Kernel Methods for Pattern Analysis
Assignment 3

Parth Joshi (CS09B051), Sudharshan GK (CS13M050) & Umesh Kanoja (CS13M024)

April 15, 2014

1 Methodology

1. **Use of normalized polynomial kernel :** The `libsvm` tool provides a polynomial kernel which is not normalized and hence visualizing the kernel gram matrix generated proved to be futile. Therefore we implemented our own polynomial kernel which is normalized using following expression

$$\widetilde{K}(m,n) = \frac{K(m,n)}{\sqrt{K(m,m) * K(n,n)}}$$

.

2. **Dealing with symbolic attributes in multivariate data for anomaly detection :** The data provided for the multivariate tasks contained 41 attributes some of which consisted of symbolic data. While some of the symbolic features were binary, the attributes `protocol_type`, `service` and `flag` contained text with 3, 66 and 11 unique values respectively. Therefore, we transformed this data representation into one where these attributes are represented by 3, 66 and 11 binary features respectively [LIBSVM]. This led to an expansion in the dimensionality from 41 to 118.

3. **Normalization of feature values :** In case of image data the color histogram were not scaled as the number of pixels corresponding to each images were different. Hence we normlized colour histogram for every image by the number of pixels. This has increased the performance of the model tremendously. For the regression the datasets were all normalized to the range $[0, 1]$ as part of pre-processing. Similarly, it was seen that the features in the multivariate data for anomaly detection had widely disparate ranges and so that data was also scaled to lie in the range $[0, 1]$.

4. **Computing kernel-gram error** : We have used the Frobenius norm of the difference between the obtained and the ideal kernel gram matrices as an evalutaion criterion to determine the kernel hyperparameters. Specifically for the classification tasks, this was used to determine the model parameters that relate to the kernel function such as the degree and co-efficients in the case of polynomial kernels or the width for Gaussian kernels before determining other empirical parameters such as C or ν.

5. **Estimating empirical parameters:** Two criteria have been used for determination of empirical parameters. The first, as indicated above, was the kernel-gram error. The second was the model error on validation data. In case of classification and novelty detection we considered validation accuracy while for regression we considered mean squared error as our model error. The kernel gram error was used to estimate kernel function specific parameters. The search for the best model parameters was done by starting with a large logarithmically scaled window and narrowing the search in successive passes to converge to the model parameters that give the best value of the chosen criterion. One point to note is that since we are using a normalized polynomial kernel, we do not need to estimate both the a and b co-efficients in the function $(a\mathbf{x}^t\mathbf{x} + b)^d$. Therefore, we always take the value of a to be 1.

6. **Reciever Operating Characteristics :** We have made use of ROC curves in some cases to compare the performance of different classifiers. These curves plot the hit rate against the false alarm rate and give a visual representation of the *discriminability* between the different classes [DH, pg 49].

