiently handles data imbalance without oversampling techniques. The performance of Transformers with BorderLine SMOTE on the test data achieves the highest AUC and high values for Precision, Recall, and Accuracy, showing that Transformer can capture long-term dependencies in time series data efficiently.

VII. FUTURE PLANS

Having observed how each of our combinations of models and synthetic data generators SMOTE and BorderlineSMOTE all work in cohesion, we feel that more can be done in this field to further reduce the imbalance issue by using ensembling methods. We could not cover numerous other solutions in the scope of our report here, like our original WRD approach that could bring about valuable data. Still, we anticipate that by incorporating other existing methods created for this specific problem, we can highlight the importance of engineering various models together. For example, one additional approach to synthetic data apart from SMOTE that we considered was the use of DoppelGAN (DGAN), which would make use of a generative adversarial network specializing in time-series data to provide additional synthetic data. Due to time constraints, it has yet to be seen whether this method of generating data could perform better than SMOTE or even if a combination of synthetic data generators could increase performance. Still, it serves as another avenue for other researchers to observe.

With the work done here, we hope to highlight the contributions that previous solutions can still provide to newly created ones in the future, showcasing that an ensemble of approaches can at times perform better than when used individually.

REFERENCES

[1] Zhou, PY., Wong, A.K.C. Explanation and prediction of clinical data with imbalanced class distribution based on pattern discovery and disentanglement. BMC Med Inform Decis Mak 21, 16 (2021). https://doi.org/10.1186/s12911-020-01356-y

[2] Sowjanya, A.M., Mrudula, O. Effective treatment of imbalanced datasets in health care using modified SMOTE coupled with stacked deep learning algorithms. Appl Nanosci 13, 1829–1840 (2023). https://doi.org/10.1007/s13204-021-02063-4

[3] Wen, Qingsong, et al. "Time Series Data Augmentation for Deep Learning: A Survey." Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 2021, https://doi.org/10.24963/ijcai.2021/631.

[4] Liu, L., Wu, X., Li, S. et al. Solving the class imbalance problem using ensemble algorithm: application of screening for aortic dissection. BMC Med Inform Decis Mak 22, 82 (2022). https://doi.org/10.1186/s12911-022-01821-w

[5] "Tappy Keystroke Data v1.0.0." Home Page, https://doi.org/10.13026/C2K08D. Accessed 3 Apr. 2023.

VIII. CONTRIBUTION

- Sergio Ramirez - Assisted in creating Project Proposal slides along with other teammates. Furthermore, assisted with the writeup of the Project Survey Report Report by finding research article Explanation and prediction of clinical data with imbalanced class distribution based on pattern discovery and disentanglement, and analyzing how well the applications of the research paper's developments can aid in our overarching project. Additionally, assisted with the writeup of the Midterm Project Report, and analyzing how imbalances present in the availability of data factor more into the overall imbalance issue. Moreover, proposed the idea for the Weight Relative to Data (WRD) class weight solution that unfortunately was dropped for other approaches. Developed a time series generator model for additional synthetic data, however there were complications in getting the newly generated data to work with our models and was dropped in favor of SMOTE. Helped in observing that the Lag Observation in the reduced dataset still correlated for usage in our models. Finally, helped present the project along with Kshitij for our presentation day.
- Vansh Jain - Explored the dataset and its properties and assisted in finalizing the dataset. Listed the key features after performing analysis and understanding their properties and features. Assisted with feature extraction, selection, preparation, and engineering for the dataset. Applied SMOTE on it to balance the dataset and built two models: Simple RNN and modified RNN. Understood the individual properties of each model and technique and, based on those, combined different oversampling techniques with simpleRNN and modifiedRNN. Constructed the experiments: SimpleRNN without smote, Simplernn with SMOTE, ModifiedRNN, ModifiedRNN with SMOTE, and BorderLineSMOTE and summarized their results in a table. Assisted Sharvari in Random Forest feature selection and training. Also, performed data visualization to aid different experiment comparisons. Collaborated with other teammates to create the midterm project report and final project presentation and report. Trained the models on the dataset and evaluated their performance using the metrics mentioned in the report. Performed a detailed survey about utilizing SMOTE, one of our proposed approaches to solve the imbalance issue in clinical datasets. Contributed to the survey report by understanding the proposed variations of SMOTE for handling class imbalance by authors A. Mary Sowjanya Owk Mrudula in "Effective treatment of imbalanced datasets in health care using modified SMOTE coupled with stacked deep learning algorithms." Highlighted the study's limitations and how our proposed methodology could overcome those.
- Sharvari Kalgutkar - Researched available clinical datasets and assisted in finalizing the Parkinson's dataset for the project. Understood the various features of the Parkinson's dataset and assisted in data preprocessing, feature engineering, and feature selection. Conducted Approach 2 of the baseline experiments by performing BorderLine Smote and modeling a Simple RNN on the oversampled dataset. Performed model evaluation of Approach 2 based on the finalized evaluation metrics. Performed experiments for Random Forest Feature Ranking with and without SMOTE using SimpleRNN and Transformer without SMOTE and assisted in understanding the experimental results of various combinations and determining the pros and cons of each approach. Assisted in drafting the Midterm Project Report and Final Report and collaborated with other teammates to create Project Proposal slides.Conducted a literature survey to explore a different method for solving class imbalance problems in clinical datasets. Contributed to the project survey report by understanding the integrated approach presented by authors Liu, L., Wu, X., Li, S. et al. in their study Solving the class imbalance problem using ensemble algorithm: application of screening for aortic dissection. Identified the limitation and future scope of the proposed integrated solution that targets data and algorithm along with ensembling.
- Kshitij Parab - Surveyed various clinical imbalanced datasets and contributed in the selection of the most appropriate dataset for the project. Performed SMOTE on the Tappy Keystroke Dataset and explored other variations of SMOTE to achieve better results. Contributed in data preprocessing and feature extraction. Implemented an initial trial RNN model that was essential in understanding the experiment settings. Implemented the transformer approach using a transformer encoder to capture long-term dependencies and learn contextual relationships between time steps. Further, I performed model evaluation using the following metrics: accuracy, precision, recall, AUC, and loss and analyzed the evaluation metrics. Presented the final project and assisted in drafting the Final Report. Aided in drafting the Midterm Project Report and covering 'Experimental Settings' and 'Results.' Presented the Project Pitch and represented the project team in giving the research project overview.

Aided in creating the Project Proposal slides that describe the objective and impact of the research. Additionally, I conducted a literature survey on ‘Time Series Data Augmentation for Deep Learning: A Survey’ by Qingsong Wen et al. and contributed to drafting the Project Survey report.