*FUDMA Journal of Sciences (FJS)*
*ISSN online: 2616-1370*
*ISSN print: 2645 - 2944*
*Vol. 5 No. 2, June, 2021, pp 447-455*
*DOI: https://doi.org/10.33003/fjs-2021-0502-649*

# HYPER-PARAMETER OPTIMIZATION AND EVALUATION ON SELECTED MACHINE LEARNING ALGORITHM USING HEPATITIS DATASET

**YUSUF Aminat Bolatito*, AKANDE Oyelola**

Department of Information and Communication Technology, Usmanu Danfodiyo University, Sokoto, Nigeria
*Corresponding Authors Email: aminbolly@gmail.com

**ABSTRACT**
Despite the popularity and utility of most machine learning techniques, expert knowledge is required in guiding choices about the suitable technique and settings that are good for solving a specific problem. The lack of expert information renders the procedures vulnerable to poor parameter settings. Several of these machine learning techniques configurations are offered under default settings. However, since different classification problems required suitable machine learning techniques, selecting the appropriate technique and tuning its settings are vital works that will rightly improve predictions in terms of reliability and accuracy. This study aims to perform grid search parameters tuning on 5-selected machine learning techniques on hepatitis disease. Comparative performance is drawn side-by-side with the default settings. The experimental results of the five tuning techniques show that using the configurations suggested in our work yield predictions of a greatly sophisticated quality than choice under its default settings. The result proves that tuning parameters of Support Vector Machine via grid search yields the best accuracy outcomes of 90% and has a competitive performance relative towards criteria of precision, recall, accuracy and Area Under the Curve. Present combinations of parameter settings for each of the techniques by identifying ranges of values for each setting that give good Hepatitis disease outcomes.

**Keywords**: Hepatitis disease, Grid Search, Parameters Tuning, Machine Learning Techniques

## INTRODUCTION

Machine Learning (ML) is a subfield of Artificial Intelligence that can learn from datasets, different from medical sciences (Parisi, 2014) (Parisi et al., 2015) through understanding the underlying patterns of any disease to guarantee that projections about the course of a disease are more accurate (Cequera & García, 2014). Researchers have proved the potential of machine learning algorithms in medical areas to develop predictive models due to easy access to clinical data (El-Salam, 2019), and several ML techniques have been implemented for the diagnosis and prediction of various diseases in this discipline. Therefore, to identify hepatitis disease and make effective decision ML techniques will play a major role. Various practices are used on ML before applying in the diagnosis of any disease (Yarasuri et al., 2019). Among them is the proper analysis of data with seasoned competence and the selection of the suitable parameters of techniques (Yarasuri et al., 2019). This work, presents parameters optimization methods for five-selected best machine-learning algorithm of K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Logistic Regression, Gaussian Naïve Bayes and Decision Trees tested on hepatitis dataset. Herein, the five ML techniques were selected for this studied since they are the most prevalent algorithms for diagnosing disorders using medical datasets (Parisi, 2014) and is widely used for the early detection of liver disorders, with a track record of success (Taradeh et al., 2019).

In the settings of suggested ML algorithms, the notion of grid search strategy was utilized. It is viewed as a distinct approach of obtaining the best parameter for any model in order for the classifier to accurately predict unlabeled data, i.e. testing data. (Ramadhan et al., 2017). The method is classified as an exhaustive approach for determining the optimal parameter values, and it must be investigated for each sorted group of prediction values. The performance of the improved algorithms was further evaluated using accuracy, precision, sensitivity, specificity analysis, and Area Under the Curve (AUC).

### Related Works

Machine learning has piqued the interest of many scholars and has been used in a variety of fields around the world. Machine learning has proved its potential in medicine, where it has been used to handle disease classification difficulties. Hepatitis is a life-threatening condition that can be fatal if not detected and treated early. As a result, researchers must use several classification strategies to identify examine the disease, as well as combine the model with parameter optimization settings.

Four distinct machine learning techniques were used to predict whether a patient would live or die after contracting Hepatitis C. (Bhargav & Kumari, 2018). Logistic regression exceeds the other techniques with an accuracy of 87.17 percent, as demonstrated in the result using the default parameter settings. In a similar line, (Yarasuri et al., 2019) looked at machine learning algorithms including SVM, KNN, and Artificial Neural Network as prediction tools for diagnosing Hepatitis C disease. It was concluded that, of all the models considered, ANN is the most accurate, with a prediction accuracy of 96%. Using categorization algorithms, (Vijayarani & Dhayanand, 2015) forecast liver disorders. Naive Bayes and SVM were the algorithms employed. The SVM classifier is regarded as the best algorithm based on the experimental findings, according to the work.