Part I

Classification

2 Bivariate Datasets

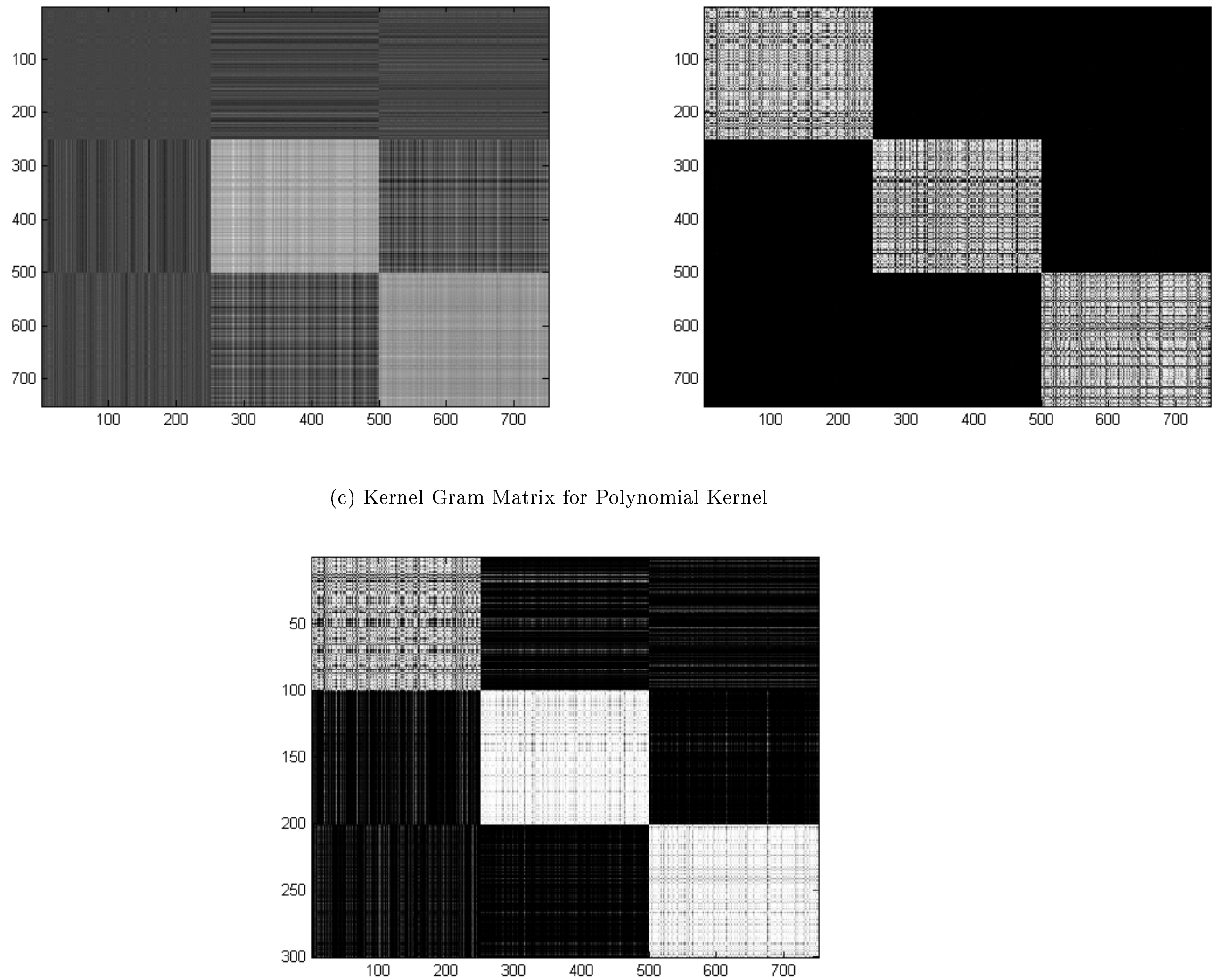
2.1 Linearly separable classes

The linearly separable dataset we recieved consists of three classes. The following three models were used for classification

- A C-SVM classifier using the linear kernel with the cost parameter $C = 1$.
- A C-SVM classifier using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 0.5$
  - Cost parameter $C = 1$
- A C-SVM classifier using the polynomial kernel with the following configuration,
  - Degree of polynomial $d = 4$
  - Constant term in the polynomial $b = 34$
  - Cost parameter $C = 1$
- A ν-SVM classifier using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 0.5$ (same as in the case of C-SVM)
  - Lower bound on fraction of support vectors $\nu = 0.25$.

The plot in figure 1 shows the kernel gram matrices for different kernels on the training dataset. The decision region plots obtained from each of the three models are shown in figure 2. The confusion matrix and classification accuracy for all models is the same and is shown in figure 3.

