10. Results & Discussion

Results of different experiments carried out in different phases have been carefully recorded for further evaluation. First of all, the class distribution in the newly formed dataset has been observed. It has been found that y-axis values are increasing as x-axis values are increasing (Figure 59) for most of the data points. This depicts the positive correlations in the feature set. This also justifies the use of the newly developed dataset in the present study.

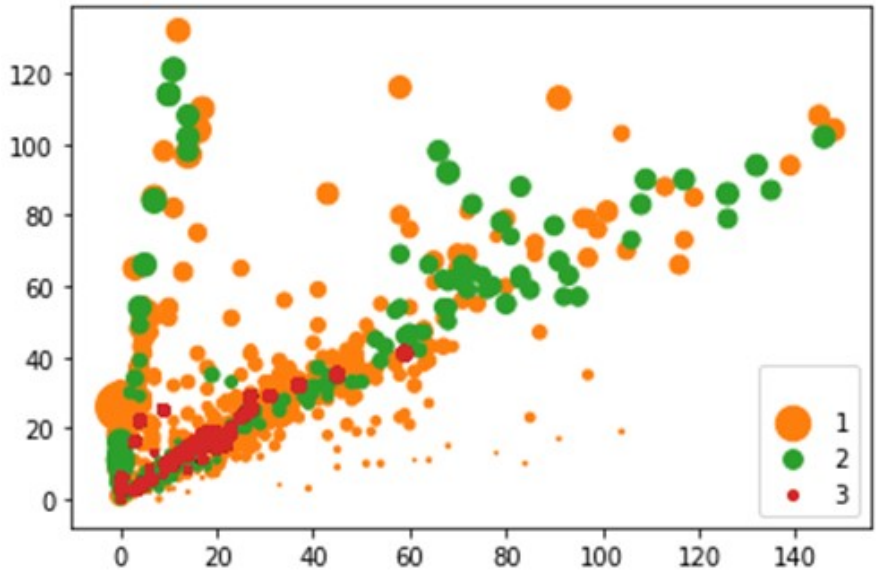


Figure 59: Class distribution observed in the dataset used in the study (1=> Malignant; 2=>Benign; 3=> others)

10.1 One Dimensional CNN

Figure 60 shows a sample of runtime structure of the newly developed model which is a combination of One Dimensional Convolutional Neural Network (1D CNN) and Recurrent Neural Network (RNN). Input parameters of a one-dimensional convolutional layer are (batch, steps, channels) and the output parameters are (batch, new_steps, filters). The first Conv1d layer has an input shape (None, 63, 1) and its output shape is (None, 61, 64). Here, *None* asserts that the batch length is not fixed; *63* is the present number of features (0… 63); *61* is the number of features to be used next, typically calculated as (no. of features – kernel size + filter bias) i.e. (63-3+1); *64* is the size of the filter/kernel used in the layer. The no. of parameters in Conv1d_37 is determined as (no. of

filters*filter size) +layer bias = (64*3+64) =256. As it is the initial layer, it does not have any output from previous layers.

Layer (type)	Output Shape	Param #	Calculation
conv1d_37 (Conv1D)	(None, 61, 64)	256	64*3+64=256
max_pooling1d_37	MaxPooling (None, 61, 64)	0	
conv1d_38 (Conv1D)	(None, 59, 32)	6176	64*3*32+32=6176
max_pooling1d_38	MaxPooling (None, 59, 32)	0	
conv1d_39 (Conv1D)	(None, 57, 16)	1552	32*3*16+16=1552
max_pooling1d_39	MaxPooling (None, 57, 16)	0	
conv1d_40 (Conv1D)	(None, 55, 8)	392	16*3*8+8=392
max_pooling1d_40	MaxPooling (None, 55, 8)	0	
gru_10 (GRU)	(None, 55, 7)	336	21*7+21*8+21=336
flatten_10 (Flatten)	(None, 385)	0	
dense_19 (Dense)	(None, 64)	24704	385*64+64=24704
dense_20 (Dense)	(None, 4)	260	64*4+4=260

Total params: 33,676
Trainable params: 33,676
Non-trainable params: 0

Figure 60: Runtime structure of the model comprising 1D CNN and RNN for AJCC Staging

For other convolutional layers, numbers of parameters are measured as (no. of filters*filter size*output filter from previous layers) +layer bias. For example, number of parameters in conv1d_38 = (64*3*32) +32=6176 and so on. Maxpooling or flatten layers don't have any trainable parameters. For gru_10 layer, number of parameters are calculated as (no. of kernel+ no. of recurrent kernel+ bias) = (21*7+21*8+21) = 336. For dense_20 layer, the no. of parameters= (input from dense_19*no. of filters) +bias= (64*4) +4=256+4=260.

10.1.1 Outcome of Experiment

Algorithm	Accuracy	F-Measure	ROC Area	RMSE	Kappa	Average Cost
CNN	0.91±0.04	0.90±0.02	0.96±0.05	0.25±0.06	0.87±0.07	0.24±0.21
CNN-RNN	0.97±0.02	0.95±0.01	1±0.01	0.1±0.01	0.92±0.04	0.09±0.1
FRNN	0.87±0.03	0.85±0.03	0.9±0.08	0.34±0.05	0.8±0.11	0.34±0.39
KNN	0.83±0.03	0.80±0.07	0.89±0.06	0.29±0.08	0.75±0.09	0.1±0.34
LR	0.72±0.03	0.71±0.08	0.79±0.07	0.19±0.08	0.65±0.11	0.97±0.39
MLP	0.76±0.03	0.74±0.08	0.83±0.05	0.38±0.07	0.69±0.11	1.4±0.41
RF	0.85±0.03	0.84±0.07	0.96±0.06	0.27±0.07	0.76±0.1	0.37±0.32
SVM	0.72±0.06	0.73±0.08	0.8±0.07	0.18±0.08	0.64±0.12	0.95±0.4

Table3: evaluation of grading by various algorithms concerning different metrics (CNN=>Convolutional Neural Network, CNN-RNN=> Convolutional Neural Network combined with Recurrent Neural Network, FRNN=>Fuzzy Rough Nearest Neighbours, KNN=>K-Nearest Neighbours, LR=>Logistic Regression, MLP=>Multi-Layer Perceptron, RF=> Random Forest, SVM=> Support Vector Machines)

Algorithm	Accuracy	F-Measure	ROC Area	RMSE	Kappa	Average Cost
CNN	0.85±0.02	0.85±0.03	0.91±0.06	0.27±0.04	0.78±0.05	0.8±0.3
CNN-RNN	0.96±0.01	0.96±0.01	0.99±0.01	0.07±0.02	0.93±0.03	0.11±0.2
FRNN	0.72±0.08	0.74±0.08	0.81±0.05	0.19±0.07	0.67±0.1	0.98±0.38
KNN	0.68±0.08	0.66±0.09	0.78±0.07	0.38±0.08	0.59±0.12	1.15±0.39
LR	0.6±0.05	0.58±0.12	0.6±0.08	0.24±0.09	0.39±0.19	1.23±0.5
MLP	0.67±0.09	0.65±0.09	0.78±0.05	0.32±0.07	0.59±0.13	1.51±0.61
RF	0.67±0.06	0.65±0.07	0.78±0.04	0.41±0.06	0.61±0.09	1.12±0.09
SVM	0.5±0.09	0.51±0.11	0.58±0.12	0.88±0.1	0.44±0.14	1.54±0.55

Table4: evaluation of T-Staging by various algorithms concerning different metrics (CNN=>Convolutional Neural Network, CNN-RNN=> Convolutional Neural Network combined with Recurrent Neural Network, FRNN=>Fuzzy Rough Nearest Neighbours, KNN=>K-Nearest Neighbours, LR=>Logistic Regression, MLP=>Multi-Layer Perceptron, RF=> Random Forest, SVM=> Support Vector Machines)

