CHAPTER 6

PERFORMANCE EVALUATION WITH CROSS VALIDATION TECHNIQUES

6.1 Introduction

When training a model, it is important not to overfit or underfit it with algorithms that are too complex or too simple. The choice of training set and test set are critical in reducing this risk. However, dividing the dataset to maximize both learning and validity of test results is difficult. Cross-validation offers several techniques that split the data differently, to find the best algorithm for the model. Cross-validation also helps with choosing the best performing model by calculating the error using the testing dataset, which has not been used to train. The testing dataset helps calculate the accuracy of the model and how it will generalize with future data.

In this section, model validation techniques, namely the k-fold and holdout method are implemented to validate the performance of MPS system as well as to provide a comparison of the performance of the validation techniques themselves. The database used from secondary datasets is divided into training and testing sets and processed in manners suitable for the validation techniques under investigation. The performance is evaluated using a range of different dataset. In more detail, earlier section concentrates on the overview explanation of model validation techniques followed by methodology to be adopted to evaluate the performance the MPS system. Performance result and comparison of different types of model validation techniques being addressed in later section and lastly, conclusions are discussed in final section.

6.2 Cross-Validation Techniques

Cross-validation is a model assessment technique used to evaluate a machine learning algorithm's performance in making predictions on new datasets that it has not been trained on. This is done by partitioning the known dataset, using a subset to train the algorithm and the remaining data for testing. Each round of cross-validation involves randomly partitioning the original dataset into a training set and a testing set. The training set is then used to train a supervised learning algorithm and the testing set is used to evaluate its performance. This process is repeated several times and the average cross-validation error is used as a performance indicator. Cross-Validation is a resampling technique with the fundamental idea of splitting the dataset into 2 parts training data and test data. Train data is used to train the model and the unseen test data is used for prediction. If the model performs well over the test data and gives good accuracy, it means the model hasn't overfitted the training data and can be used for prediction. There are many ways to perform Cross-Validation and in this section will be using two common methods which are holdout and k-fold cross validation method.

6.2.1 Holdout Method

Holdout method can be consider as a basic validation method for result estimation. The holdout technique is an exhaustive cross-validation method, that randomly splits the dataset into train and test data depending on data analysis. The holdout method for training a machine learning model is the process of splitting the data into different splits and using one split for training the model and other splits for validating and testing the models. The holdout method is used for both model evaluation and model selection. When the entire data is used for training the model using different algorithms, the problem of evaluating the models and selecting the most optimal model remains. The primary task is to find out which model out of all models has the lowest generalization error. In other words, which model makes a better prediction on future or unseen datasets than all other models. This is where the need to have some mechanism arises wherein the model is trained on one data set and tested on another dataset. This is where the holdout method comes into the picture.

The holdout method for model evaluation represents the mechanism of splitting the dataset into training and test datasets. The model is trained on the training set and then tested on the testing set to get the most optimal model. This approach is often used when the data set is small and there is not enough data to split into three sets (training, validation, and testing). This approach has the advantage of being simple to implement, but it can be sensitive to how the data is divided into two sets. If the split is not random, then the results may be biased. Overall, the hold out method for model evaluation is a good starting point for training machine learning models, but it should be used with caution. The following represents the holdout method for model evaluation.

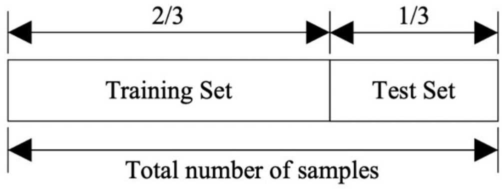

Figure 6.1 Holdout validation distribution ratio

6.2.2 k-fold Method

k-fold cross validation is one way to improve over the holdout method. Partitions data into k randomly chosen subsets (or folds) of roughly equal size. One subset is used to validate the model trained using the remaining subsets. This process is repeated k times such that each subset is used exactly once for validation. The average error across all k partitions is reported as ε. This is one of the most popular techniques for cross-validation but can take a long time to execute because the model needs to be trained repeatedly. Figure 6.2 below illustrates the process with 5-folds validations.

