Hepatocellular Carcinoma (HCC) Liver Cancer prediction using Machine Learning Algorithms

Sanapala Rajesh
Department of Computer Science and Engineering,
National Institute of Technology Meghalaya
Shillong, India
sanapalarajesh04@gmail.com

Nurul Amin Choudhury
Department of Computer Science and Engineering,
National Institute of Technology Meghalaya
Shillong, India
nurul0400@gmail.com

Soumen Moulik
Department of Computer Science and Engineering,
National Institute of Technology Meghalaya
Shillong, India
mouliksoumen@gmail.com

2020 IEEE 17th India Council International Conference (INDICON) | 978-1-7281-6916-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/INDICON49873.2020.9342443

***Abstract*—In this paper, we have discussed regarding the prediction model of Hepatocellular carcinoma (HCC) liver cancer. As it is one of the most common types of liver cancer which takes the lives of thousands of people around the world, we have proposed a straight-forward approach of predicting the HCC liver cancer using publicly available dataset. Different types of data pre-processing are done on the dataset for extracting the most optimal information, and also different types of classification models (Like KNN, Decision Tree, etc.) are used in our experiment to see the prediction results. We achieved good results in terms of accuracy and computational time.**

***Index Terms*—*Hepatocellular Carcinoma (HCC)*, *Hepatic*, *Hepatomas*, *Hepatitis-B*, *Hepatitis-C*, *Hepatocytes*, *Machine Learning*, *Ordinal*, *Nominal*, *Confusion Matrix*.**

I. Introduction

Hepatocellular carcinoma, aka HCC is one of the most common types of primary liver cancer. It is also popularly known as hepatomas or hepatic tumours. HCC is mostly caused by the infection of the Hepatitis-B (DNA Virus) or Hepatitis-C (RNA Virus) virus or due to excessive alcohol intake. Many everyday consumables like Azo-dyes (e.g. Food Additives), pollutants such as pesticides and insecticides, parasitic infestations like clonorchiasis also cause HCC.

There are three types of HCC which are known as-

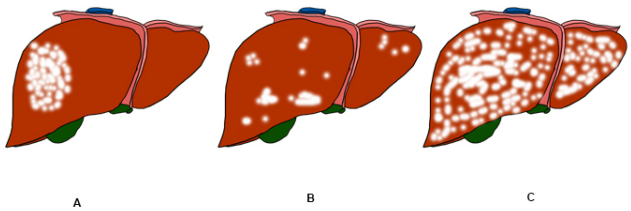


Fig. 1: Types of Hepatocellular carcinoma (HCC) cancer * (A) Expanding type, (B) Multifocal type, (C) Infiltrating type

- **Expanding type:** It is the most frequent or standard type of HCC where it forms a single, yellow-brown, large mass, most often in the right lobe of the liver with central necrosis, haemorrhage and occasional bile staining.
- **Multifocal type:** It is a less often type of HCC where multifocal, multiple masses (3-5 cms in diameter), scattered throughout the liver like tumours are seen.
- **Infiltrating type:** It is a rare type of HCC where the HCC tumour cells are diffused in the whole area of the liver.

Some of the major factors of HCC are *Genetic factors, Age, Gender, Chemicals, Hormones and Nutrition.* If we see the HCC cancer microscopically, the tumour cells in the standard HCC resembles the hepatocytes but vary in different degree of differentiation. Most of the HCC has the trabecular growth pattern, and the tumour cells tend to penetrate and grow along with the blood vessels. Like other cancer diseases, HCC also grows slowly in different stages, and if it is detected in the early stage we can cure it very efficiently.

Machine Learning is the branch of computer science in which the machine learns forms present data and try to predict an optimal outcome from those data. Many researchers categorise machine learning based on three steps-

1) Training or Learning from past data.
2) Continue to carry out tasks like classification, prediction, etc.
3) Boost the performance based on the experience gained from the past and present data.

For our experiment, we will be using supervised machine learning algorithms to perform the HCC predictions. The used ML algorithms will be described in the remaining sections below.

The work which will be described in this paper is to predict the HCC cancer for the different subjects so that the proper precautions can be taken by them.

978-1-7281-6916-3/20/$31.00 ©2020 IEEE

II. Related Works

The authors in [1] try to improve the different disease prediction by using Machine Learning and Genetic algorithms. They have used the dataset which contains all the information of different patients. Their dataset consists of both structured and unstructured data. The genetic algorithms were used to trouble the missing information in the dataset, and by using the RNN, they extracted the necessary features from unstructured data. Then by using the different ML algorithms like Support Vector Machine (SVM), Naive Bayes and K- nearest neighbours (KNN) they calculated the accuracy of the system.