Algorithm	Accuracy	F-Measure	ROC Area	RMSE	Kappa	Average Cost
CNN	0.93±0.02	0.95±0.03	0.97±0.05	0.21±0.05	0.89±0.06	0.39±0.2
CNN-RNN	0.98±0.01	0.98±0.01	1±0.0	0.09±0.02	0.94±0.07	0.09±0.1
FRNN	0.92±0.05	0.91±0.04	0.99±0.03	0.2±0.01	0.87±0.05	0.45±0.4
KNN	0.82±0.06	0.78±0.07	0.88±0.05	0.3±0.02	0.7±0.11	0.41±0.4
LR	0.9±0.06	0.87±0.04	0.98±0.02	0.09±0.01	0.82±0.07	0.59±0.38
MLP	0.82±0.05	0.80±0.03	0.87±0.06	0.28±0.06	0.75±0.12	0.74±0.2
RF	0.83±0.07	0.84±0.05	0.9±0.03	0.21±0.04	0.75±0.09	0.38±0.1
SVM	0.8±0.09	0.78±0.06	0.87±0.04	0.18±0.02	0.71±0.07	0.7±0.9

Table5: evaluation of N-Staging by various algorithms concerning different metrics (CNN=>Convolutional Neural Network, CNN-RNN=> Convolutional Neural Network combined with Recurrent Neural Network, FRNN=>Fuzzy Rough Nearest Neighbours, KNN=>K-Nearest Neighbours, LR=>Logistic Regression, MLP=>Multi-Layer Perceptron, RF=> Random Forest, SVM=> Support Vector Machines)

Algorithm	Accuracy	F-Measure	ROC Area	RMSE	Kappa	Average Cost
CNN	0.95±0.02	0.92±0.04	0.98±0.03	0.23±0.06	0.9±0.04	0.29±0.1
CNN-RNN	0.99±0.01	0.98±0.02	1±0.01	0.09±0.02	0.94±0.01	0.08±0.09
FRNN	0.93±0.03	0.91±0.05	0.98±0.02	0.19±0.3	0.88±0.02	0.43±0.09
KNN	0.88±0.02	0.87±0.06	0.92±0.06	0.27±0.01	0.84±0.07	0.9±0.7
LR	0.92±0.04	0.86±0.06	0.98±0.05	0.08±0.7	0.85±0.07	0.41±0.8
MLP	0.88±0.02	0.89±0.08	0.92±0.05	0.24±0.01	0.84±0.06	0.51±0.9
RF	0.87±0.02	0.86±0.07	0.95±0.04	0.23±0.04	0.78±0.05	0.35±0.09
SVM	0.88±0.02	0.85±0.07	0.96±0.03	0.11±0.1	0.84±0.03	0.53±0.8

Table6: evaluation of M-Staging by various algorithms concerning different metrics (CNN=>Convolutional Neural Network, CNN-RNN=> Convolutional Neural Network combined with Recurrent Neural Network, FRNN=>Fuzzy Rough Nearest Neighbours, KNN=>K-Nearest Neighbours, LR=>Logistic Regression, MLP=>Multi-Layer Perceptron, RF=> Random Forest, SVM=> Support Vector Machines)

Algorithm	Accuracy	F-Measure	ROC Area	RMSE	Kappa	Average Cost
CNN	0.86±0.07	0.87±0.03	0.94±0.06	0.31±0.03	0.78±0.05	0.35±0.1
CNN-RNN	0.97±0.02	0.96±0.02	1±0.01	0.23±0.001	0.93±0.02	0.32±0.08
FRNN	0.71±0.08	0.71±0.05	0.82±0.09	0.27±0.02	0.66±0.09	0.89±0.2
KNN	0.72±0.09	0.72±0.04	0.81±0.04	0.35±0.3	0.69±0.1	1.18±0.42

LR	0.59±0.2	0.61±0.04	0.67±0.08	0.23±0.45	0.49±0.19	1.33±0.38
MLP	0.66±0.1	0.64±0.08	0.75±0.07	0.34±0.07	0.58±0.08	1.52±0.39
RF	0.68±0.1	0.67±0.05	0.79±0.06	0.39±0.09	0.59±0.08	1.18±0.27
SVM	0.52±0.02	0.52±0.05	0.62±0.06	0.86±0.08	0.42±0.1	1.53±0.35

Table7: evaluation of AJCC Staging by various algorithms concerning different metrics (CNN=>Convolutional Neural Network, CNN-RNN=> Convolutional Neural Network combined with Recurrent Neural Network, FRNN=>Fuzzy Rough Nearest Neighbours, KNN=>K-Nearest Neighbours, LR=>Logistic Regression, MLP=>Multi-Layer Perceptron, RF=> Random Forest, SVM=> Support Vector Machines)

Table3 shows the performances of different machine learning algorithms, including the newly developed CNN and CNN+RNN models while classifying histopathological grades. Similarly, table4 shows the performances during T-staging, table5 shows the same for N-staging, table6 shows it for M-staging, and table7 records it for AJCC staging. All of these tables have tabulated mean values of accuracy, F-measure, ROC area, RMSE, Kappa, and average cost along with their respective standard deviations for each method. From the result, it has been observed that the CNN+RNN model has performed more satisfactorily than other methods. Not only the accuracy of the CNN-RNN model is higher, but also the F-score is consistently on the higher side. This shows higher true positive values and lesser false positive and false negative values, i.e., lesser type-I and type-II errors respectively. This does not only show the relevance in the positive case detection but also shows the high rate of correctly identified negative cases. The High ROC area shows a greater true positive rate than the false positive rate. This implies that most of the positive cases have been identified correctly and the High Kappa value affirms the accuracy by showing perfect agreement between the true value and the predicted value. Low RMSE shows data concentration is intense along the line of best fit. The average cost is also lower for the CNN+RNN model than others. Thus, the performance of the CNN+RNN model concerning all the evaluation metrics is better than other machine learning methods. All these observations cast a decision in favor of the newly developed CNN+RNN model as far as the classification of malignant tumors is concerned.

```
acc: 1.0000 - val_loss: 0.0261 - val_acc: 0.9894
54/54 [==============================] - 0s 3ms/step
[0 0 3 1 0 2 0 1 0 0 6 1 0 2 1 1 3 1 1 3 0 3 1 2 0 2 3 3 1 6 3 0 2 1 4 4
0
 1 0 0 3 3 3 0 0 0 1 3 2 6 0 0 0 2]
['IA' 'IA' 'IIB' 'IB' 'IA' 'IIA' 'IA' 'IB' 'IA' 'IA' 'O' 'IB' 'IA' 'IIA'
 'IB' 'IB' 'IIB' 'IB' 'IB' 'IIB' 'IA' 'IIB' 'IB' 'IIA' 'IA' 'IIA' 'IIB'
 'IIB' 'IB' 'O' 'IIB' 'IA' 'IIA' 'IB' 'IIIA' 'IIIA' 'IA' 'IB' 'IA' 'IA'
 'IIB' 'IIB' 'IIB' 'IA' 'IA' 'IA' 'IB' 'IIB' 'IIA' 'O' 'IA' 'IA' 'IA'
 'IIA']
```

Figure 61: Inverse transformation of the AJCC stage prediction by the 1D CNN encoder model

From figure 61, the accuracy of the decoded prediction of AJCC staging done by the 1D CNN encoder model may be observed. The validation accuracy is around 98% which is quite encouraging and speaks for the efficacy of the model itself.