<table>
<tr><td>Fold 1</td><td>Testing set</td><td colspan="4">Training set</td></tr>
<tr><td>Fold 2</td><td>Training set</td><td>Testing set</td><td colspan="3">Training set</td></tr>
<tr><td>Fold 3</td><td colspan="2">Training set</td><td>Testing set</td><td colspan="2">Training set</td></tr>
<tr><td>Fold 4</td><td colspan="3">Training set</td><td>Testing set</td><td>Training set</td></tr>
<tr><td>Fold 5</td><td colspan="4">Training set</td><td>Testing set</td></tr>
</table>

Figure 6.2 The process of k-fold validations with 5-Fold validations (k=5)

6.3 Result and Discussion

In this evaluation, a total of 28 QMR audio samples were used for each class, i.e. Bayyati, Hijaz, Jiharkah, Nahawand, Rast, Siqah and Soba. As mentioned, the MATLAB Classification Learner Toolbox was used to train and test various classifiers for evaluating the performance of the proposed system. The toolbox has the advantage that it can train multiple classifiers using parallel pool with five-fold cross validation was used for evaluating the performance of the classifiers.

Use the confusion matrix plot to understand how the currently selected classifier performed in each class. After you train a classification model, the app automatically opens the confusion matrix for that model. If you train an "All" model, the app opens the confusion matrix for the first model only. To view the confusion matrix for another model, select the model in the Models pane. On the Classification Learner tab, in the

Plots section, click the arrow to open the gallery, and then click Confusion Matrix (Validation) in the Validation Results group. The confusion matrix helps you identify the areas where the classifier performed poorly.

Figure 6.3 shows the Scatter Plot for the model predictions obtained using the Cubic SVM classifier for 7 classes. The x and y axes represent the peak-magnitude-to-RMS-ratio for spectral centroid and roll-off point which gives the most significant features. Figure 6.4 and 6.5 shows the Confusion Matrix for True Positive Rates (TPR) and False Negative Rates (FNR) and confusion matrix for Positive Predictive Values (PPV) and False Discovery Rates (FDR) respectively, meanwhile Figure 6.6 shows the parallel coordinates plot with model predictions.

In this example, which uses the second data set, the third and fourth row from the top shows all maqamat types with the true class Jiharkah and Rast respectively. The columns show the predicted classes. For the maqamat type from Jiharkah and Rast, only 50% are correctly classified, so 50% is the true positive rate for correctly classified points in this class, shown in the blue cell in the TPR column. The other maqamat in the Jiharkah row are misclassified: 25% of the maqamat are incorrectly classified as from Nahawand, and 25% are also misclassified as from Soba. The false negative rate for incorrectly classified points in this class is 50%, shown in the orange cell in the FNR column. This rate is consider relatively high due its significance characteristics in acoustical element and matched with experimental results in previous section 5.6. The PPV is the proportion of correctly classified observations per predicted class. It shows that only maqamat type Nahawand is correctly classified with 66.7% and 80% for Soba meanwhile other types are correctly classified with 100%. For FDR, Nahawand and Soba scores 33.3% and 20% respectively, that is the proportion of incorrectly classified observations per predicted class. Positive predictive values are shown in blue for the correctly predicted points in each class, and false discovery rates are shown in orange for the incorrectly predicted points in each class.

From amongst various classifiers used, Cubic SVM classifier give the highest accuracy of 89.3%. Most of SVM classifiers give promising results because its ability to classifies the available data by optimally finding the best possible hyperplane which separates the data points of the various classes. In these experiments there are more than two classes, so the toolbox uses *fitcecoc* function and reduces the multiclass classification problem to a binary state, with individual SVM learner for every subclass.

