Multi Class Classification Of Fetal Health

Gugulothu Ananth Naik
Electrical and Electronics Engineering
Indian Institute of Technology Patna
Patna, India
2201ee27_gugulothu@iitp.ac.in

Bhukya Prem Naik
Electrical and Electronics Engineering
Indian Institute of Technology Patna
Patna, India
2201ee22_bhukya@iitp.ac.in

***Abstract*—In this study we analyse the effectiveness and accuracy of models by using a few important metrics such as precision, recall, F1-score, and accuracy. These metrics offer insights on the models and help compare the performance of the models in comparison to each other and also the improvements made during the study. For this research, we performed SVM, Random Forest, and XGBoost on a non-time series data annotation file to classify fetal health categories(Normal, Suspect, Pathological). Additionally, we worked with another dataset containing 552 CSV files with time series data on fetal heart rate and uterine contractions along with an annotation file for the 552 samples, applying Time Series Transformer and LSTM models to analyze fetal heart rate and uterine contractions over time. This study allows us to compare the effectiveness of both the models on time series data, helping determine the effectiveness of machine learning models for static data, and deep learning models on sequential data.**

***Index Terms*—SVM (support vector machine), XGBoost (extreme gradient boosting), Random forest (RF), long-short-term memory (LSTM), time series transformer (TST),Gausian Process Regression(GPR), K-Nearest Neighbour(KNN), SHAP (shapley additive explanations), SMOTE (synthetic minority oversampling technique).**

I. Introduction

CTG stands for Cardiotocography, a monitoring technique used during pregnancy and labor to assess fetal heart rate (FHR) and uterine contractions, helping detect signs of distress. It is also known as Electronic Fetal Monitoring (EFM) or Non-Stress Test (NST). The device used in cardiotocography is called a cardiotocograph, which involves placing two sensors on the abdomen of a pregnant woman. One sensor records fetal heart rate using ultrasound, while the other monitors uterine contractions by detecting tension in the abdominal wall.

CTG readings are analyzed based on several parameters, including baseline rate, variability, acceleration, and deceleration. The baseline rate represents the average fetal heart rate, typically ranging from 110 to 160 bpm. Variability refers to fluctuations in FHR from one beat to the next, with a normal range of 5–25 bpm. Acceleration is a temporary increase in FHR above the baseline, indicating fetal well-being (an increase of more than 15 bpm for over 15 seconds). In contrast, deceleration is a decrease in FHR exceeding 15 bpm for more than 15 seconds. CTG data is classified into three categories: Normal, Suspect, and Pathological, based on expert-defined criteria.

While no conclusive evidence suggests that monitoring high-risk pregnancies significantly benefits the mother or baby, research indicates that computerized CTG machines have led to fewer infant deaths compared to traditional CTG. AI-based fetal health prediction has gained prominence, focusing on the analysis of CTG data. Machine learning techniques such as SVM, XGBoost, Random Forest, and Deep learning techniques such as LSTM and Time Series Transformers have been employed to classify CTG signals into normal, suspicious, or pathological categories. Research efforts aim to automate CTG interpretation to enhance diagnostic accuracy and reduce human error.

Machine learning significantly improves CTG analysis by increasing accuracy, automating diagnosis, and offering predictive insights. This advancement makes real-time fetal monitoring more reliable, ultimately improving health outcomes. The CTG dataset is widely utilized in machine learning for fetal health classification and risk prediction based on fetal heart rate and uterine contraction signals.

The summary of the study:

- In this study we analyse the effectiveness and accuracy of models by using a few important metrics such as precision, recall, F1-score, and accuracy. These metrics offer insights on the models and help compare the performance of the models in comparison to each other and also the improvements made during the study.
- We perform SVM (Support Vector Machine), XGBoost (Extreme Gradient Boosting) and Random Forest (RF) on a non-time series data annotation file of both Dataset A and Dataset B to classify fetal health categories(Normal=1, Suspect=2, Pathological=3).
- Used SHAP (SHapley Additive exPlanations) with an XGBClassifier to interpret feature contributions and identify the most influential features in our annotation dataset resulting in classifying the fetal health condition.
- We considered Gausian Process Regression(GPR) and K-Nearest Neighbour(KNN) to solve the problem of missing signal data of FHR and UC that had been lost and used SMOTE(Synthetic Minority Over-sampling Technique) to deal with the class Imbalance in the dataset.
- Applied Deep Learning models such as Time seies Transformer (TST) and Long Short-term Memory (LSTM) on Raw, GPR and KNN interpolated time series 'FHR' and

'UC' signal data of Dataset B to classify them into Fetal health categories(Normal = 1, Pathological = 0) .