On several datasets, (Ambesange et al., 2020) provided a feasible method for liver disease diagnosis. The work selects

features, removes outliers, and then applies grid search tuning to the KNN classification model. The results demonstrate that the model works better, with a 91 percent accuracy. (Shekar & Dagnew, 2019) proposed utilizing a grid search strategy to enhance hyperparameters of a random forest tree to diagnose microarray cancer. The random forest technique is optimized to obtain the best parameters, which are then used to validate the method. The proposed method provides the optimal settings for splitting a node with the maximum number of features, the number of decision trees in a forest, the depth of the trees as well as criterion to split a given node into a child node. The work used the grid search method to tune parameters in the well-known classification algorithm random forests (Ramadhan et al., 2017). Random forest was used to detect gender based on the characteristics of the human voice on the voice gender dataset. To get the best results, two settings were tweaked. The best accuracy of random forest with parameter tuning is 0.96907, which is higher than the accuracy of the model without parameter tuning, according to experimental results on the voice gender dataset (0.9675). The tuning parameter produces appropriate parameters for producing the best classifier, according to this study.

Following the observation of (Ramadhan et al., 2017) (Bhargav & Kumari, 2018) on the same subject. The remainder of the paper is laid out as follows. The dataset description, proposed classification strategies, and evaluation are all covered in Section 3. The analysis and discussion of the findings are presented in Section 4. Finally, in the last section, conclusions are drawn.

## MATERIAL AND METHODS

This section is divided into four subsections. First, go over the detail of the data. Second, the preparation and data restoration methods utilized before applying ML techniques are presented. Then fundamental concepts of the various ML approaches used in this work are presented. Finally, the fundamental functioning and derivation of the various performance measures are discussed. The structure for how the suggested method finds the best settings is shown in Fig 1.

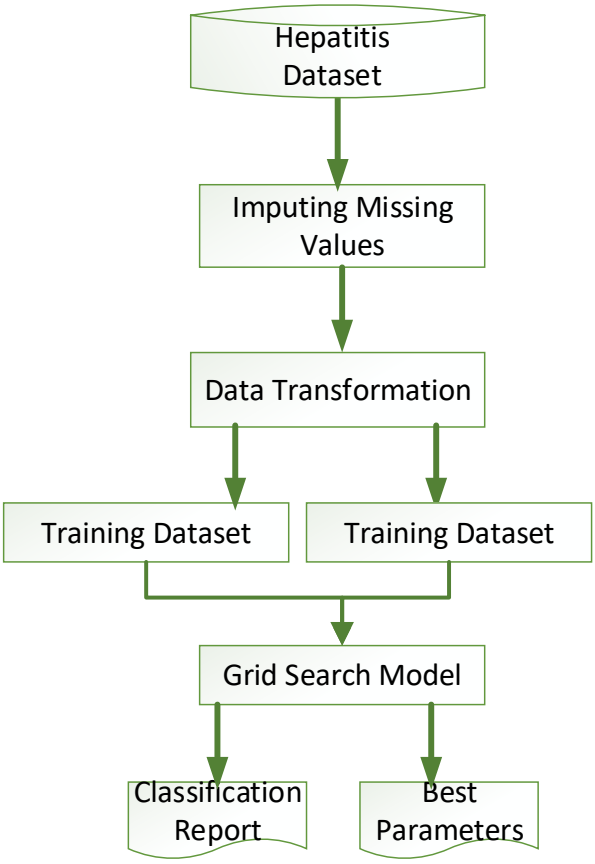


*Figure 1: Proposed Model for Optimal Search*

## DATASET DESCRIPTION

Any disease prognosis requires a great deal of data. Hepatitis data was gathered from the UCI Machine Learning repository. The dataset has 155 entries with 20 features and the class label. It contains 32 and 123 entries indicating whether the patient will live or die as a result of the sickness.

**Table 1: Shows the characteristics in-depth as well as the types of characteristics.**

Table 1: Data Features and Types

| FEATURES | VALUES |
|---|---|
| Class | Integer |
| Age | Integer |
| Sex | Integer |
| Steroid | Integer |
| Antivirals | Integer |
| Fatigue | Integer |
| Malaise | Integer |
| Anorexia | Integer |
| Liver_big | Integer |
| Liver_firm | Integer |
| Spleen_palable | Integer |
| Spiders | Integer |
| Ascites | Integer |
| Varices | Integer |
| Bilirubin | Float |
| Alk_phosphate | Float |
| Sgot | Float |
| Albumin | Float |
| Protime | Float |
| Histology | Integer |

**Data Preprocessing and Transformation**

Machine learning approaches for prediction and diagnosis rely heavily on the quality of data. The inaccurate prediction may result from the quality of data used. As a result, the dataset quality and pre-processing techniques have a substantial influence on the performance and accuracy of the predictive model.