10.1.2 Analysis of Result

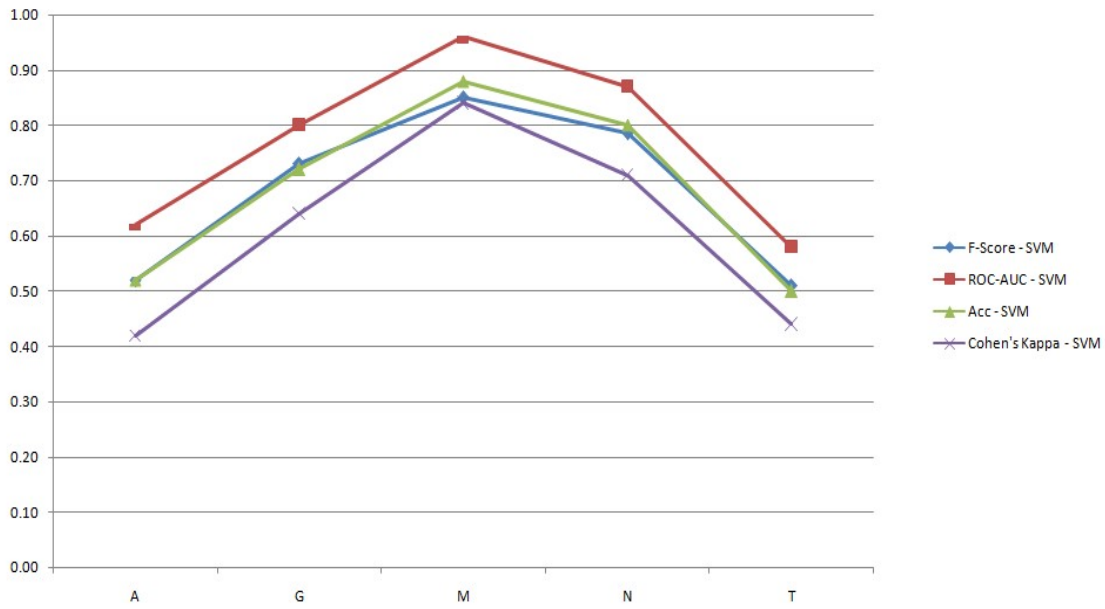


Figure 62: Performance of Support Vector Machines (SVM) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Charasteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

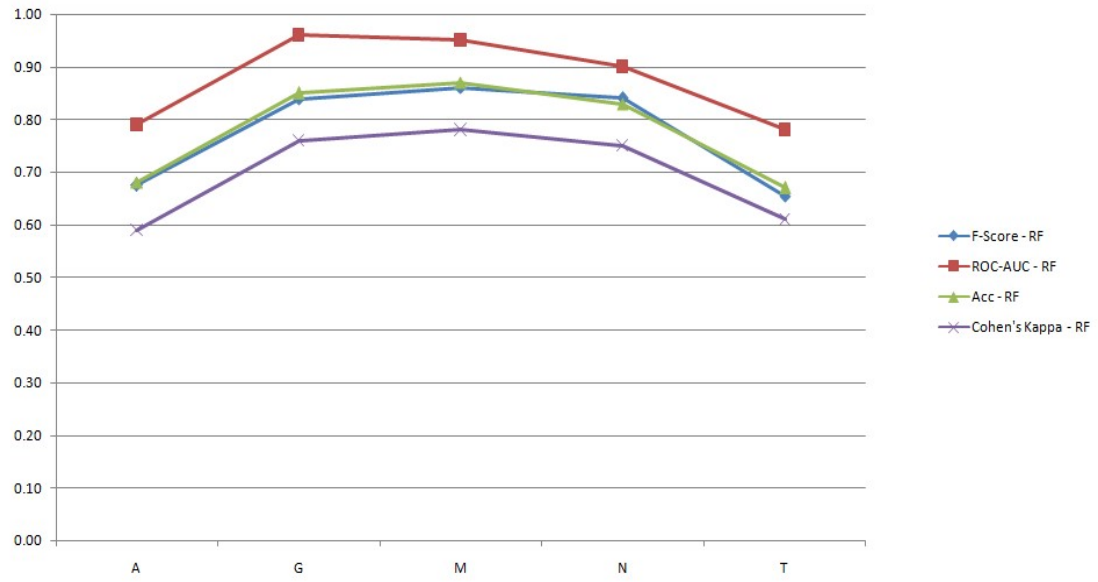


Figure 63: Performance of Random Forest (RF) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

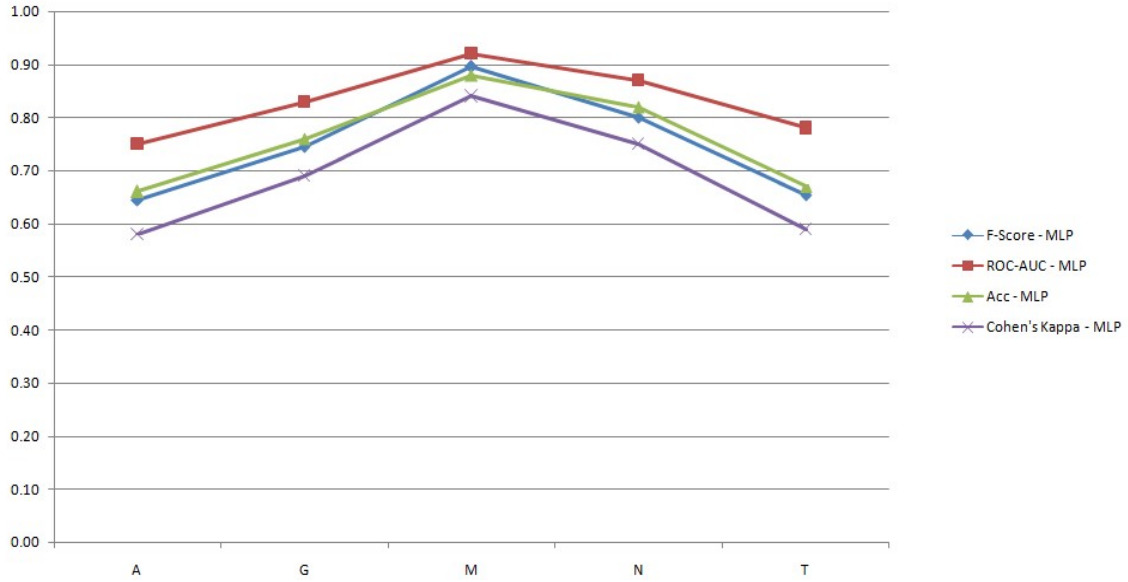


Figure 64: Performance of Multi-Layer Perceptron (MLP) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

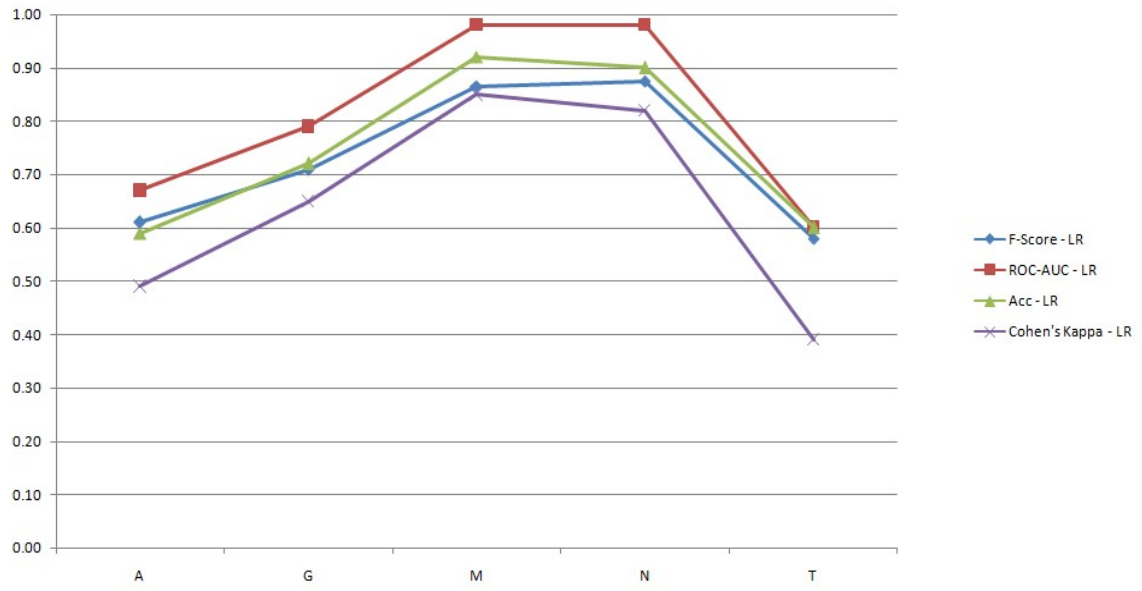


Figure 65: Performance of Logistic Regression (LR) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