II. Cardiotocogrpahy Dataset for Fetal monitoring

A. Cardiotocography [1]

The cariotocgraphy dataset (consider as dataset A in this work) includes 2126 fetal cardiotocograms (CTGs) that were classified by expert obstetricians into fetal state condition as normal, suspect and pathological classes. We are not including the morphological pattern classes available in this work. We consider 80:20 training and testing data split with k-fold cross validation to prevent overfitting and ensure good model generalization.

B. CTU-CHB-Intrapartum-Cardiotocography Dataset [2]

This dataset (consider as dataset B in this work) consists of 552 cardiotocography (CTG) recordings from the Czech Technical University (CTU) in Prague and the University Hospital in Brno (UHB). It is classified into Normal and Pathological fetal health classes. We work with the annotation data and the time series data of the 552 recordings containing the FHR and UC signals in order to classify it into fetal state condition as normal, pathological classes.

III. Experiment Analysis Of Non-Time Series Data

The models we consider here are Support Vector Machine(SVM), Extreme Gradient Boosting (XGBoost), and Random Forest(RF) to evaluate Dataset A and Dataset B. Time Series Transformer and(TST) long-short-term memory (LSTM) on Dataset B.

A. Data Preprocessig for Dataset A:

The Annotation Data file with 2126 fetal cardiotographic samples with their Multi-class classification of dataset into Normal(1), Suspect(2), Pathological(3) classes is taken as our input file. It contains 21 features that are used to determine the class of fetal health of the sample. After assigning X and y the input and output for the machine learning model. The labels are encoded to zero index the y fetal health classes from (1,2,3) to (0,1,2) as standard machine learning models assume class labels to start from 0 and in cases starting from 0 helps direct mapping to array indices. The normalization (or feature scaling) is then done using Standard Scaler to ensure all the features contribute equally to the model even though they contain different ranges of data, so the features are scaled to have zero mean and standard deviation to 1. The train and test split of the data is done in the 80% and 20% ratio as machine learning models perform better with more training data and 20% of the test data are used to test the model performance for new unseen data maintain an equal balance of the data split. Models like XGboost and Random forest do not require feature scaling as they are based on decision trees. Fig. 1 shows the model accuracies for Dataset A.

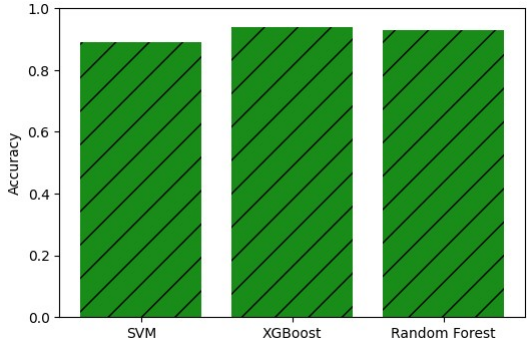


Fig. 1: Model Accuracy comparisons for Dataset A

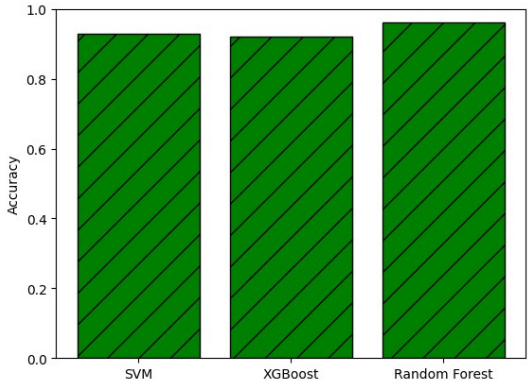


Fig. 2: Model Accuracy comparisons for Dataset B

B. Data Preprocessig for Dataset B:

The Annotation Data file with 552 fetal cardiotographic samples with their Multiclass classification of dataset into Normal (1), Pathological (0) classes is taken as our input file. It contains 37 features that are used to determine the class of fetal health of the sample. After assigning X and y the input and output for the machine learning model.The lables for y are (0,1) so no label encoding is done. The train and test split of the data is done in the 80% and 20% ratio as machine learning models perform better with more training data and 20% of the test data are used to test the model performance for new unseen data maintain an equal balance of the data split. We then perform normalization (or feature scaling) done using Standard Scaler to ensure all the features contribute equally to the model even though they contain different ranges of data, so the features are scaled to have zero mean and standard deviation to 1. Models like XGboost and Random forest do not require feature scaling as they are based on decision trees.Fig. 2 shows the model accuracies for Dataset B.