The preprocessing stage was used to sanitize the data to make it fit for an accurate prediction. The cleaning procedure eliminates noisy data, manages null or missing values, and verifies field data types. Various approaches for addressing missing values datasets have been proposed in the literature; Mean imputation was used to replace missing values in the"Bilirubin", "Alk phosphate", "Sgot", "Albumin", and "Protime" datasets because the datatypes of these inputs are numeric. Finally, the values were rounded up to the nearest integer. The missing values for "Steroid", "Fatigue", "Malaise", "Anorexia", "Liver big", "Liver firm", "Spleen palable,", "Spiders", "Ascites" and "Varices" were then imputed using median imputation because they contain categorical values.

Finally, all input characteristics were randomized and normalized to reduce values to a narrow range, in this instance [0,1], to appropriately fit the classifier (see Equation 1).

$$\sigma = \frac{X - Min(X)}{Max(X) - Min(X)} \ldots \quad \ldots \quad \ldots \text{ equation 1}$$

Where X is the original feature value and $\sigma$ is the normalized value of the features. These parameters can impact the quality, predictive power, and accuracy of our model's performance if they aren't handled properly. The data was divided into two categories: training and testing. The classifier was then tuned with tuning parameters, and 10-fold cross-validation was performed to fit it to the training samples. The findings of the grid search were applied to the test samples.

**The Proposed predictive models**

The proposed strategies for categorizing the hepatitis dataset are presented in this section. In machine learning, optimizing the model parameters is critical for constructing an accurate prediction. Grid search identifies combinations of best parameter values, resulting in superior accuracy performance measures. To predict hepatitis diseases, tuning parameters were used to ML algorithms of SVM, Gaussian Nave Bayes, Logistic Regression, Decision Tree, and KNN classifier. The dataset was split into two parts: training and testing. The training data is fitted to the testing data using a tuning model based on 10-fold

cross-validation. The model's and our approach's performance is then evaluated.

*4.1 K-Nearest Neighbors Algorithm*

The KNN method is a supervised ML method used for classification. For statistical estimation and pattern identification, this approach is widely employed. It records all of its instances and then uses a similarity measure to classify the most recent ones. It does not formulate any prior assumption as a non-parameter technique. KNN has several tuning parameters such as the power parameter ("p"), number of neighbors ("n_neighbors"), algorithm, weight function and so forth. By assessing the similarity between data points, the power parameter is utilized to optimize the needed number of features. The settings of its values range from 1 to 7, where $p = 1$, this is equivalent to using euclidean_distance which is the default setting, $p = 2$ for manhattan_distance and arbitrary p, Minkowski distance is used. When an item is categorized, the number of neighbors determines how many neighbors should be verified. The parameter range is 1 to 30, with 5 being the default value; otherwise, a majority vote of its neighbors will result in a case classification, with the case being allocated to the most frequent class among its K closest neighbors, as determined by a distance role. For candidate prediction, the weight function is utilized to distribute values on how weight should be divided amongst neighbors. The values that are taken into account are uniform and distance. Distance assigns weight points by the inverse of their distance, allowing closer neighbors of a query point to have a higher impact than neighbors further away. Uniform allows all points in each neighborhood to be weighted equally. To choose the best appropriate approach, three closest neighbor algorithms are used: ballTree, KDTree, brute-force search, and auto, for ball_tree, kd_tree, brute, and auto, respectively. When the parameter algorithm is left default, the algorithm 'auto' is chosen as the best-suited algorithm based on the values supplied to match the procedure.

**Naive Bayes Algorithm**

The Naive Bayes (NB) classifier is a straightforward method for predicting the likelihood of candidates. It is based on estimating conditional probability using Bayes' theorem and a strong independent assumption. A Naive Bayes classifier posits that the existence of a class feature is independent of the existence of any other feature. Even if the underlying assumption is false, the Naive Bayes classifier works pretty well (Abd El-Salam et al., 2019). Because of their ease of use and great performance in earlier investigations, the two NB parameters priors and Var_smoothing were chosen in this investigation. Priors represent the prior probabilities of the classes. If this parameter is specified while fitting the data, the prior probability will not be justified by the data. The default value for var smoothing is $1\ 1_{e^{-9}}$, which is the percentage of the biggest variance of all features added to variances for computation stability.