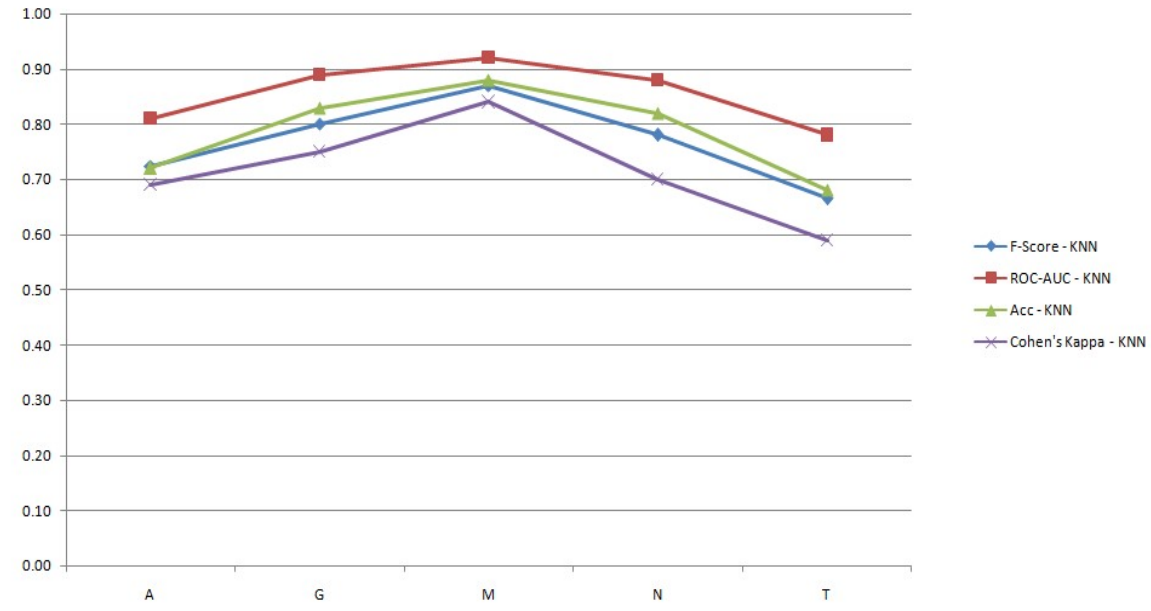


Figure 66: Performance of K-Nearest Neighbours (KNN) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

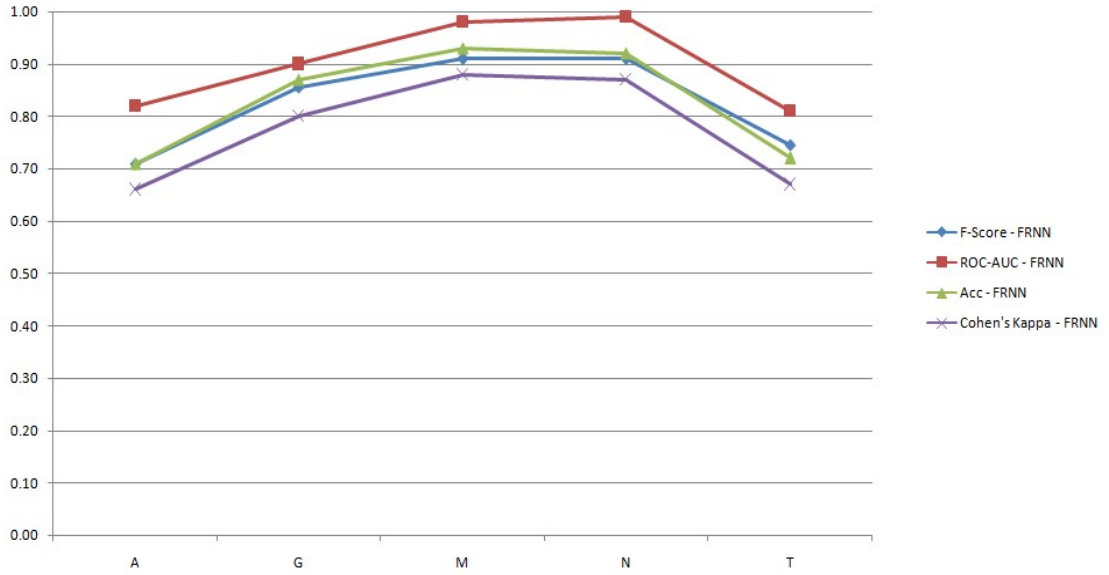


Figure 67: Performance of Fuzzy Rough Nearest Neighbours (FRNN) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Charasteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

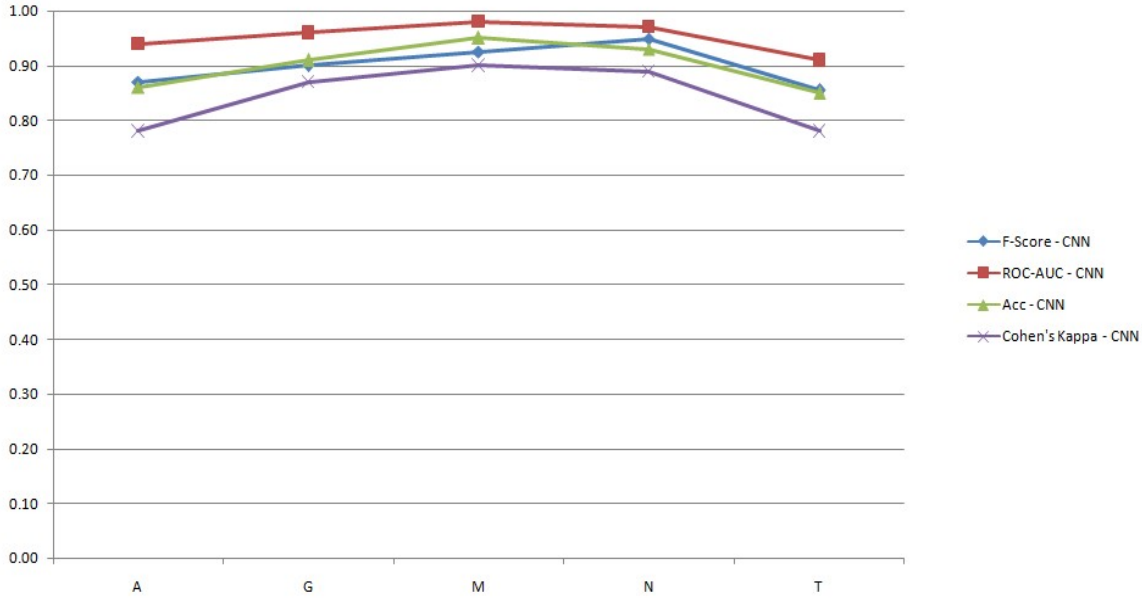


Figure 68: Performance of Convolutional Neural Network (CNN) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