C. Addressing Class Imbalance of Dataset B:

The huge class difference with the class labeled "Normal"(507) and the class labeled "Pathological"(45).

The imbalance in the classes is dealt with by using SMOTE(Synthetic Minority Oversampling Technique). SMOTE works by creating synthetic data points for the minority class. These synthetic data points are created by interpolating between existing data points in the minority class. The imbalance in the classes often leads to problems in machine learning algorithms, and they work better on majority classes. Addressing class imbalance can help improve the performance of the models. Here, in the code, X is reshaped into 2D for SMOTE and resampled to increase the minority sample. After SMOTE is applied, the shape of X_resampled is (1012,1500,2) and y_resampled is (1012,).

D. Models Used for Dataset A and B:

1) Support Vector Machine (SVM) model: : The Support Vector Machine (SVM) is a powerful machine learning algorithm for classification and regression. It works by finding the best boundary (hyperplane) that separates different classes while keeping as much space as possible between them. This is why it's great for handling high-dimensional and complex datasets. SVC creates a SVM classifier from the library sklearn.svm, the kernel 'rbf'(Radial Basis Function) helps handle non-linear data by mapping it to a higher-dimensional space. The regularization parameter C is 1.0, which controls the trade-off between maximizing the margin and minimizing the misclassification, having gamma as the 'scale' determines the decision boundary, and 'fit(X_train, y_train)' trains the SVM model using the training data. The 'rbf' kernel is used in case the data is not linearly separable, where it maps to a higher dimension to separate the data.

2) Extreme Gradient Boosting (XGBoost) model: : The Extreme Gradient Boosting (XGBoost) algorithm is a powerful ensemble learning method based on decision trees, mostly used of structured data like classification and regression. It builds new decision trees , each new one correcting errors of the previous ones improving accuracy. It handles data efficiently and regularization reduces the overfitting. It also XGBClassifier()is a gradient boosting model bases on decision trees imported from xgboost library. The objective 'multi:softmax' is used for multi classification where it assigns and pics the class with the most probability, num classes here is 3 which represents the number of classes, and the evaluation metric is 'mlogloss'(multi class logarithmic loss) measures the model performance for predicting classes. The fit() function then trains the model using the split training data.

3) Random Forest(RF) model: : The Random forest is a ensemble learning algorithm that builds multiple decision trees and combines their outputs to improve accuracy. It averages the multiple decision trees to reduce overfitting and works well with datasets having multiple features. Random forest classifier requires the input of how many decision trees it must create, around 100-500 trees provide good performance, and random state 42 makes sure to give the same result every time the model is run, then fit() function trains the model using the split training data.

E. Training of the data:

The model is trained with the input training time series data and class labels (Normal(1), Suspect(2) and Pathological(3) for Dataset A and Normal(1), Pathological(0) for Dataset B) checking how well the model performs to new unseen data with a batch size of 16, running for 50 epochs to improve predictions, verbose is used to show live updates, displaying progress, loss and accuracy after each epoch. The test accuracy is then calculated.

F. SHAP (Shapley Additive Explanations) Values Analysis:

The data set contains an annotation for 2126 fetal cardiotocographic signals with 21 features classified into three classes, Normal, Suspect, and Pathological, The Figures 3, 4, and 5, show the SHAP values for the following classes for Dataset A. Figures 6, 7 Contain SHAP value Analys is for Dataset B contains 552 fetal cardiotocographic signals with 37 features, these signals are classified into Normal and Pathological. These figures show beeswarm plot, it is used to show how each feature contributes to the model predictions of a machine learning model. The X-axis shows the SHAP values representing the impact of each feature on the model and Y-axis shows the features sorted in the order of importance from top to bottom based on the SHAP values. In SHAP beeswarm plot, each dot represents a data point's feature contribution to the plot, the Red dot represents high feature value and Blue dot shows low feature value. The plot on the right side of the Y-axis shows the positive contribution of the feature towards that class and the left side of the axis shows its negative contribution of the feature towards the class pulling the prediction of the model away from the class. Having more red dots on the right side of the axis represents that high value of the feature contributes more to the class and having more red dots to the left side represents that low value for the feature contributes more to the class. Having more blue dots on the right side of the axis represents that low value of the feature contributes more to the class and having more blue dots to the left side represents that high value for the feature contributes more to the class.