In [2] the authors proposed a diabetes disease prediction using data mining. They used a very simple and straightforward approach for predicting the diabetes of a person using two classification algorithms i.e. KNN and Naive Bayes. Their experiment starts with feeding the patients data into a system by the admin with their privacy maintained. They managed to collect 2000 diabetic patient data. When the data is collected, then the admin will choose an appropriate classification algorithm between KNN and Naive Bayes and the prediction is performed. They achieved good prediction outcome, but because of the fewer number of records, the accuracy was only feasible for KNN and Naive Bayes classification algorithm. In future, they will try to increase their accuracy and efficiency of their proposed model by generating more data.

In [3] the authors use different kinds of machine learning algorithms like support vector machine, random forest, logistic regression, Decision tree, etc. and various types of disease dataset to show the application of ML in disease prediction. They also followed the traditional way of performing the classification using the pre-processing, feature selection, model training and testing for producing the results. They used feature extraction for decreasing the computational overhead. Also, to get the most optimal outcome, they divided every dataset into 90:10 ratio for training and testing.

III. Proposed Model

To perform our experiment, we have used a publicly available dataset from the UCI machine learning repository named as HCC STo perfoem our experimenr we have used the publicly available dataset from Kaggle name as HCC Dataset [4]. This dataset was generated from 165 clinical patient data who were suffering from HCC disease at the University Hospital in Portugal. The dataset contains the 49 features which are recommended by the *European Association for the Study of the Liver - European Organisation for Research and Treatment of Cancer* (EASL-EORTC). Following are the detailed steps which we have used for the experiment.

A. Data Pre-processing

The dataset, which is mentioned above, has faults and have scattered data. To make the dataset useful and extract the information from it, we have performed pre-processing. To handle faulty data, we have analyzed the dataset for the unusual entries and corrected them manually. Missing values are handled with the help of calculating the median of that particular feature and assigning it to the missing spaces. In order to make the dataset useful, we have used Pandas [5] and NumPy [6] library for handling the dataset scatteredness and easy data handling throughout the experiment.

B. Exploring the Dataset

In the second phase, we tried to understand all the features which are present in the HCC survival dataset. There are a total of 49 features/attributes which are present in the dataset in which 23 attributes are quantitative, and 26 are qualitative variables. The target class is a binary variable in which 0 (Dies) and 1 (Lives) as per the assessment of 1-year outcome. Table I shows the type of features which are there in HCC dataset.

TABLE I: Types of feature present int the dataset

Sl. No.	Type of Attribute	Attributes/Features
1	Nominal	Gender, Symptoms, Alcohol, Hepatitis B, Surface Antigen, Hepatitis B e Antigen, Hepattitis B Core Antibody, Hepatitis C Virus, Antibody, Cirrhosis, Endemic, Countries, Smoking, Diabetes, Obesity, Hemochromatosis, Arterial Hypertension, Chronic Renal Insufficiency, Human Immunodeficiency Virus, Non-alcoholic Steatohepatitis, Esophageal Varices, Splenomegaly, Portal Hypertension, Portan Vein Thrombosis, Liver Metastasis, Radiological Hallmark
2	Integer	Number of Nodules, Age at Diagnosis
3	Continuous	Grams of Alcohol per day, Packs of Cigarettes per year, International Normalised Ratio, Alpha-Fetoprotein (ng/mL), Haemoglobin (g/dL), Mean Corpuscular Volume (fl), Leukocytes (G/L), Platelets (G/L), Albumin (mg/dL), Total Bilirubin (mg/dL), Alanine transaminase (U/L), Aspartate transaminase (U/L), Gamma glutamyl transferase (U/L), Alkaline phosphatase (U/L), Total Proteins (g/dL), Creatinine (mg/dL)
4	Ordinal	Performance Status, Encephaalography degree, Ascites degree

C. Setting Classification and Accuracy Matrices

To classify the disease, we need to set some metrics which will help us in predicting the HCC disease. As we are using scikit-learn machine learning library [7] for our experiment, we have used confusion matrix as the classification measure matrix. All the used metrics i.e. *Precision, Recall, F1-Score and Accuracy* in our experiment are listed below.