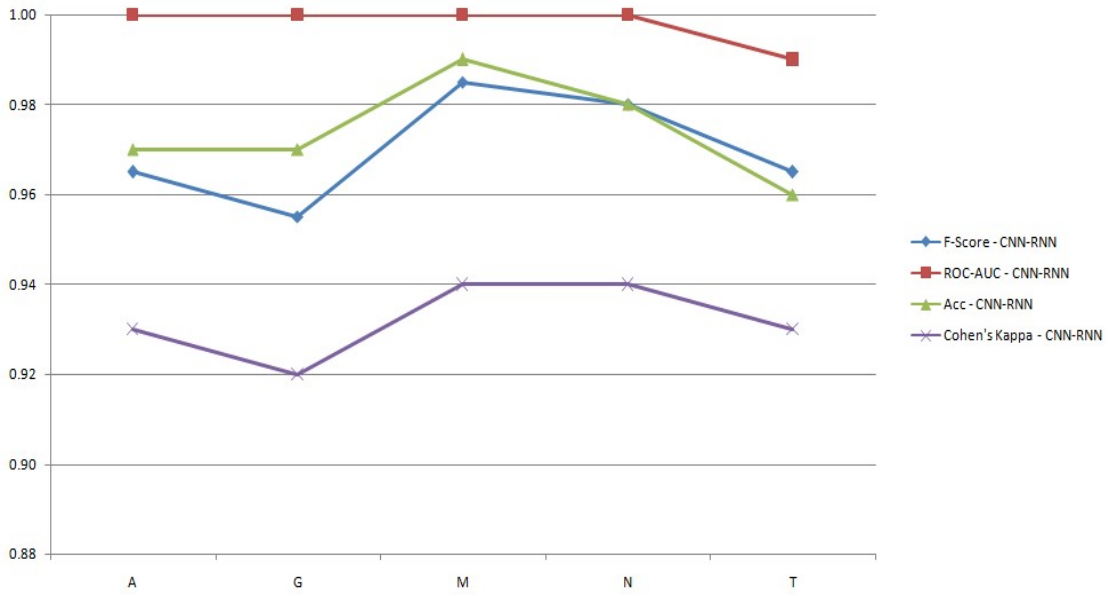
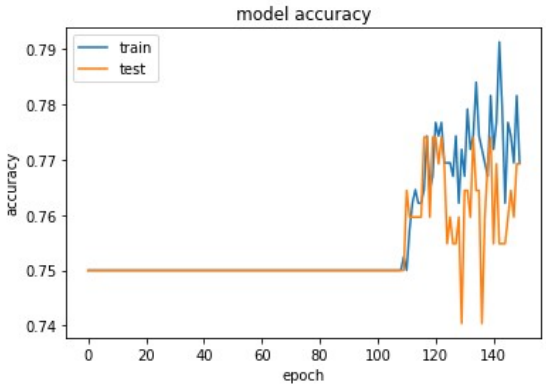


Figure 69: Performance of Convolutional Neural Network combined with-Recurrent Neural Network (CNN-RNN) with respect to major evaluation parameters (A=> AJCC staging, G=>grading, M=>M-staging, N=>N-staging, T=>T-staging, Acc=>Accuracy, F-Score=> F-Measure, ROC-AUC=> Receiver Operating Characteristics Curve – Area Under Curve, Cohen's Kappa=> Kappa Statistics)

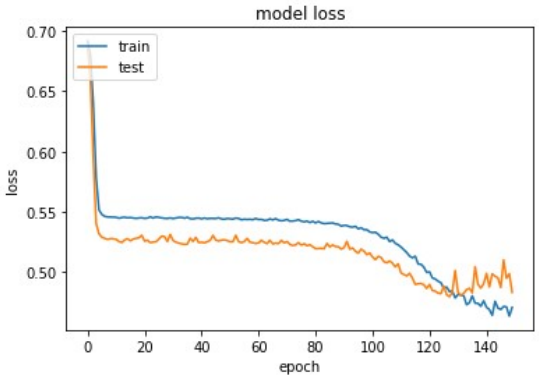
Figures 62, 63, 64, 65, 66, 67, 68, and 69 show the graphical comparison of performances of each machine learning method used in the study while conducting different staging and grading operations. The analysis reveals that all the machine learning methods have performed well when the number of class variables is less in number. Thus, in the case of N-staging or M-staging, performances of different algorithms are satisfactory. In such cases, the classification problems are less complicated with less number of target variables almost like a binary classification task. But the situation changes when target variables are more in numbers. In such cases, the algorithms face true multivariable-multiclass problems with real numbers having higher precisions as data values which have very tiny differences among each other. Orthodox machine learning methods find it very difficult to perform steadily in such cases. As a result, the performance curve for these algorithms concerning different parameters falls while classifying in terms of grade, AJCC stage, or T-stage. The newly developed CNN+RNN and CNN models perform commendably even in such adverse situations. Their performance curves always stay higher than the other methods employed in the study. These observations depict the superiority of convolutional models over other traditional models. CNN models even

performed better than the hybrid FRNN or MLP models. CNN+RNN model has a higher consistent curve than the simple CNN model. Although the CNN model has deeper convolutional layers, the CNN+RNN model has an edge over the former one. It has got recurrent layers which can memorize the past results. Thus while down-sampling features; the earlier important ones do not get faded away. This characteristic gives the later one an extra boost during classification.

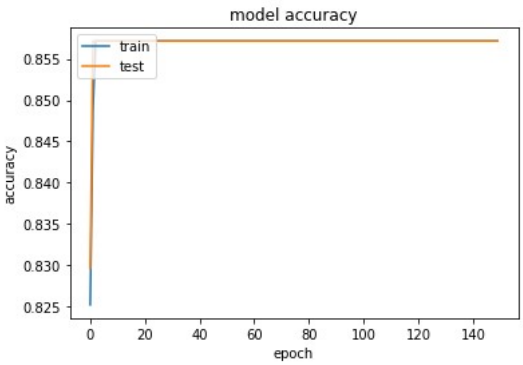
Although the CNN+RNN model has performed better than the CNN model, there are indeed some close calls observed. Before selecting the best possible model, further analysis of the performance of these two models has been done by using the validation accuracy curve and loss curve.