Figure 1: Kernel Gram matrices for different Kernels

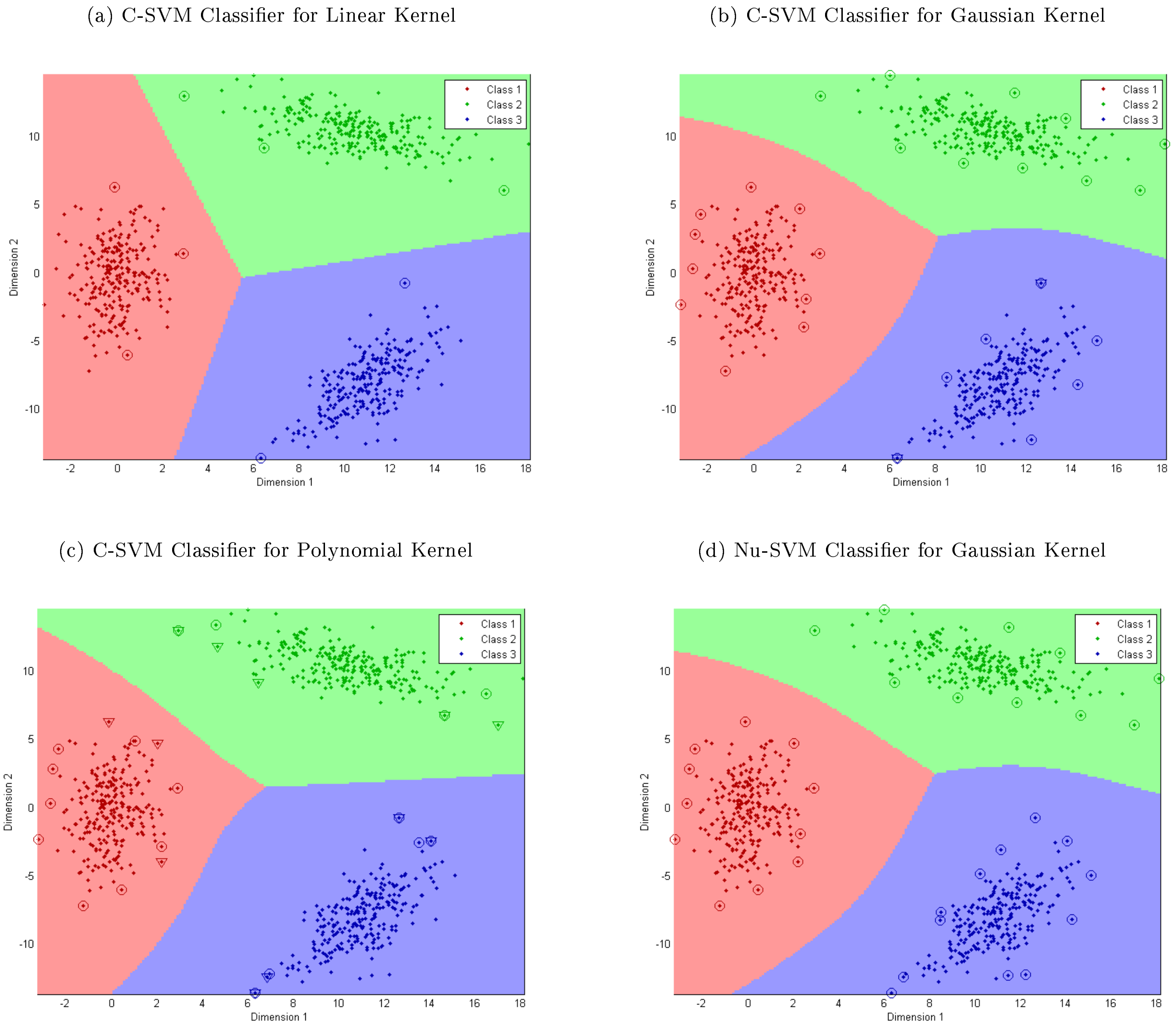


(a) Kernel Gram Matrix for Linear Kernel

(b) Kernel Gram Matrix for Gaussian Kernel

(c) Kernel Gram Matrix for Polynomial Kernel

Figure 2: Decision region plots for different classifiers on the linearly separable dataset. The points in the train dataset are superimposed on the decision regions. The points that are shown in triangles are bounded support vectors and the ones which are shown in circles are unbounded support vectors.



