Deep Learning Models for Imbalanced Time-Series Clinical Data

Trojan Transformers

Vansh Rajesh, Sharvari Kalgutkar, Kshitij Harish Parab, Sergio Ramirez

Overview

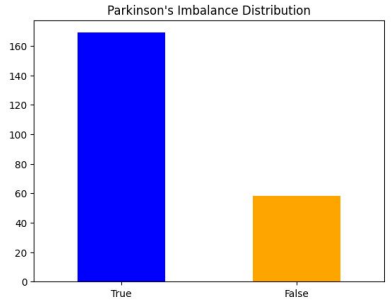
- What is an imbalanced dataset?
  - A dataset containing an uneven distribution of samples among all classes
  - Majority class having most samples
  - Minority class having fewer samples
  - Common plague introducing bias in deep learning models

- Why is this bad for clinical data?
  - Lives are at stake, misclassification (misdiagnosis) can have dire consequences

***Goal*:** An empirical study 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.

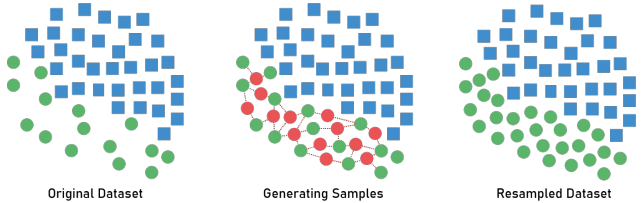
Previous Work and Dataset

- Tappy Keystroke Dataset:
  - Imbalance Ratio
    - True (Parkinson's) - 169 (74.45%)
    - False (No Parkinson's) - 58 (25.55%)
  - Features: Hold Time, Flight Time, Latency, Hand, Direction, Birth Year, Parkinsons, Tremors, Age



- SMOTE (Sowjanya AM, Mrudula O. 2022)
  - Oversampling Method generating synthetic data for model use
  - Generated using distance between sample points in linear space

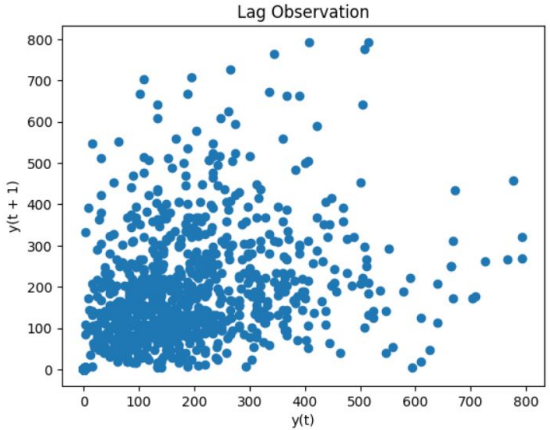
Synthetic Minority Oversampling Technique



- Echo State RNN Network
  - RNN variant specializing in time series tasks, synthetic data
  - Non-linear, only last layer reservoir weights updated

Preprocessing

- Entire Tappy Dataset too large to handle, utilize smaller batch for training/testing/validation
  - Training - 60K
  - Validation- 15K
  - Testing - 25K
  - Despite reduction, correlation still preserved



- SMOTE variants applied to create synthetic data
  - More data = Reduce imbalance
  - **SMOTE**:synthesizes data using k-nearest neighbors.
  - **BorderLine SMOTE** : data generated along borderline area using SVM algorithms.
    - Unlike SMOTE, it generates data without crowding the training set.

SimpleRNN and Modified SimpleRNN

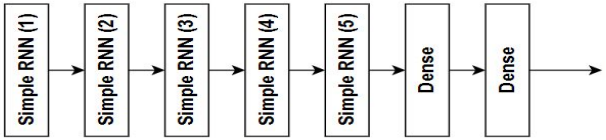


Fig 1. SimpleRNN

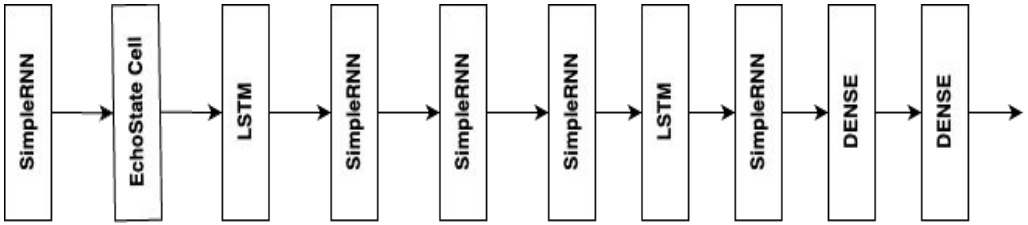


Fig 2. Modified RNN

- Modified RNN includes RNN, EchoState, and LSTM cells.
- Both these models were trained for 15 epochs with the following hyperparameters:
  Batch Size -64
  Optimizer - Adam
  Learning Rate - 0.001

EXPERIMENTS

- Baseline experiments:
  Simplified RNN - with SMOTE, without SMOTE, with BorderLine SMOTE
- Modified RNN - with SMOTE