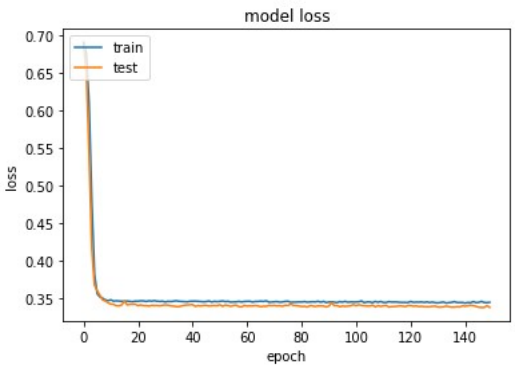
(a) grading accuracy



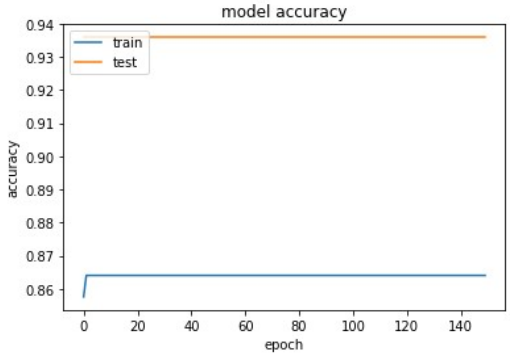
(b) grading loss



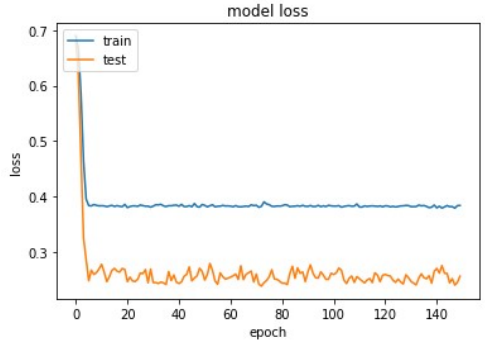
(c) T-staging accuracy



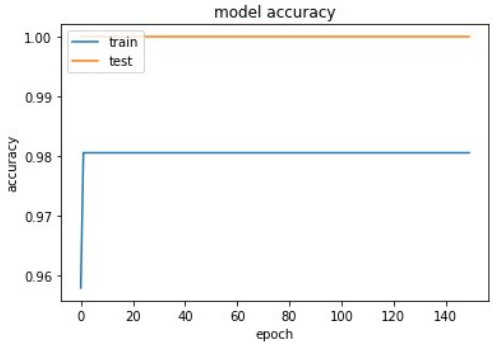
(d) T-staging loss



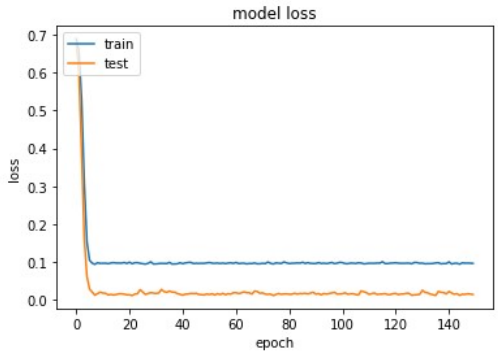
(e) N-staging accuracy



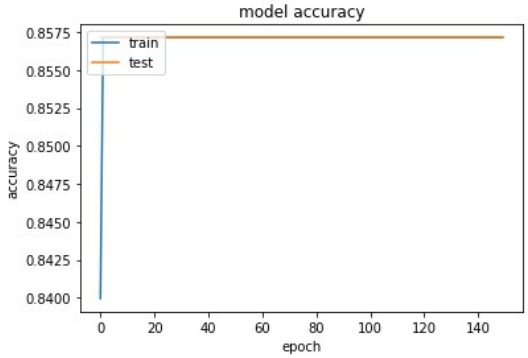
(f) N-staging Loss



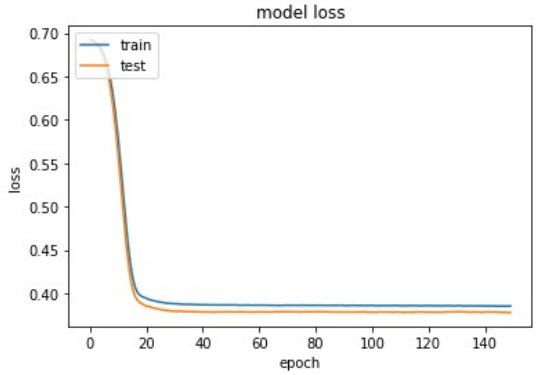
(g) M-staging accuracy



(h) M-staging loss

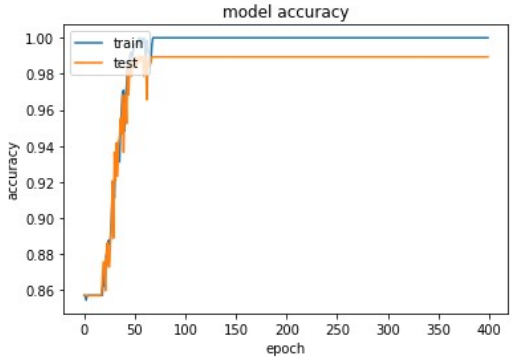


(i) AJCC staging accuracy

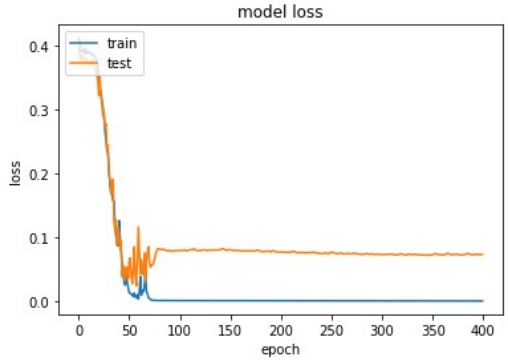


(j) AJCC staging loss

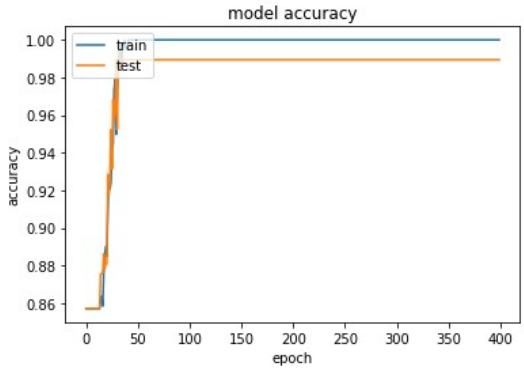
Figure 70: Train and test curves for CNN model accuracy and CNN model loss