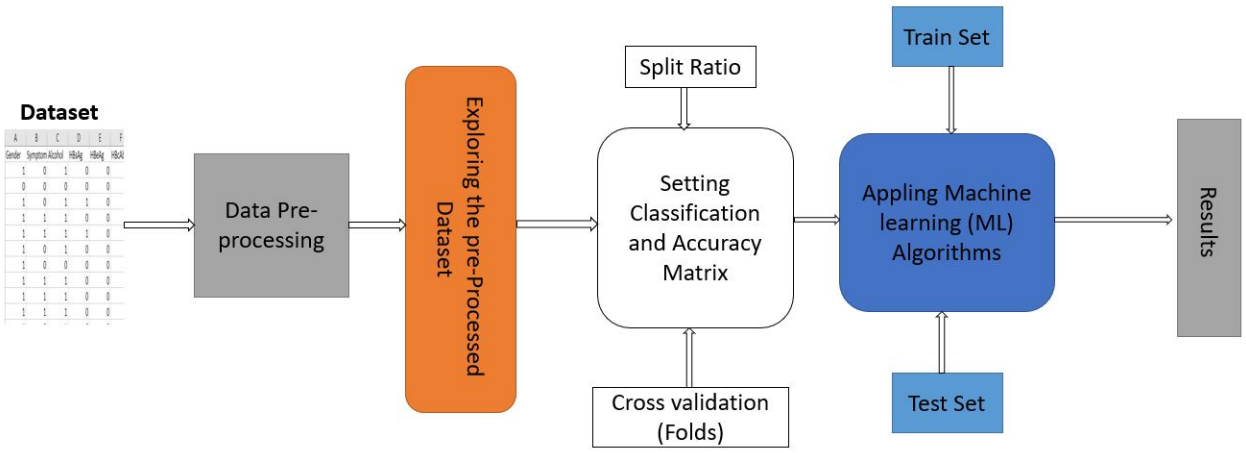


Fig. 2: Architecture of Proposed System

- Precision (P) is defined as the number of true positives (T_p) over the number of true positives plus the number of false positives (F_p). Mathematically,

$$P = \frac{T_p}{T_p + F_p} \tag{1}$$

- Recall (R) is defined as the number of true positives (T_p) over the number of true positives plus the number of false negatives (F_n).

$$R = \frac{T_p}{T_p + F_n} \tag{2}$$

- F1-Score $(F1)$ is defined as the harmonic mean of precision and recall.

$$F1 = 2 * \frac{P * R}{P + R} \tag{3}$$

- Accuracy (A) is defined as follows.

$$A = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{4}$$

Model training is an essential step for achieving good results and for avoiding the model overfitting and model underfitting. In order to prevent the model overfitting and underfitting, we have chosen the 90% of our data as training data and the remaining 10% as testing data. This 90:10 ratio of train and test set will be adequate for achieving good classification accuracy as well. At last, we have used another evaluation method, namely k- Fold cross-validation for precise use of dataset and also for calculating most optimal accuracy results [8]. The final step in our experiment is to apply different classifiers or ML algorithms to see the achieved prediction. We have used five machine learning algorithms which are *K-Nearest Neighbor (KNN), Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), Vector Machine (SVM)*. The entire flow or architecture of our system can be seen in Fig. 2 above.

IV. Experimental Results

We have used python 3.7 version with scikit-learn [7] for our result analysis. We have chosen different classifiers to get the variation in results and also to see how the dataset is performing on different classifiers. The Table II shows the result values of different performance metrics for each classifier that we have used. Now, with the help of accuracy-score from scikit-learn, corresponding accuracy were also calculated using Eq. 4. Fig. 3. represents the accuracy of each classifier, achieved by our experiment without applying cross-validation.

TABLE II: Confusion Matrix and Performance Metrics

Classifier	TP	FP	FN	TN	P	R	F1
KNN	27	6	7	22	0.79	0.82	0.81
NB	5	28	3	26	0.62	0.15	0.24
DT	23	10	9	20	0.72	0.70	0.71
RF	28	5	7	22	0.80	0.85	0.82
SVM	26	7	14	15	0.65	0.79	0.71