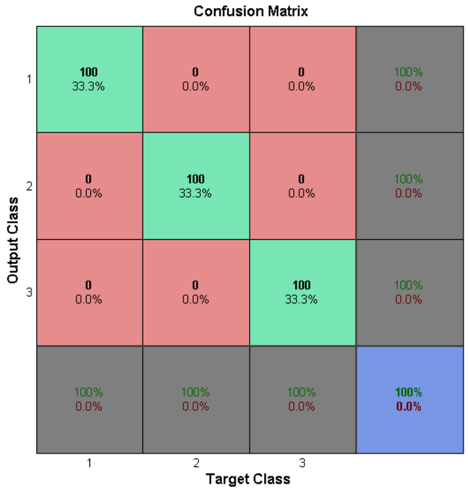
(a) C-SVM Classifier for Linear Kernel

(b) C-SVM Classifier for Gaussian Kernel

(c) C-SVM Classifier for Polynomial Kernel

(d) Nu-SVM Classifier for Gaussian Kernel

Figure 3: Confusion matrix and classification accuracy of all classifier model



Confusion Matrix

Output Class	1	2	3	
1	100 33.3%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	100 33.3%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	100 33.3%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

Target Class

Observations

- The linearly separable case is the easiest to classify and all three models are able to give 100% accuracy on the given data.
- It was seen that the validation accuracy was 100% for a wide range of C.
- The decision surfaces obtained from the linear kernel are always linear, while other kernels can give non-linear decision boundaries.

Inferences

- Any of the above models can be used for classifying linearly separable classes with good results.
- Choosing ν to be too small reduces the flexibility of the model and reduces the margin which in turn makes the generalization performance of the model poor. Choosing ν to be too large results in a large number of support vectors which can make testing a new point computationally expensive.