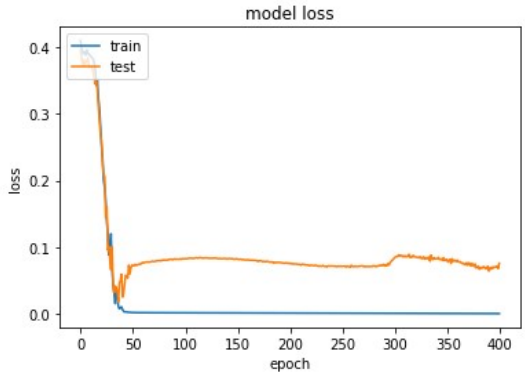
(a) grading accuracy



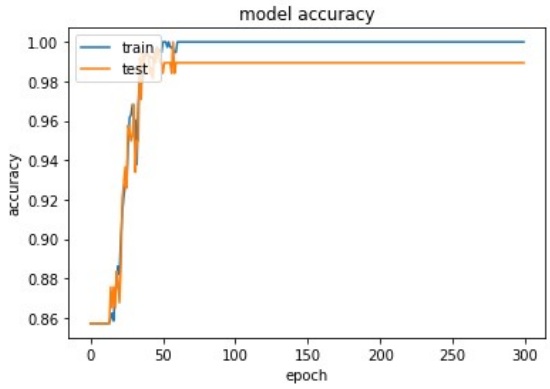
(b) grading loss



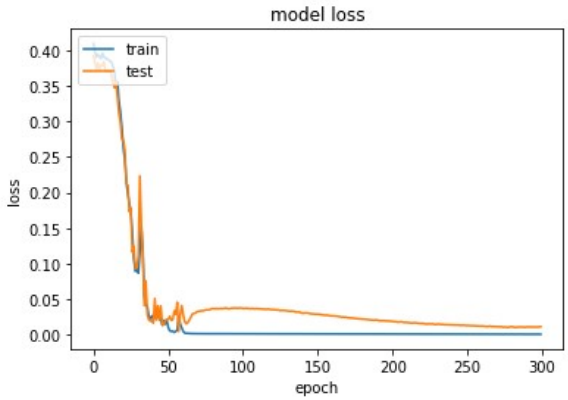
(c) T-staging accuracy



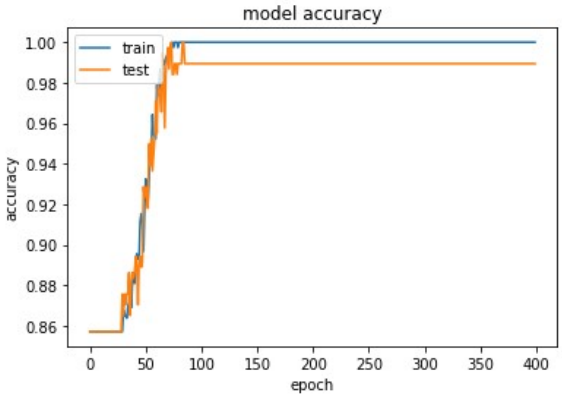
(d) T-staging loss



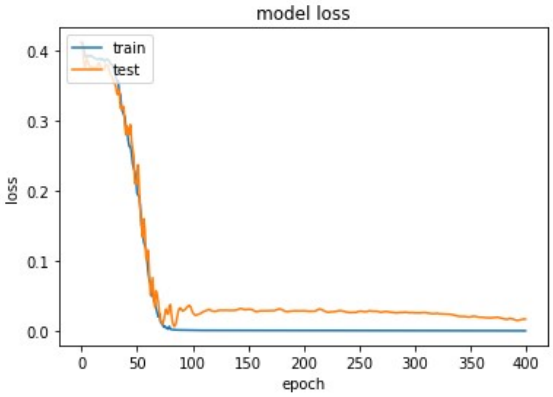
(e) N-staging accuracy



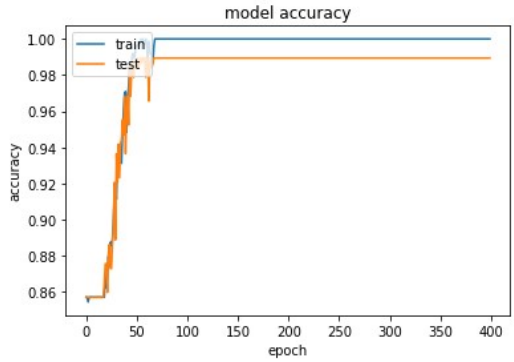
(f) N-staging loss



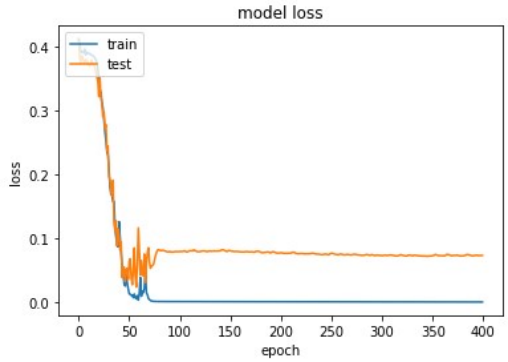
(g) M-staging accuracy



(h) M-staging loss



(i) AJCC staging accuracy



(j) AJCC staging loss

Figure 71: Train and test curves for CNN+RNN model accuracy and loss

After comparing figures 70 and 71, it is quite clear that the CNN model suffers from the problem of overfitting. There are frequent intersections between training and validation curves and fluctuations are also observed at later stages of iterations. In some cases, the test accuracy has been more than the training accuracy. This happens as many features become dormant due to down-sampling, dropout, and regularization operations. Thus, training accuracy becomes lower and loss becomes higher, however, during the test phase, all the features remain active and result in higher accuracy and lower loss. In this way, the CNN model becomes less trustworthy for validating new cases. On the other hand, the CNN+RNN model shows fewer intersections between the train and test curves. Initial fluctuations are stabilized by regularizes which is evident from figure 71. The training accuracy is always higher than the test accuracy and the training loss is always lower than the test loss. This shows that the CNN+RNN model is quite stable and has not lost important features during the training phase due to the presence of recurrent layers.

Thus, this model is more trustworthy for validating new or unknown cases. These results of analysis undoubtedly give a verdict in favor of the CNN+RNN model.

10.2 Two Dimensional CNN

Two dimensional CNN model has many facets: it starts with a simple CNN model, and then gradually turned into a CNN combined with bidirectional RNN (CNN+BiRNN). After that, it becomes a sequential ensemble of CNN and bidirectional RNN. Later, a new non-sequential recurrent model has been introduced.

10.2.1 Outcome of Experiment

All the experiments have been carried out for 5000 epochs with *early stop* callback value along with a 10-Fold Repeated Stratified Cross-validation and *patience=200*. The repeated stratification ensures that the cases selected for validation are not correlated. Models have been employed mainly to detect AJCC staging as TNM staging can easily be retrieved from it. In the case of the CNN model, the best result has been found at epoch 400 of iteration 6. The best result of the single CNN+BiRNN model has been found at epoch 450 of iteration 3. The CNN+BiRNN model ensemble achieved the best result at epoch 492 of iteration 5. The non-sequential model ensemble has attained the best result at epoch 300 of iteration 2.

Model	Validation Accuracy	F1-Score	Cohen's Kappa	ROC AUC
Non-sequential CNN Model Ensemble	0.93	0.92	0.86	0.98
Sequential CNN+BiRNN Model Ensemble	0.84	0.84	0.75	0.9
Sequential CNN+BiRNN Model	0.80	0.79	0.70	0.87
Sequential CNN Model	0.65	0.67	0.59	0.76

Table8: Best results recorded for various models used in the study

From table8, it is quite clear that the non-sequential model has almost outperformed other models w.r.t. all the important evaluation parameters. The same may be observed from the comparison of confusion matrices produced by different models (figure72). The credit of correctly classifying the highest number of instances per class goes to the non-sequential model. These results depict that the non-sequential model is pretty consistent.

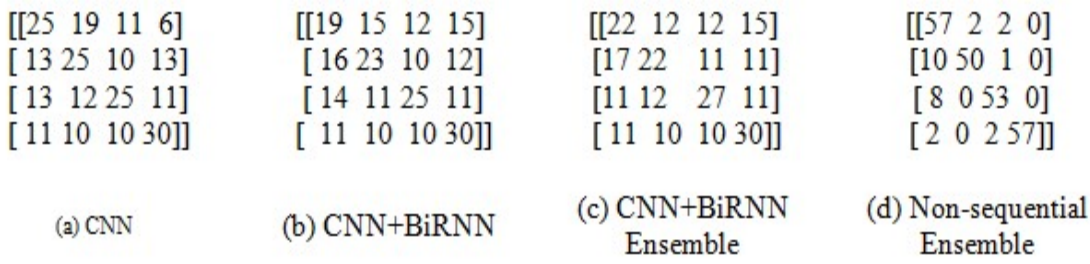


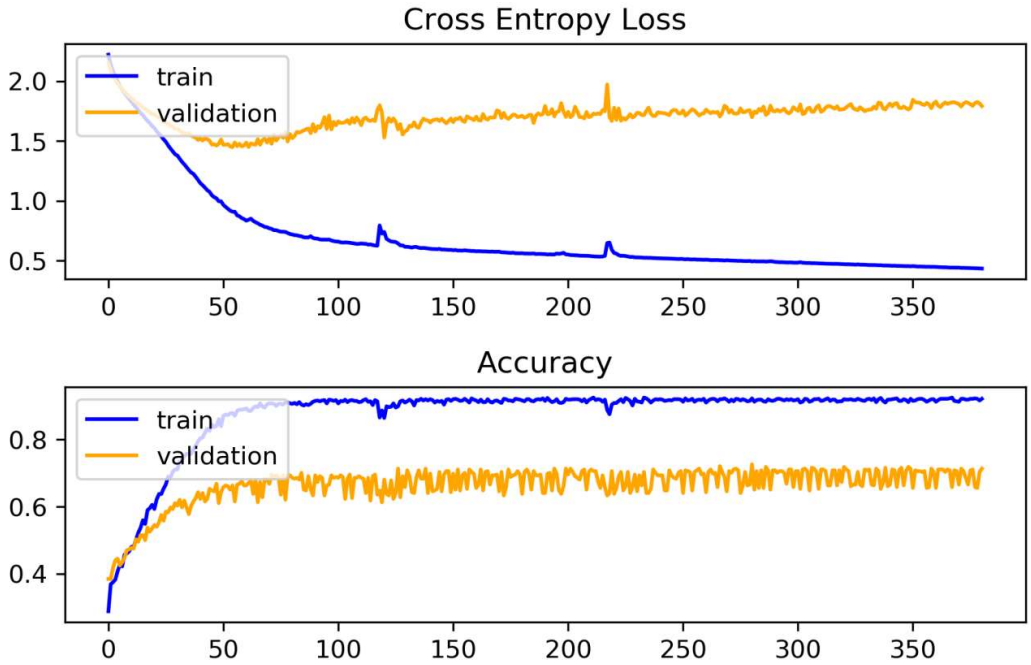
Figure 72: Sample confusion matrices produced by different models used in the study

Model	Validation Accuracy	F1-Score	Cohen's Kappa	ROC AUC
Non-sequential CNN Model Ensemble	0.91±0.001	0.9±0.004	0.84±0.005	0.97±0.002
Sequential CNN+BiRNN Model Ensemble	0.69±0.01	0.68±0.03	0.62±0.01	0.76±0.02
Sequential CNN+BiRNN Model	0.62±0.02	0.6±0.01	0.59±0.03	0.73±0.02
Sequential CNN Model	0.59±0.03	0.56±0.02	0.57±0.01	0.69±0.01

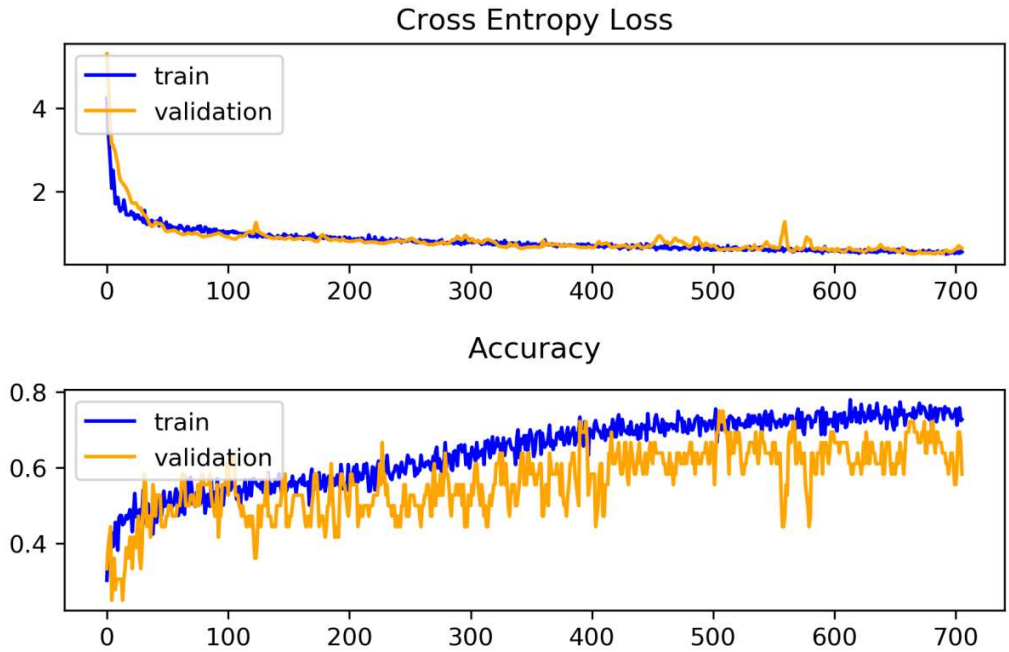
Table9: Average results (with standard deviations) of different models evaluated by various metrics