Transformer

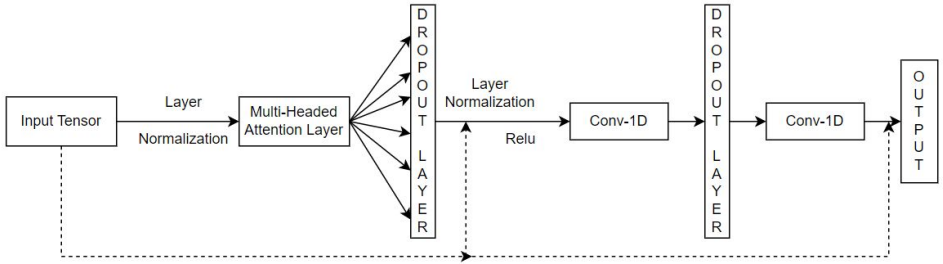


Fig 3. Transformer Encoder

- The transformer encoder blocks use attention to learn contextual relationships between time steps, capturing long-term dependencies
- The Global Average Pooling layer reduces tensor dimensionality, and the Dense layers with mlp_units learn complex patterns and relationships in the time series data.

Random Forest Feature Ranking for feature selection

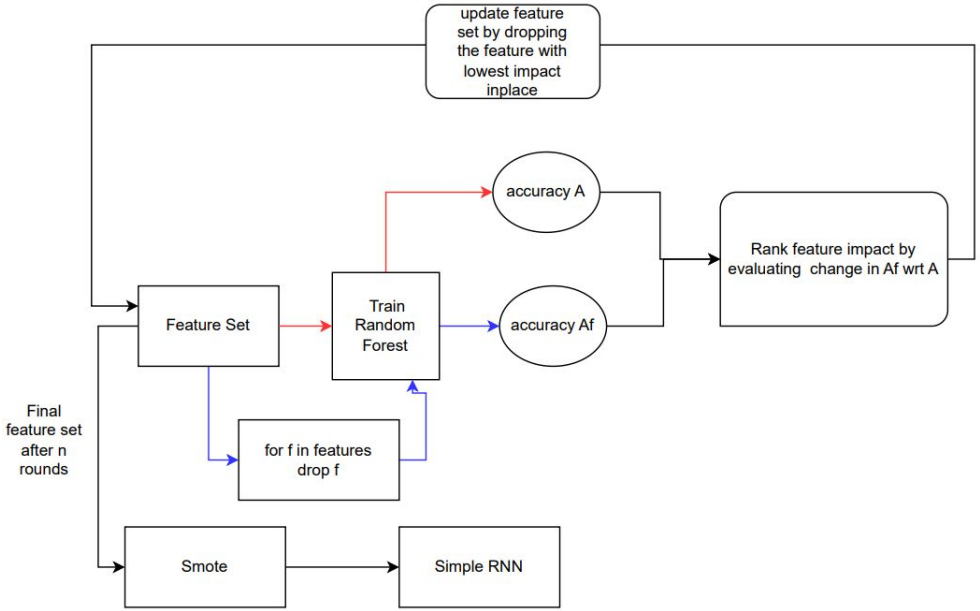


Fig 4. Random Forest Feature Selection

- Reduced feature set: top 10 features out of 17 selected.
- Oversampled with SMOTE and predicted using a Simple RNN model.
- Same hyperparameters and epochs as before.

Results and Comparisons

Method	Loss			Accuracy			Recall			Precision		
	Train	Val	Test	Train	Val	Test	Train	Val	Test	Train	Val	Test
1. RNN without SMOTE	0.59	0.59	0.59	0.71	0.71	0.718	1	1	1	0.71	0.71	0.71
2. RNN with SMOTE	0.22	0.23	0.27	0.91	0.91	0.88	0.87	0.90	0.90	0.95	0.90	0.92
3. RNN with Borderline SMOTE	0.21	0.28	0.28	0.91	0.88	0.88	0.87	0.86	0.85	0.95	0.97	0.97
4. Modified RNN with SMOTE	0.21	0.26	0.26	0.92	0.88	0.89	0.87	0.88	0.88	0.95	0.95	0.95
5. Random Forest with SMOTE AND RNN	0.26	0.28	0.27	0.91	0.88	0.884	0.85	0.88	0.88	0.95	0.95	0.95
6. Transformer with SMOTE	0.21	0.20	0.25	0.91	0.91	0.88	0.86	0.86	0.88	0.95	0.96	0.94

Fig 5. Results and Comparison

Future Work and Research

- New Approaches to creating Synthetic Data
  - DoppelGAN (DGAN) - A Generative Adversarial Network suited specifically for Time-series data that can be used to make promising predictions for additional data.

- Extension of Evaluation Metrics
  - ROC Curve
  - AUC Score

- Most research papers focus on new implementations or algorithms. Rarely do we see an analysis on the engineering aspect of how well they all work in tandem with previous ones.

- Here in this project we provide an analysis on the “ensembling” of numerous solutions to highlight the contribution each can provide now and in the future.

Thank You!