**Support Vector Machine Algorithm**

SVM is used to classify data into linear and non-linear categories. It uses non-linear mapping to transfer the original training data to a higher dimension. It looks for the linear optimum separating a hyperplane within this new dimension. With an adequate nonlinear mapping to a sufficiently high dimension, data from two classes may be separated by a hyperplane. SVM locates this hyperplane using support vectors and margins. The classification job is carried out using SVM by maximizing the margin that correctly identifies both classes while reducing classification mistakes (Abd El-Salam et al., 2019). The SVM model uses several mathematical features or parameters to classify data variables, such as Regularization ('C'), Kernel, and Gamma. The kernel type to utilize in the algorithm is specified by the kernel. Linear, poly, rbf, sigmoid, precomputed, or callable values are available. If no value is specified, "rbf" will be used instead. The kernel coefficient for "rbf", "poly", and "sigmoid" is gamma, and the range of values is scale and auto, with the scale being the default. The regularization parameter is used to determine how important misclassifications are. SVMs provide a quadratic optimization issue in which the goal is to maximize the margin between the two classes while reducing the number of misclassifications. However, to identify a solution for non-separable issues, the misclassification restriction must be loosened, which is done by setting the previously stated "regularization" parameter.

**Logistic Regression Algorithm**

Logistic regression is a statistical methodology for analyzing a data set in which one or more independent factors predict an outcome. When there are only two possible outcomes, a binary variable is used to assess the outcome. The purpose of logistic regression is to identify the best-fitting model to represent the connection between a set of independent (predictor or explanatory) factors and a dichotomous feature of interest. The solver and the maximum number of iterations were the two Logistic regression parameters employed in this investigation. The algorithm used in the optimization problem is known as the solver. The selection on the algorithm are: "newton-cg", "sag", "saga", "liblinear", and "lbfgs". The number of iterations required for the solvers to converge is specified between 100 and 500.

**Decision Tree Algorithm**

The decision tree (DT) algorithm, the final model, is one of the most used techniques for creating a tree-structured model (Pilz, 2018). Cases are divided into groups using this approach, or the values of a target variable may be predicted using the values of predictor variables (Ginde et al., 2009). The root, internal, and end nodes are the three types of nodes in this method. This strategy makes the nodes repeatedly create homogeneous subgroups according to the class label (Lanham-New et al., 2011). Features having the best rate of splitting criteria remain in the model. As splitting criteria, the Gini index and entropy is utilized. A set of features is determined by decision nodes. Finally, all of the leaves are made. The maximum depth of the tree, the number of features to examine while looking for the best split, the minimum number of samples necessary to divide an internal node, and the criterion used for splitting are the primary parameters that are optimized. A set of features is determined by decision nodes. The Gini-Index and entropy are utilized as the dataset's dividing criterion. Furthermore, log2, auto, and sqrt are among the properties examined for the number of features when splitting a node. The maximum depth

and the minimum number of samples required for splitting are integer values between 10 and 200, and 1 and 20, respectively.

**The evaluation Parameters**

Data were segmented into 10-fold cross-validation training and testing data sets before modeling.
The scores collected for each fold are averaged out and utilized as a single score after a 10-fold cross-validation repeat. This means that the model is trained with 90% of the data for each fold and evaluated against the remaining 10%. This style's cross-validation avoids the bias of training the model primarily on negative or positive data. (Chown, 2019).
To assess a model's performance, one must determine how effectively it can predict unknown data, and so the quality of the prediction is assessed using the criteria. When measuring the predictive potential of algorithms, it is vital to measure model performance correctly. A technique that is agnostic to the algorithm is designed since performance measurements across different algorithms must be comparable. In some cases, the label returned a single class for each occurrence in the prediction set (e.g., regression), while in others, a vector expressing the probabilities for each occurrence must be allocated to a specific class (Bermúdez-Chacón et al., 2015). There are a variety of measures for comparing anticipated and predicted classes, and different metrics allow for the examination of different prediction qualities. The output of these numerous metrics can have a wide range of values, and the interpretation of the actual numbers can change from one statistic to the next. For example, the accuracy of a forecast is assessed on a scale of 0 to 1, with a higher number indicating a better forecast. The accuracy and reliability of the proposed classifier were employed as the primary criterion for evaluating its prediction accuracy and validating its robustness (Luca Parisi & RaviChandran, 2020). Accuracy is a measure of the classifier's prediction capacity to distinguish between the two classes of interest (Luca Parisi & RaviChandran, 2020). Reliability is also defined as the classifier's ability to assign an acceptable level of certainty to the prediction outcome, as evaluated by the sensitivity (SN) (Luca Parisi & RaviChandran, 2018), specificity (SP) (Luca Parisi et al., 2018), and, more broadly, the area under the Receiver Operating Characteristic (ROC) curve or simply Area Under the Curve (AUC). By adjusting the discriminating threshold for prediction values, ROC is a valuable tool for assessing the robustness of a model. This gives more information than accuracy. As a result, the following measures are proposed in this study: accuracy receiving-operator characteristic (ROC) curves (Metz, 1978), precision, and recall (Chown, 2019). True positive (TP), false positive (FP), true negative (TN), and false negative (FN) values are generated as a consequence of the analysis. The x-axis of ROC curves represents FP rates, while the y-axis is TP rates. The following are the definitions:

a. True Positives (TP) occur when the data point's actual class was true and the forecasting class was true as well.
b. True Negatives (TN) occur when the data point's actual class was false and the forecasting class was likewise false.
c. False Positives (FP) occurs when the data point's actual class was False but the forecasting class was true. False because the model anticipated the wrong thing, and true because the class anticipated the right thing.
d. False Negatives (FN) are when the data point's actual class was true but the forecasting class was False. False because the model anticipated the wrong thing, and negative because the class predicted the wrong thing.

The True Positive Rate (TPR), also known as sensitivity or recall, is the likelihood that a test result will be positive for the disease. Equation 2 illustrates this.

$$TPR = \frac{TP}{(TP + FN)} \ldots \quad \ldots \quad \ldots equation\ 2$$

The Specificity is defined as the True Negative Rate (TNR). Equation 3 expresses the chance that a test result will be negative when the disease is not present.

$$TNR = \frac{TN}{(TN + FP)} \ldots \quad \ldots \quad \ldots \quad equation\ 3$$

Precision, also known as positive predictive value (PPV), is calculated using equation 4.

$$Precision = \frac{TP}{(TP + FP)} \ldots \quad \ldots. \quad \ldots \quad eqation\ 4$$

The accuracy is defined as the likelihood of a total number of correct predictions, as shown in equation 5. A forecast's accuracy is assessed on a scale of 0 to 1, with a higher number indicating a better forecast.

$$Accuracy = \frac{TP + TN}{(TP + FP + TN + FN)} \ldots \quad \ldots \quad \ldots equation\ 5$$

The receiver operating characteristic (ROC) curve is a graph that shows the whole picture of a trade-off between sensitivity, true-positive rate (TPR), and false-positive rate (FPR) through a succession of cut-off points. The x-axis is represented by FPR, while the y-axis is represented by TPR. The area under the ROC curve (AUC) (Kumar & Indrayan, 2011) assesses a diagnostic test's capacity to distinguish between unhealthy and healthy individuals and is regarded as an effective measure of a diagnostic test's inherent validity. It's also utilized to discover the best cut-off point for distinguishing between diseased and non-diseased people. It shows how a binary classifier system performs while the discriminating threshold is changed (Kumar & Indrayan, 2011). When the sum of sensitivity and specificity was highest, the appropriate cut-offs were established.

**RESULTS, ANALYSIS, AND DISCUSSION**

A total of 155 hepatitis patient files were acquired from the UCI archive for this investigation. The dataset was subjected to a variety of machine learning methods. KNN, SVM, Logistic Regression, Decision Tree, and Gaussian Naive Bayes were the ML algorithms used on the datasets. The study compares the performance metrics obtained using the default parameter to those acquired using enhanced hyperparameters in predicting hepatitis illness. In general, the study of the metrics reveals that optimization has a significant impact on model performance. Furthermore, when comparing the optimized models to their

default counterparts, there was a greater discrepancy in model performance across tuning parameters in the optimized models. The optimized models' performance metrics are much higher than the default settings' performance metrics. Only a few models performed better than the default settings. The best classification result was obtained in these circumstances by modifying the support vector machine classifier. The different machine learning models and visualizations were created using Python packages such as Scikit Learn's class GridSearchCV and Matplotlib's Matplotlib.

The confusion matrix was used to evaluate the criteria of the default and adjusted optimal models, which are displayed in Tables 2 and 3 correspondingly.
This enables for a visual representation of the model's performance by showing how far the misclassified samples are from the true classes, as well as which degrees are better understood (Stoean et al., 2011).