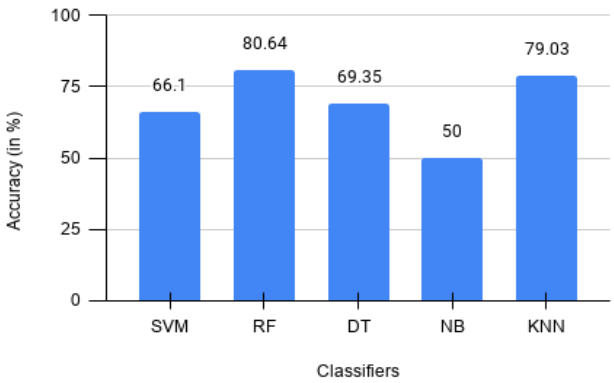
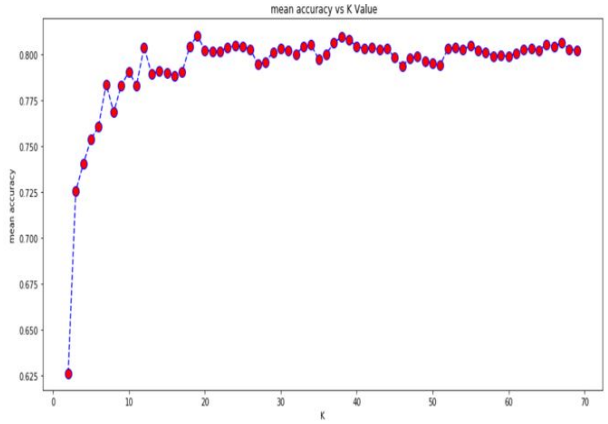
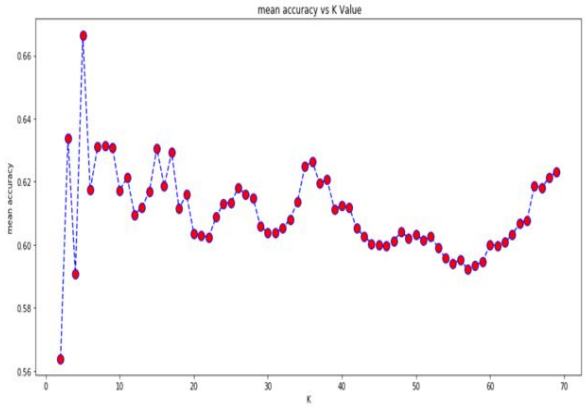


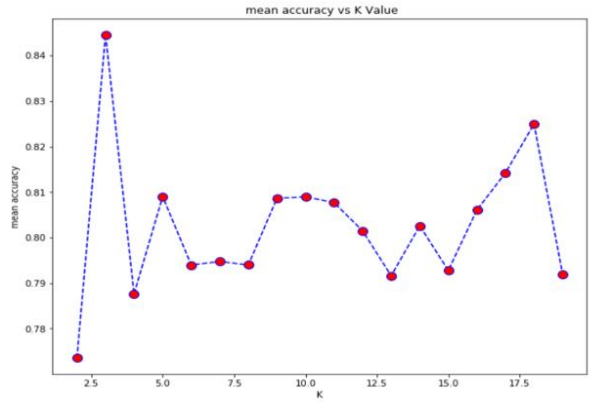
Fig. 3: Accuracy graph without cross-validation



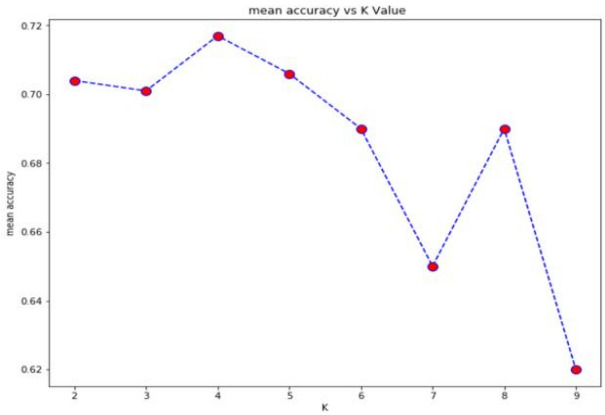
(a) Mean accuracy vs different values of k in k-Fold cross validation of KNN classifier



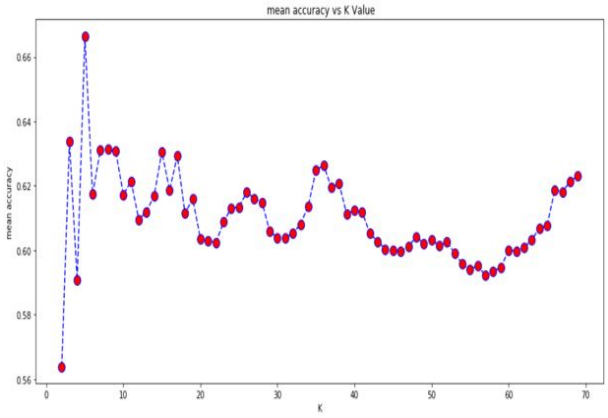
(b) Mean accuracy vs different values of k in k-Fold cross validation of Decision Tree classifier