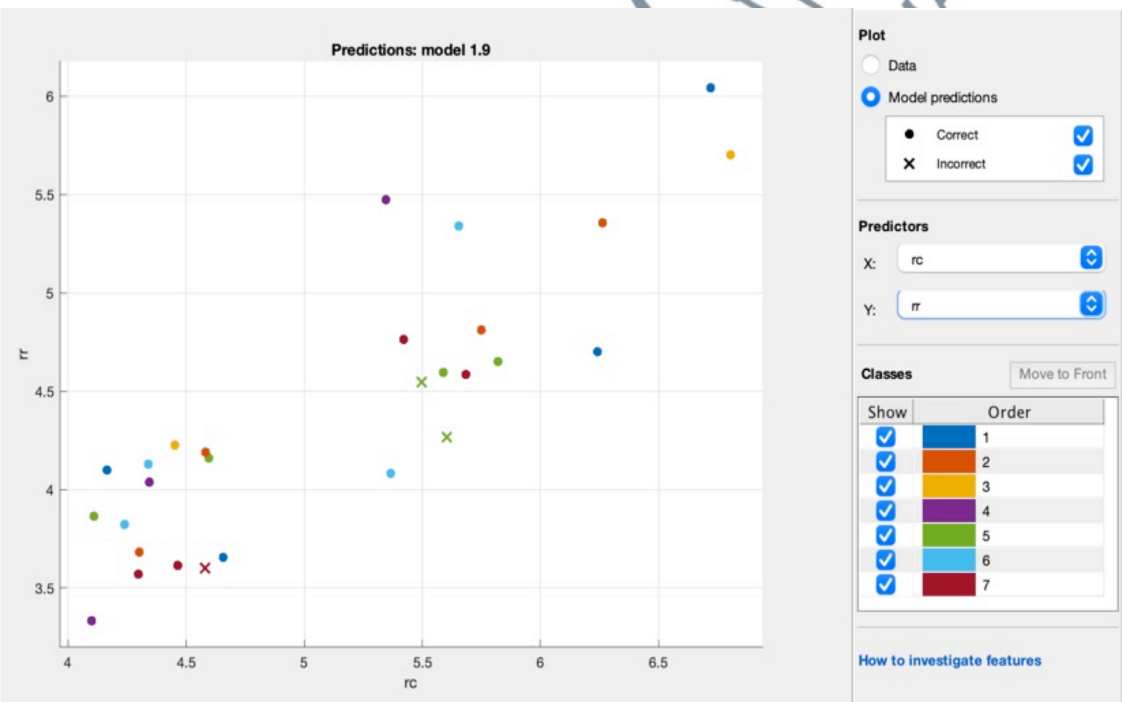

Figure 6.3 Scatter Plot with prediction model

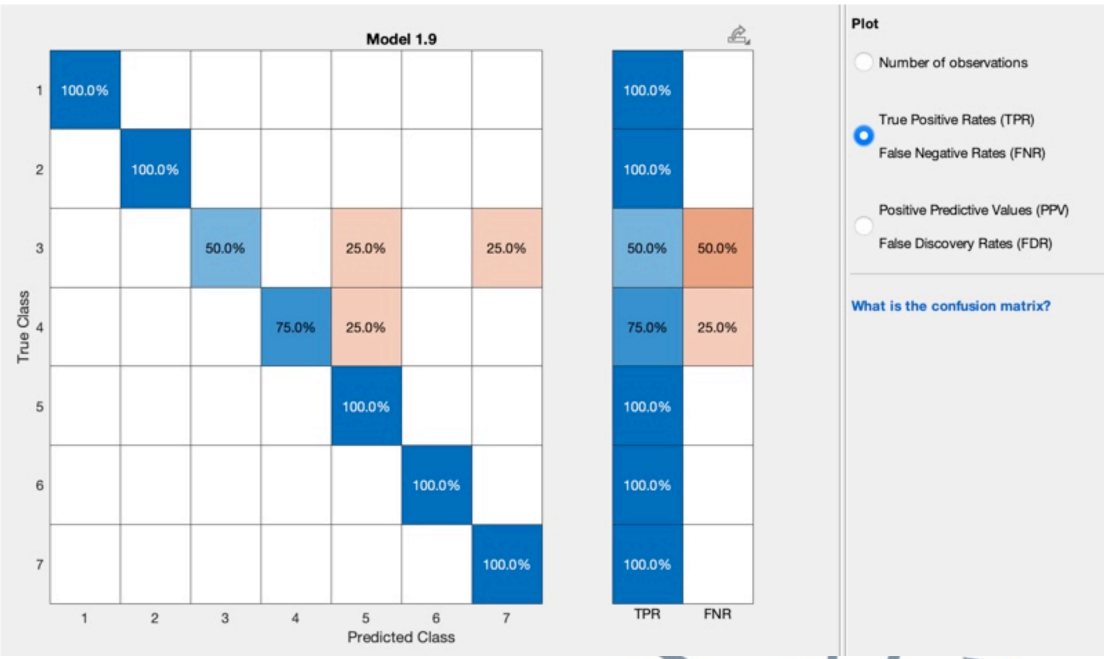

Figure 6.4 Confusion matrix for TPR and FNR

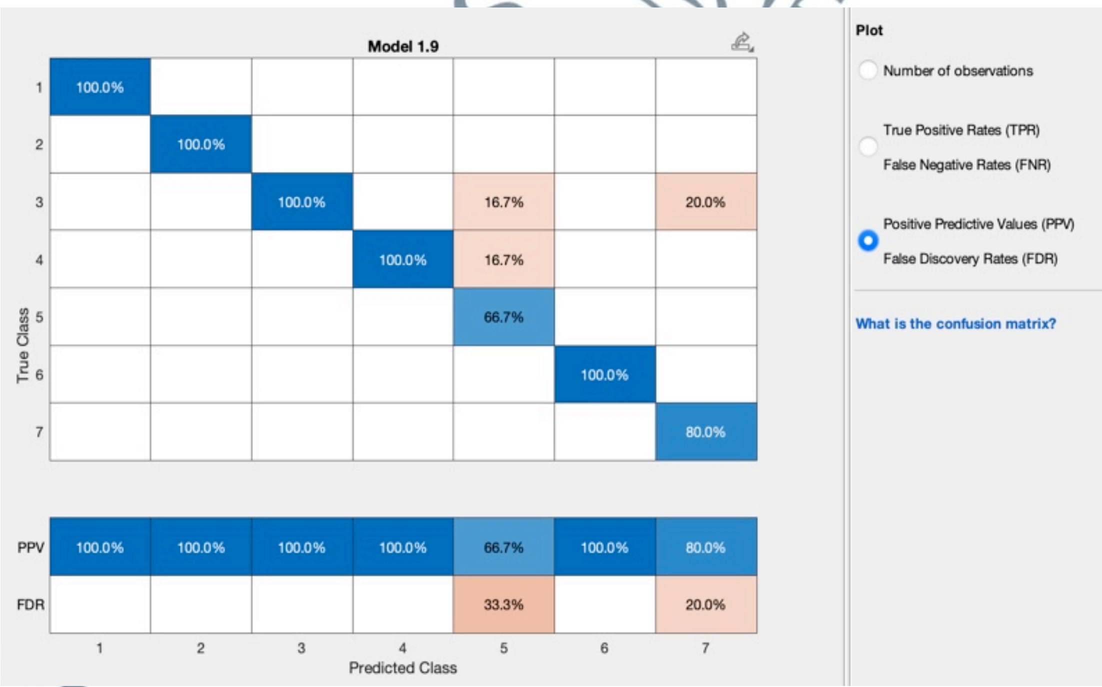

Figure 6.5 Confusion matrix for PPV and FDR

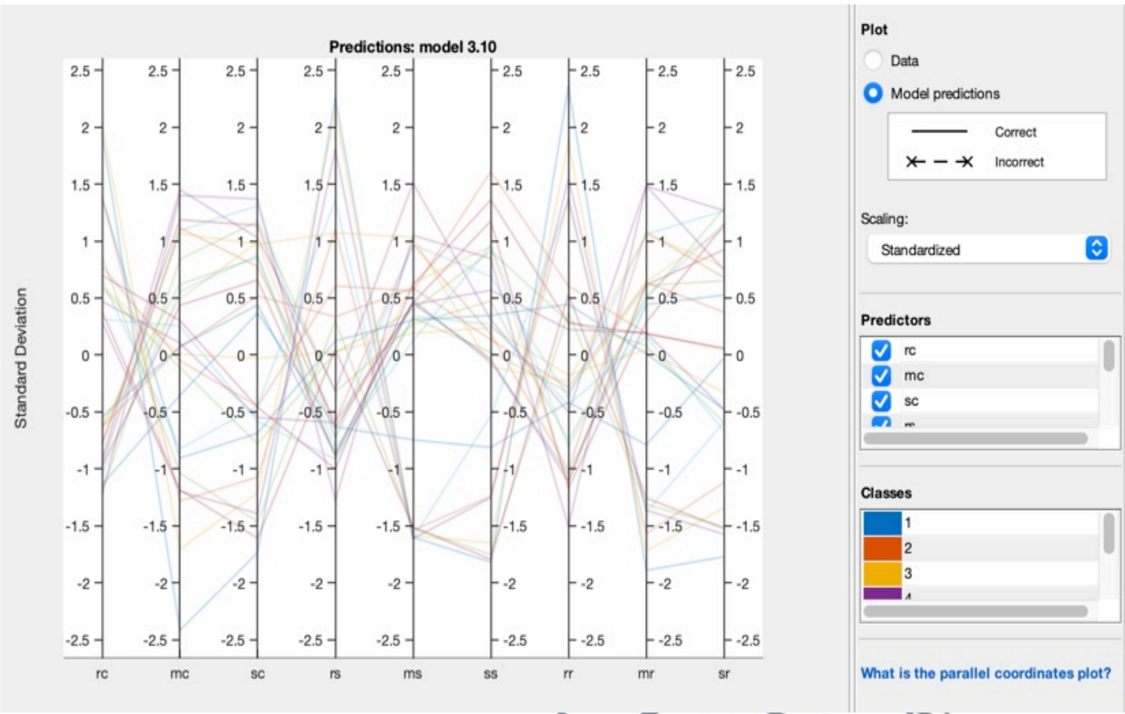

Figure 6.6 Parallel coordinates plot with model predictions

Table 6.1 Accuracies of k-fold and holdout validations with various classifiers used

Classifier Type	**k-fold (%)**	**Holdout (%)**
Linear SVM	35.7	56.8
Quadratic SVM	45	53
Cubic SVM	**89.3**	**91.6**
Fine Gaussian SVM	75	70
Medium Gaussian SVM	64.3	73.8
Coarse Gaussian SVM	81.4	60
Fine KNN	78.7	73.9
Cubic KNN	17.9	45
Weighted KNN	75	67
Subspace KNN	75	85.7

6.4 Summary

This chapter presented result cross validation evaluation comparison on varies validation techniques on MPS system using a set of 42 samples from both primary and secondary datasets. The validation techniques used are k-fold and hold-out method. This technique manage to evaluate quality of every samples and gain the final accuracy by averaging the result from each samples. The proposed algorithm proves to be accurate and efficient in identifying the types of maqamat based on the spectral features. It works very well in identifying few maqamat with more than 70% accuracy such as Jiharkah, Nahawand and Sikah. These 3 types of maqamat give significant peaks value justifying a strong emotion within. Positive predictive values with 100% for the correctly predicted points for all types except Nahawand and Soba with false discovery rates are shown in orange for the incorrectly predicted points in each class. The minimal value of accuracy is obtained in identification of the Soba and Nahawand type with accuracy below 30% due to its elements of emotion being minimal.