**Table 2 Confusion Matrix for model default parameters**.

| Models | Predicted Values | | | |
|---|---|---|---|---|
| | TN | FN | TP | FP |
| SVM | 3 | 1 | 7 | 28 |
| GuassianNB | 10 | 11 | 0 | 18 |
| Logistic Regression | 5 | 0 | 5 | 29 |
| Decision Tree | 8 | 4 | 2 | 25 |
| KNN | 3 | 0 | 7 | 29 |

**Table 3: Confusion Matrix for model parameters tuning**

| Models | Predicted Values | | | |
|---|---|---|---|---|
| | TN | FN | TP | FP |
| SVM | 6 | 0 | 4 | 29 |
| GuassianNB | 6 | 2 | 4 | 27 |
| Logistic Regression | 5 | 0 | 5 | 29 |
| Decision Tree | 4 | 3 | 6 | 26 |
| KNN | 8 | 7 | 2 | 22 |

The confusion matrix was used to create the two comparison model tables. True negative values in the models showed instances without hepatitis, whereas true positive rates of values showed proper illness categorization. In this scenario, false negatives predicted readings that indicated the absence of hepatitis, although the individuals were suffering from the condition. When hepatitis is anticipated in a patient, but the patient does not have the disease, the phrase "false positive" is used.
Table 4 shows the accuracy of classifiers as a result of these methods. On the classifiers, 10-fold cross-validation was used. The findings demonstrate that hyperparameter adjustment improves the accuracy of all classifiers, with SVM outperforming all other classification methods on the datasets, with an accuracy of 0.90.

**Table 4: Accuracy of default settings and grid search tuning parameters**

| Algorithms | Accuracy on default settings | Accuracy on tuning parameters |
|---|---|---|
| SVM | 0.79 | 0.90 |
| GaussianNB | 0.72 | 0.85 |
| Logistic Regression | 0.87 | 0.87 |
| Decision Tree | 0.85 | 0.77 |
| KNN | 0.82 | 0.77 |

**Table 5: Precision of default settings and grid search tuning parameters**

| Algorithms | Precision on default settings | Precision on tuning parameters |
|---|---|---|
| SVM | 0.8 | 0.88 |
| GaussianNB | 1 | 0.87 |
| Logistic Regression | 0.85 | 0.85 |
| Decision Tree | 0.93 | 0.81 |
| KNN | 0.81 | 0.92 |

**Table 6: Recall of default settings and grid search tuning parameters**

| Algorithms | Recall on default settings | Recall on tuning parameters |
|---|---|---|
| SVM | 0.97 | 1 |

| | | |
|---|---|---|
| GaussianNB | 0.62 | 0.93 |
| Logistic Regression | 1 | 1 |
| Decision Tree | 0.86 | 0.90 |
| KNN | 1 | 0.76 |

This study used the grid search methodology to tune the parameters of all five algorithms to enhance accuracy, precision, and recall. To achieve ideal classifiers, they all have numerous parameters that must be changed. SVM has increased performance for all metrics, as seen in Tables 4, 5, and 6, when compared to the other algorithms' SVM values (0.90, 0.88, and 1) for accuracy, precision, and recall, respectively. All of the models are better at predicting the absence of hepatitis than the presence of the disease, as evidenced by higher specificity and lower sensitivity.