IV. Experiment Analysis of Time Series Data of Dataset B

The models we consider here are Support Vector Machine(SVM), Extreme Gradient Boosting (XGBoost), and Random Forest(RF) to evaluate Dataset A and Dataset B. Time Series Transformer and(TST) long-short-term memory (LSTM) on Dataset B.

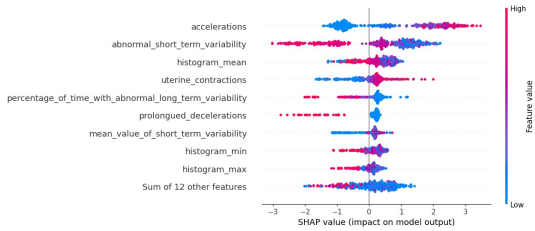


Fig. 3: SHAP Value Analysis for Normal Class of Dataset A

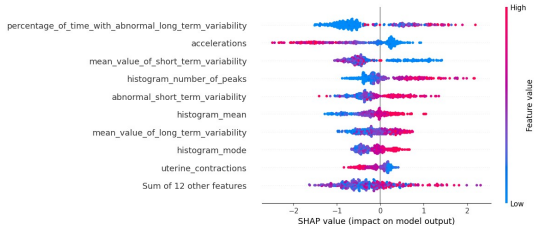


Fig. 4: SHAP Value Analysis for Suspect Class of Dataset A

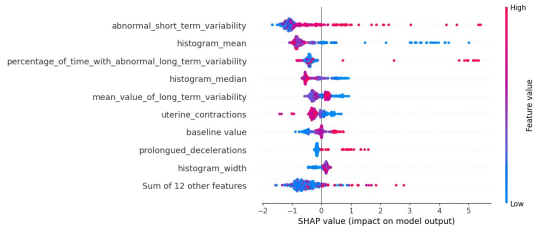


Fig. 5: SHAP Value Analysis for Pathological Class of Dataset A

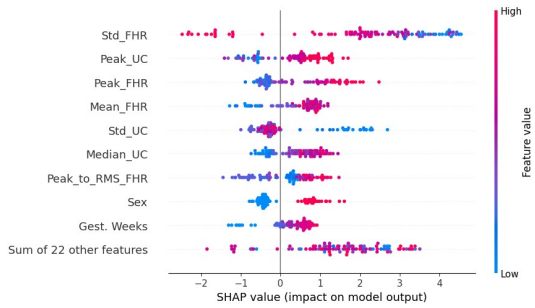


Fig. 6: SHAP Value Analysis for Normal Class of Dataset B

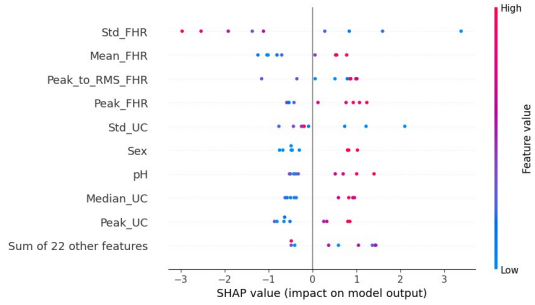


Fig. 7: SHAP Value Analysis for Pathological Class of Dataset B

A. Data Preprocessing:

The data folder containing the 552 csv samples and the ctu_df (Annotaion csv) file is loaded. From ctu_df the ”ID” and ”target” columns are taken, where the ”ID” is taken as a string. Then from the 552 csv samples we remove the ”.csv” to get the id of each file. The number of samples, sequence length taken (1500 seconds), the features (FHR and UC) stored in X_list, and ID and target values stored in y_list are converted into numpy arrays X(552,1500,2) and y (552,).

