# **Models Trained with their respective parameters and metrics.**

|  |  |  |
| --- | --- | --- |
| Model Name | Parameters and Hyper Parameters | Metrics |
| Bidirectional LSTM  MFCCs sequence  (frame\_length=0.025 hop\_length=0.01 num\_samples= 700)  NON-BIASED SPLIT  [BEST MODEL] | def build\_lstm\_bidirectional(input\_shape, output\_units=2, learning\_rate=0.0001):      model = Sequential()      model.add(Bidirectional(LSTM(units=32, activation='tanh'), input\_shape=input\_shape))      model.add(Dense(units=output\_units, activation='softmax'))        adam\_optimizer = Adam(learning\_rate=learning\_rate)      model.compile(optimizer=adam\_optimizer,                    loss='sparse\_categorical\_crossentropy',                    metrics=['accuracy'])        return model | Acc: 79.17% |
| LSTM  MFCCs sequence  (frame\_length=0.025 hop\_length=0.01 num\_samples= 700)  NON-BIASED SPLIT | def build\_lstm(input\_shape, output\_units=2, learning\_rate=0.0001):      # Model architecture      model = Sequential()      model.add(LSTM(units=32, return\_sequences=False, input\_shape=input\_shape, activation='tanh'))      model.add(Dense(units=output\_units, activation='softmax'))        # Compile the model with a specified learning rate      adam\_optimizer = Adam(learning\_rate=learning\_rate)      model.compile(optimizer=adam\_optimizer,                    loss='sparse\_categorical\_crossentropy',                    metrics=['accuracy'])        return model | Acc:  75% |
| GRU  MFCCs sequence  (frame\_length=0.025 hop\_length=0.01 num\_samples= 700)  NON-BIASED SPLIT | def build\_gru(input\_shape, output\_units=2, learning\_rate=0.0001):      model = Sequential()      model.add(GRU(units=32, activation='tanh', input\_shape=input\_shape))      model.add(Dense(units=output\_units, activation='softmax'))        adam\_optimizer = Adam(learning\_rate=learning\_rate)      model.compile(optimizer=adam\_optimizer,                    loss='sparse\_categorical\_crossentropy',                    metrics=['accuracy'])        return model | ACC:  70.8% |
| Bidirectional LSTM  MFCCs sequence  (frame\_length=0.025 hop\_length=0.01 num\_samples= 2000)  NON-BIASED SPLIT | Same as above | Acc: 75% |
| Bidirectional LSTM  MFCCs sequence  (frame\_length=0.025 hop\_length=0.01 num\_samples= 700)  BIASED SPLIT | Same as above | Acc:  83.33% |
| SVM  (Mean MFCCs)  BIASED SPLIT | SVC(kernel='rbf', C = 10) | Acc:  79.16% |
| Random Forest  (Mean MFCCs)   BIASED SPLIT | (RandomForestClassifier(n\_estimators=1500, max\_leaf\_nodes=10,max\_depth=50) | Acc:  87.5% |
| SVM  (Mean MFCCs)  NON-BIASED SPLIT | Same as above | Acc:  54.16% |
| Random Forest  (Mean MFCCs)   NON-BIASED SPLIT | Same as above | Acc:  58.33% |
|  |  |  |

Results are only recorded on court trial set.

Note:

The accuracies recorded above are not always achieved, sometimes the model performs much worse yielding even 50% accuracy, this can be attributed to the random weights initialization, the miniscule amount of data that is fed into the model, and maybe even possibly the fact that it is easier for the model to achieve a biased result where it gets good train results and poor test results simply because the weights are adjusting to the train set in which multiple subjects appear many times in one class (lie, truth) which explains why it performs well on training set but poor on the manual split test set that doesn’t have those repeating subjects in the training set, nonetheless, after several attempts accuracies above 70% can be achieved on both training and testing sets which signals that the model is actually detection deceit from audio cues.