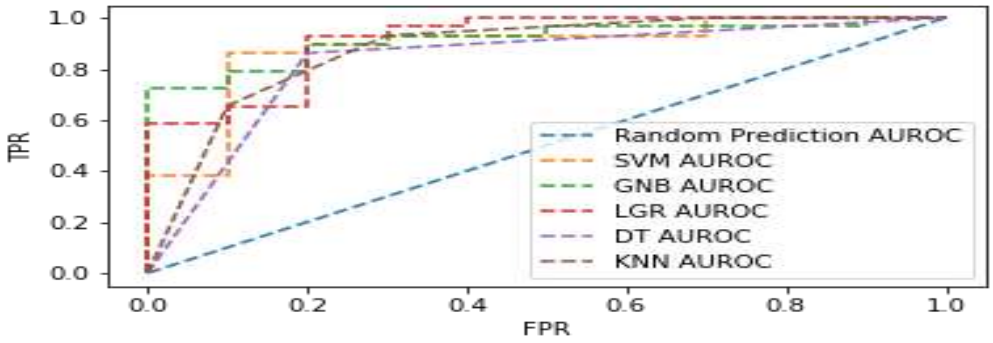


*Figure 2: Default Settings Techniques on Models*

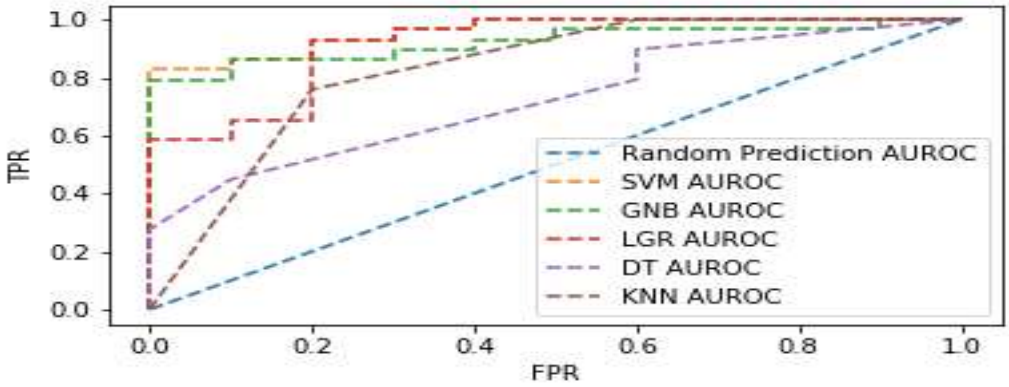


*Figure 3: Grid Search Techniques on Models*

Figures 2 and 3 provide a graph comparing ROC Scores for the 5-selected algorithms while utilizing default settings and adjusted parameters. Figure 3 shows that the AUC for the optimized SVM model outperformed all other optimization models on the validation data set, with much greater and more consistent performance.

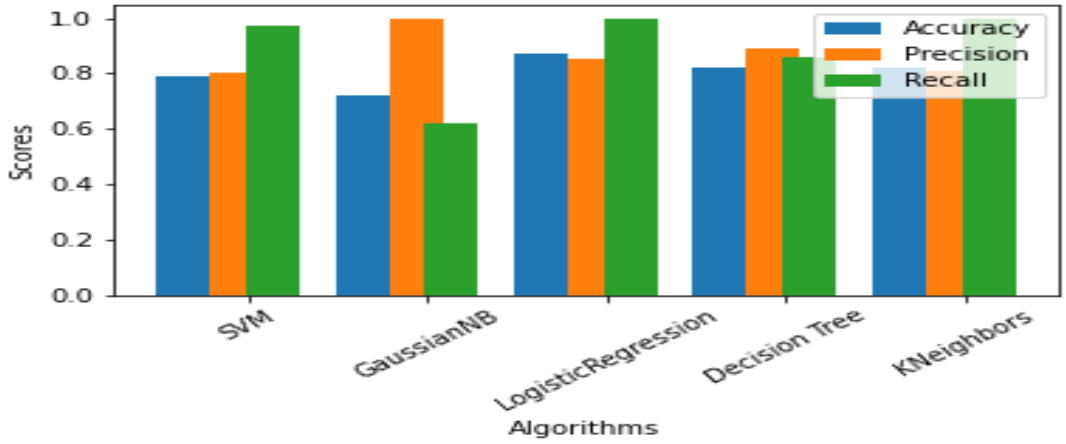


*Figure 4: Performance Metrics Comparison on default parameters*

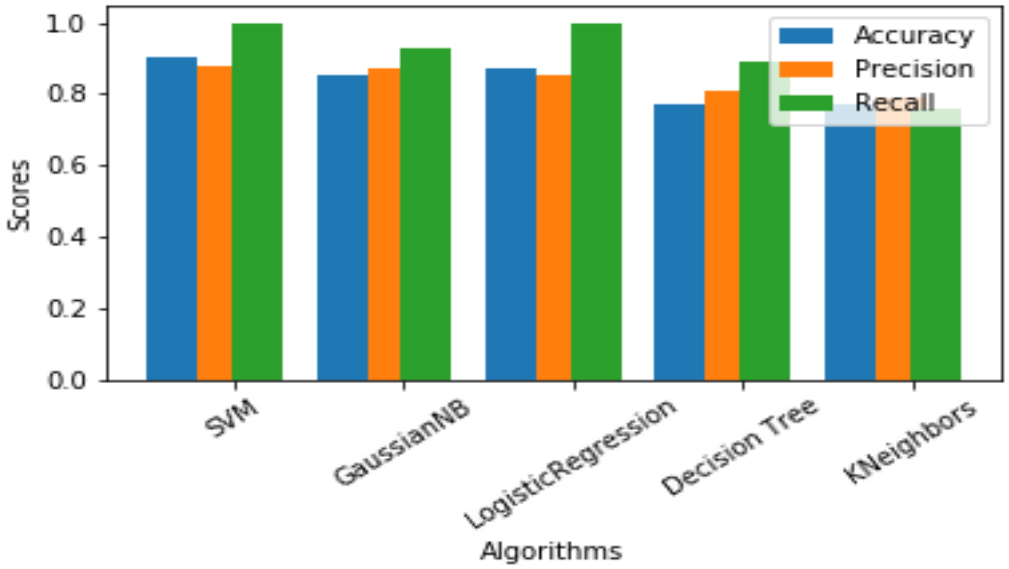


*Figure 5: Performance Metrics Comparison on parameters tuning*

For the default settings and adjusted parameter models, Figures 4 and 5 provide a bar chart of the accuracy-Precision-Recall graph. On the three measures used, the adjusted parameters in figure 5 demonstrate an improvement for the majority of the algorithms. It may be concluded that the performance of most models increased when the parameters were fine-tuned.