B. Addressing Class Imbalance:

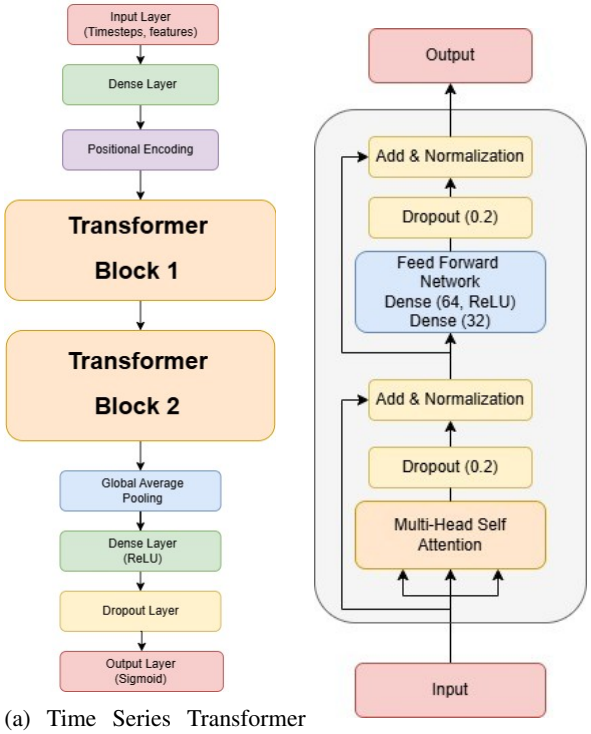
The huge class difference with the class labeled ”Normal”(507) and the class labeled ”Pathological”(45). The imbalance in the classes is dealt with by using SMOTE(Synthetic Minority Oversampling Technique). SMOTE works by creating synthetic data points for the minority class. These synthetic data points are created by interpolating between existing data points in the minority class. The imbalance in the classes often leads to problems in machine learning algorithms, and they work better on majority classes. Addressing class imbalance can help improve the performance of the models. Here, in the code, X is reshaped into 2D for SMOTE and resampled to increase the minority sample. After SMOTE is applied, the shape of X_resampled is (1012,1500,2) and y_resampled is (1012,).

C. Addressing the missing Signal Data:

We use Gausian Process Regression(GPR) and K-Nearest Neighbour(KNN) on the time series data signals containing FHR and UC signals to interpolate the missing or zero/NaN’s in the signal. As most Machine Learning and Deep Learning models do not work well with missing data or NaNs. We apply each of the models (TST and LSTM) on Raw, GPR and KNN interpolated signals to compare the model accuracies.

D. Time Series Transformer:

A Time Series Transformer is an advanced deep learning model. It utilizes multi head self attention to analyse time series Data. It has parallel processing which makes it more suitable for large dataset , fast training and interpretably. The transformer block consists of two key components, Multi Head Attention and Feed Forward Network. Multi head attention is a layer that applies self attention over the input sequence with 4 attention heads and the dimensionality of each attention head is 32, a dropout of 0.2 and layer normalization. The feed forward network processes each token separately after attention with two dense layers, one that expands features using ReLU(Rectified Linear Unit) and one that compresses it back to its original dimension followed by a dropout of 0.2 and layer normalization. Dropout of 0.2 means 20 percentage of the neurons are disconnected



(a) Time Series Transformer Model (b) Transformer Block

Fig. 8: Time Series Transformer Model (b) Transformer Block

to avoid overfitting and Layer normalization normalizes the inputs across all the features and performs residual connection (inputs + attention) to ensure information is not lost. The Figure-6 and Figure-7 shows the architecture of the transformer block and time series transformer used.

The Time Series Transformer model takes in sequence length and number of features for each sample. Through the dense layer the time series data is projected into embedded space, which goes through positional encoding as the Transformers do not process sequences in order, where positional embeddings are created which are considered unique identities for each timestep and add the position embedding to the input data. The data is then passes through the transformer blocks and Global Average pooling compresses the sequence into a single feature vector per sample. The Dense layer with ReLU (Rectified Linear Unit) activation followed by dropout of 0.2 and binary classification to Normal(1) and Pathological(0) through a dense layer with sigmoid activation. The model is built and compiles using Adam(Adaptive Moment Estimation) optimizer with a learning rate of 0.001. Since this is a binary classification, "Binary Cross entropy" is used to measure loss by measuring how well the model is predicting, and "accuracy" is used to determine how many predictions are correct. Fig. 8 shows the Transformer Block and the Time Series Transformer Model.