From the average results tabulated (Table 9), it may be observed that the validation accuracy of the non-sequential model is also on the higher side along with a high F1-score and ROC-AUC score. As accuracy may get inflated as a result of pre-processing bias, often precision and recall are considered to measure the false positives and false negatives, respectively. The problem is that both precision and recall come at each other's cost and can't have arbitrarily high value at the same time. Thus, if the weighted average of precision and recall, i.e., F1-score is high, it may be concluded that the performance of the model is free from type-I and type-II errors to a significant extent. Moreover, a high ROC-AUC score and kappa statistics confirm the high efficacy of the model by depicting the higher true positive rate over lower false-positive rate with fewer deviations.

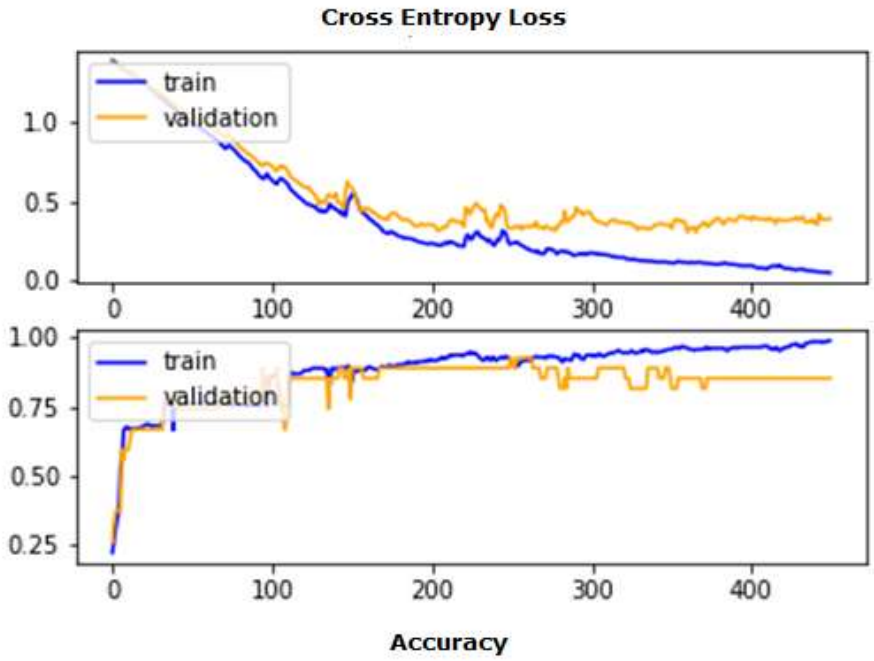
10.2.2 Analysis of Result



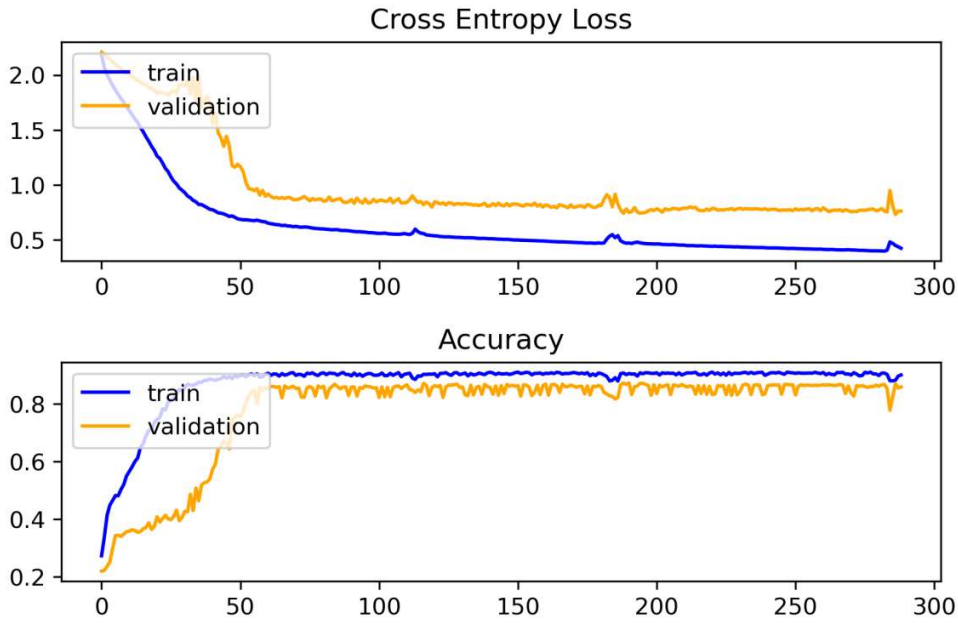
(a) Accuracy and Loss in the CNN model



(b)Accuracy and loss in CNN+BiRNN model



(c) Accuracy and loss in CNN+BiRNN model ensemble



(d) Accuracy and loss in the non-sequential model ensemble

Figure 73: Train and validation curves for AJCC staging by different two-dimensional CNN models

From figure 73 it may be observed that the validation accuracy of the newly developed model has been always higher than other models used in the study. On the other hand, the cross-entropy loss of the non-sequential model has been always lower than other models. Moreover, in the case of the CNN model, the validation loss tends towards higher value

and uneven fluctuations have also been witnessed. The CNN+BiRNN model or its ensemble has shown a lot of intersections between the train and validation accuracy. These observations show that these models are suffering from the problem of overfitting. The non-sequential model, on the contrary, shows less fluctuation and less intersection both in the case of accuracy and loss. Thus, the newly developed non-sequential model ensemble has a clear edge over other models under consideration.

The sheer accuracy of the non-sequential recurrent model is due to the inception and residual layers which act as efficient pre-processors. The point-wise convolution layer like (1X1) or (3X3) plays the role of cheap filters like the Sobel filter or edge detecting filter, etc. These layers also do not let the important features die out of down-sampling or dimensionality reduction process. Moreover, the bidirectional recurrent layers memorize the past and future incidents. Thus, the important features never get lost during the training or the validation phase. The non-sequential model has also used less number of timedistributed layers than other sequential models. It has also been observed that the memory usage during the execution of the non-sequential model was much less than those of other sequential models. All these observations have made the non-sequential recurrent model a lightweight yet efficient classifier. In the existing literature, it can hardly be observed that a deep learning model has been able to classify such a varied mix of tumor images with high accuracy. Every other model did it within a limited periphery of tumor genre, scanner modality, degree of classification, etc. Thus, none of the existing models could be a match for the newly developed model. The non-sequential model used in the study may further be explored by bringing in several other types of tumors, embedding different meta-learners, experimenting with hyper-parameters, etc.

10.3 Conclusion

In this chapter, the efficiency of both the 1D CNN models and 2D CNN models have been evaluated based on their performances as recorded during various experiments conducted so far. In the case of 1D CNN, the CNN+RNN model, and in the case of 2D CNN, the non-sequential recurrent model ensemble has emerged as the most powerful model in their respective domain. When the hardware resources are scarce, the manual feature extraction and implementation of the 1D CNN model is obvious. Otherwise, the 2D CNN model may be employed to classify tumors.