*Table 7: Best parameter of tuning parameter using GridSearchCV*

| **Algorithms** | **Tuning Parameters** | **Best Parameters** | **Best scores(10 fold cross validation)** |
|---|---|---|---|
| SVM | C = [0.1,1, 10, 100]<br>Gamma = [scale, auto]<br>kernel = [rbf, poly, sigmoid] | {'C':1, 'gamma': 'scale', 'kernel': 'sigmoid'} | 0.92 |
| GaussianNB | var_smoothing = [logspace(0,-9, num=100)] | {'var_smoothing': 0.66} | 0.84 |

| | | | |
|---|---|---|---|
| Logistic Regression | solver = [lbfgs, newton-cg, liblinear, sag, saga]<br>max_iter = range(100,500,100) | {'solver':'lbfgs',<br>'max_iter':100} | 0.88 |
| Decision Trees | min_samples_split = [range(10,200,10)]<br>max_depth = [range(1,20,1)]<br>criterion = [gini, entropy]<br>max_features = [auto, sqrt, log2] | {'min_samples_split':40,<br>'max_depth': 8,<br>'criterion':'gini',<br>'max_features': 'sqrt'} | 0.92 |
| KNN | algorithm = [auto, ball_tree, kd_tree, brute]<br>n_neighbors = [range(1,30,1)]<br>p = [1,7]<br>weights = [uniform, distance] | {'algorithm': 'auto',<br>'n_neighbors':2, 'p':1,<br>'weights': 'uniform'} | 0.92 |

According to Table 7, the optimized highest best scores are in SVM, Decision Tree and KNN. The tuning SVM are initially being reported to have the highest accuracy of the classifiers as 0.90. Therefore at the parameter {'C':1, 'gamma': 'scale', 'kernel': 'sigmoid'} the accuracy of SVM will perform better when used to classify the features of hepatitis disease.

## CONCLUSION

The grid search strategy was used in this work as a step-by-step strategy to optimizing and selecting acceptable algorithms in hepatitis classification. The SVM, Decision tree, Gaussian Nave Bayes, Logistic regression, and KNN were among the six optimization techniques presented For prediction metrics, a comparison was made between the default settings of five algorithms and their corresponding parameter adjustments. When compared to other methods, it is obvious that the tuning SVM is more accurate. For accuracy, precision, recall, and AUC, the improved SVM algorithm obtained 90 percent, 88 percent, 100 percent, and 82 percent, respectively. The six algorithms were optimized to obtain the optimal parameters, which were then used to validate the approach. The optimal settings for SVM, Decision Tree, and KNN algorithm tuning deliver the optimum performance.

## REFERENCES

Bermúdez-Chacón, R., Gonnet, G. H., & Smith, K. (2015). *Automatic problem-specific hyperparameter optimization and model selection for supervised machine learning: Technical Report: Technical Report* (p. 52 p.) [Application/pdf]. ETH Zurich. https://doi.org/10.3929/ETHZ-A-010558061

Bhargav, K. S., & Kumari, T. D. (2018). *Application of Machine Learning Classification Algorithms on Hepatitis Dataset*. *13*(16), 6.

Cequera, A., & García, L. M. (2014). Biomarkers for liver fibrosis: Advances, advantages and disadvantages. *Revista de Gastroenterología de México (English Edition)*, *79*(3), 187–199. https://doi.org/10.1016/j.rgmxen.2014.07.001

Chown, H. (2019). A Comparison of Machine Learning Algorithms for the Prediction of Hepatitis C NS3 Protease Cleavage Sites. *Journal of Proteomics & Bioinformatics*, *12*(5). https://doi.org/10.35248/0974-276X.19.12.501

Parisi. (2014). *Exploiting Kinetic and Kinematic Data to Plot Cyclograms for Managing the Rehabilitation Process of BKAs by Applying Neural Networks*. *8*(10), 5.

Parisi, Biggs, P. R., Whatling, G., & Holt, C. A. (2015). *A Novel Comparison of Artificial Intelligence Methods for Diagnosing Knee Osteoarthritis*. https://doi.org/10.13140/RG.2.1.4197.6163

Ramadhan, M. M., Sitanggang, I. S., Nasution, F. R., & Ghifari, A. (2017). *Parameter Tuning in Random Forest Based on Grid Search Method for Gender Classification Based on Voice Frequency*. 6.

Yarasuri, V. K., Indukuri, G. K., & Nair, A. K. (2019). *Prediction of Hepatitis Disease Using Machine Learning Technique*. 5.