E. LSTM (Long Short-Term Memory):

The LSTM (Long Short-Term Memory) model is a type of RNN (Recurrent Neural Network) model used to handle sequential data and capture long term dependencies. It remembers the information for long sequences, reduces vanishing gradient problem and works well for time series data. The LSTM model here consists of 6 layers, the input layer take the time sequence and features as input, the first LSTM layer with 64 units processes time dependencies and returns full sequence for next LSTM layer, the second layer processes it and reduces it to 32 units and returns only last hidden state for final classification. The dense layer with 32 neurons and ReLU activation adds a fully connected layer to learn deeper pattens, and uses ReLU(Rectifies Linear Unit) for non linearity, Then the Dropout layer drops 20% of the neurons during training to prevent overfitting, and the final layer Output Dense layer with sigmoid activation for binary classification (Normal(1) and Pathological(0)).The model is then built and compiles using optimizer Adam(Adaptive Moment Estimation) with a learning rate of 0.001 . Since this a binary classification ,"Binary Cross entropy" is used to measure loss by measuring how well the model is predicting, and "accuracy" is used to determine how many predictions are correct. Fig. 9 shows the LSTM model architecture used.

F. Training of the data:

The model is trained with the input training time series data (Raw Signal Data, GPR Signal Data and KNN Signal Data) and class labels (Normal(1) and Pathological(0)) obtained from Annotation Data, checking how well the model performs to new unseen data with a batch size of 16, running for 50 epochs to improve predictions, verbose is used to show live updates, displaying progress, loss and accuracy after each epoch. The test accuracy is then calculated.

V. Results:

Here, Figure-10 shows the obtained Accuracy, Macro and Weighted values for Precision, Recall and F1-Score for SVM, XGBoost and Random Forest models of both Dataset A, Dataset B and also for Time Series Transformer (TST) and Long Short-term Memory(LSTM) Model application on all Raw, GPR and KNN time signal data.

Acknowledgment

References

[1] D. Campos and J. Bernardes, "Cardiotocography," UCI Machine Learning Repository, 2000, DOI: https://doi.org/10.24432/C51S4N.

[2] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

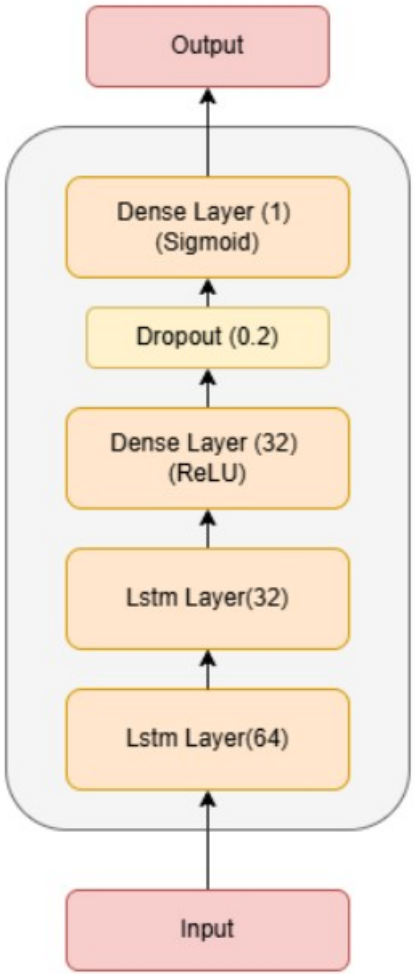


Fig. 9: LSTM Model

Model	Accuracy	Precision		Recall		F1-Score	
		Macro	Weighted	Macro	Weighted	Macro	Weighted
Dataset-A:							
SVM	0.89	0.82	0.89	0.75	0.89	0.78	0.89
XGBoost	0.94	0.90	0.93	0.88	0.94	0.89	0.94
Random Forest	0.93	0.88	0.92	0.84	0.93	0.86	0.92
Dataset-B:							
SVM	0.93	0.93	0.93	0.93	0.93	0.93	0.93
XGBoost	0.92	0.92	0.92	0.92	0.92	0.92	0.92
Random Forest	0.96	0.96	0.96	0.96	0.96	0.96	0.96
Dataset-B Time Series Data:							
TST	0.83	0.85	0.85	0.83	0.83	0.83	0.83
TST_GPR	0.73	0.73	0.73	0.73	0.73	0.73	0.73
TST_KNN	0.64	0.64	0.64	0.63	0.64	0.63	0.63
LSTM	0.76	0.76	0.76	0.76	0.76	0.76	0.76
LSTM_GPR	0.65	0.66	0.66	0.65	0.65	0.64	0.64
LSTM_KNN	0.62	0.64	0.64	0.62	0.62	0.61	0.61

Fig. 10: Results of Dataset A, Dataset B and its Time Series Data