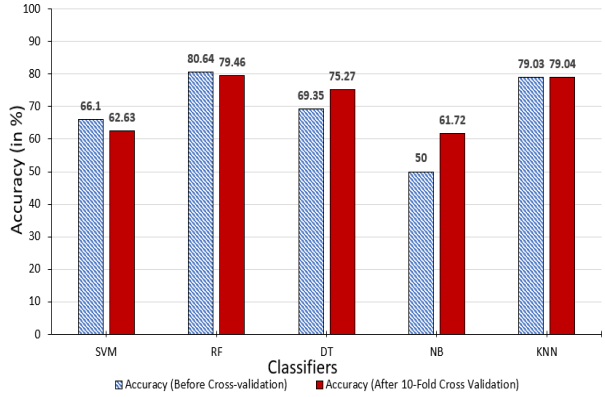
(c) Mean accuracy vs different values of k in k-Fold cross validation of Random Forest classifier



(d) Mean accuracy vs different values of k in k-Fold cross validation of Support Vector Machine classifier



(e) Mean accuracy vs different values of k in k-Fold cross validation of Naive Bayes classifier



(f) Accuracy comparison graph for different classifiers with and without cross-validation.

Fig. 4: Mean Accuracy vs K- Fold valuess graph of different classifiers used in our experiment and Accuracy comparison graph.

Then, we did the cross-validation using k- fold cross validation and we calculated the mean accuracy by taking the average of all accuracies which we have achieved from different classifiers. To get the visualization of our mean accuracy with respect to the "k" values, we have plotted a *Mean Accuracy* vs *K value* graph as shown in Fig. 4. In order to see the efficiency of our proposed experiment, we have plotted and accuracy comparison graph between the accuracies which were achieved before cross-validation and after cross-validation. In Fig. 4(f), we can see that

TABLE III: Computational time (In Sec.) of different classifiers with and without Cross- validation (CV)

Classifier	C_t without CV	C_t with CV
KNN	0.013	0.045
NB	0.022	0.053
DT	0.026	0.057
RF	1.7	17.8
SVM	59	1901

after applying cross-validation and taking the mean accuracy in consideration, we have managed to achieve good and stable accuracy with most of our classifiers. Among all the classifiers Random forest manages to achieve the highest accuracy of 79.46% with cross-validation and also KNN manages to achieve good accuracy of 79.04% with cross-validation as well.

Finally, we will see the computational time of all the classifiers in our experiment with and without cross-validation. Computational time is calculated with the help of training and testing time of each classifier. The time required for training (learning from the given data) a classifier is known as the training time. On the other hand, the time needed for testing (checking the results by cross verifying the new data to trained data) is termed as testing time. Computational time is calculated as per the following equation -

$$C_t = T_t + T_s \tag{5}$$

where T_t and T_s represents training time and testing time, respectively. In Table III we can see that the SVM classifier has the highest computational time, which directly implies high processor and memory usage. Also, it is not giving good accuracy as compared to other classifiers like Random Forest, KNN and Decision Tree, which has very less computational time. In practice, we can say that the classifiers which are providing better accuracies have less computational time. Because of this, we can say that proposed system is efficient in terms of computational time.

V. Conclusion

In this paper, we have predicted the Hepatocellular Carcinoma cancer (HCC) disease outcome for the patient, whether they die or lives with the provided information as features in the dataset with success. We managed to achieve the highest accuracy of 80.64% with Random Forest classifier without applying cross-validation and a feasible accuracy with all other classifiers by employing 10- Fold cross-validation. In future, we will try to extract for features from the available dataset and try to increase the achieved accuracy. Also, we will try to apply for more Machine learning algorithms and deep learning approaches like ANN, CNN, to get the most optimal results.

References

[1] S. Singh and D. Hanchate, "Improving disease prediction by machine learning," 06 2018.

[2] D. Shetty, K. Rit, S. Shaikh, and N. Patil, "Diabetes disease prediction using data mining," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2017, pp. 1–5.

[3] P. S. Kohli and S. Arora, "Application of machine learning in disease prediction," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, 2018, pp. 1–4.

[4] M. Santos, P. Henriques Abreu, P. García-Laencina, A. Simao, and A. Carvalho, "A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients," *Journal of biomedical informatics*, vol. 58, pp. 49–59, 10 2015.

[5] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, Eds., 2010, pp. 56 – 61.

[6] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, p. 28252830, 2011.

[8] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, 2016, pp. 78–83.