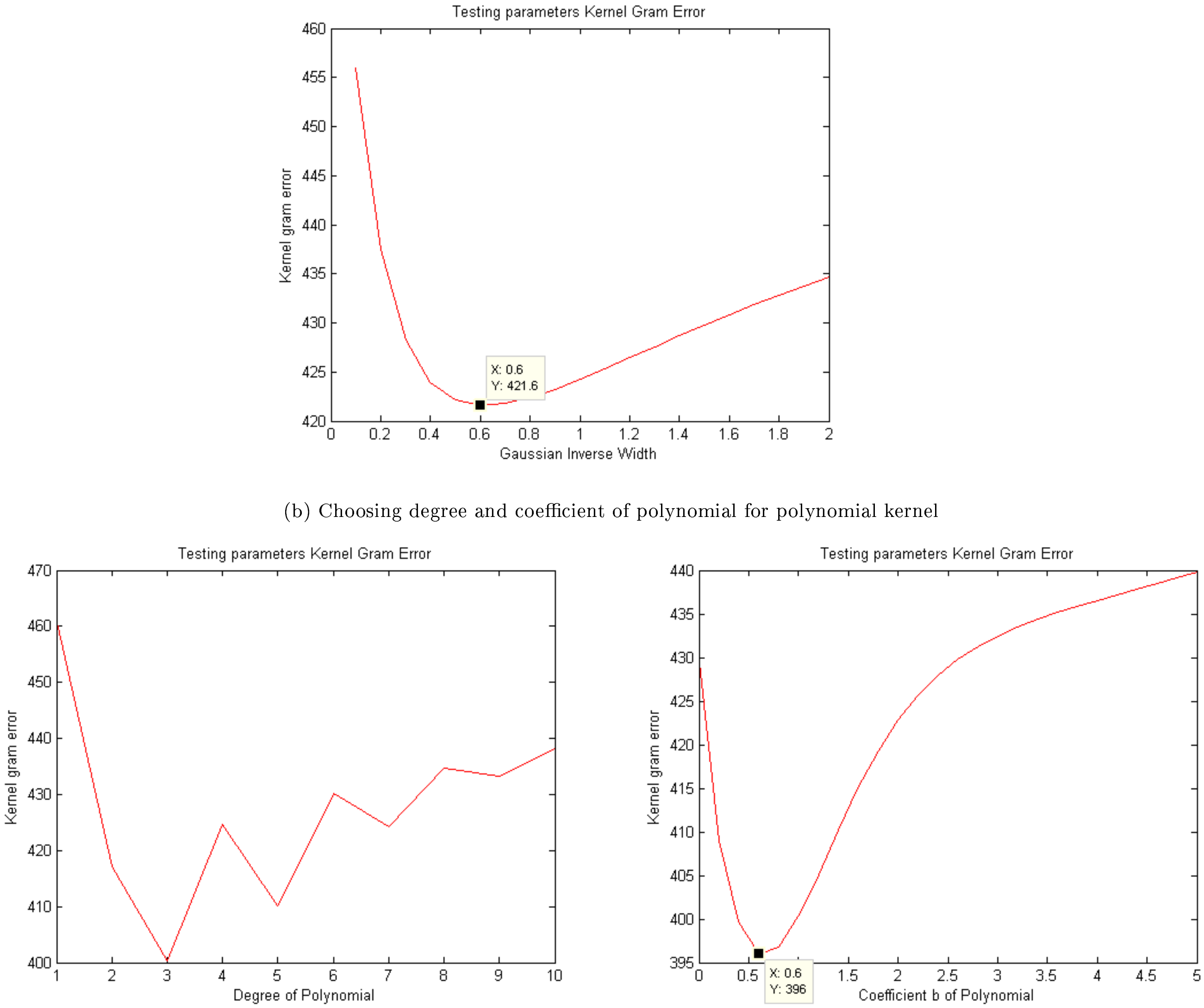
2.2 Non-linearly separable classes

The given dataset consists of two classes. The following classifier models were built for this dataset,

- A C-SVM classifier using the gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 0.6$
  - Cost parameter $C = 1$
- A C-SVM classifier using the polynomial kernel with the following configuration,
  - Degree of polynomial $d = 3$
  - Constant term in the polynomial $b = 0.6$
  - Cost parameter $C = 1$
- A ν-SVM classifier using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 0.5$ (same as in the case of C-SVM)
  - Lower bound on fraction of support vectors $\nu = 0.001$

  The plots in figure 4 indicate why the above choices of parameters was made. The plot in figure 4 and figure 5 show the kernel gram matrices for different kernels on the training dataset. The decision region plots obtained from each of the three models are shown in figure 6. The confusion matrix and classification accuracy for all models is the same and is shown in figure 3.

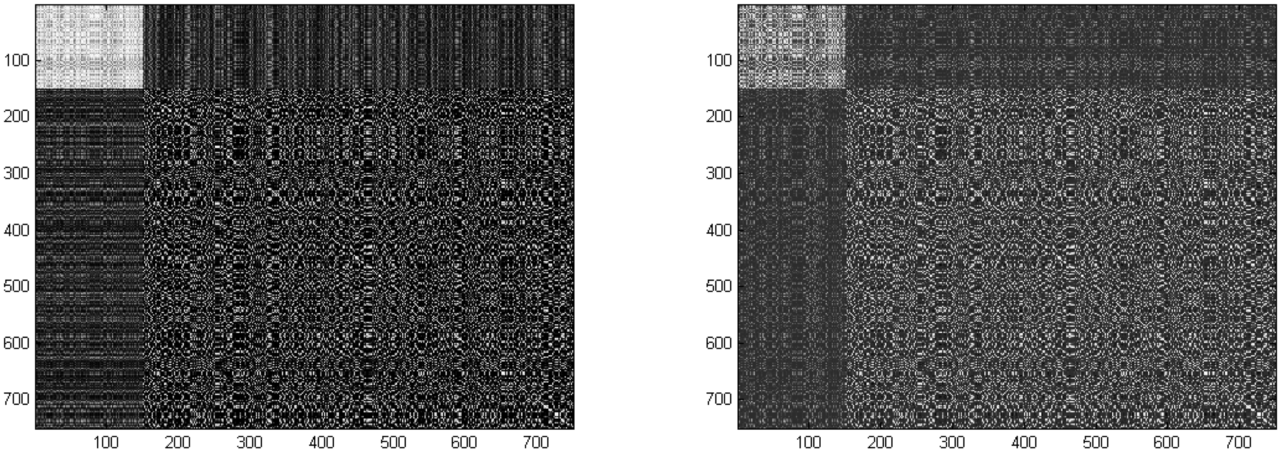
Figure 4: Determination of different kernel parameters



(a) Choosing Gaussian kernel width

(b) Choosing degree and coefficient of polynomial for polynomial kernel

Figure 5: Kernel gram matrices for different kernels

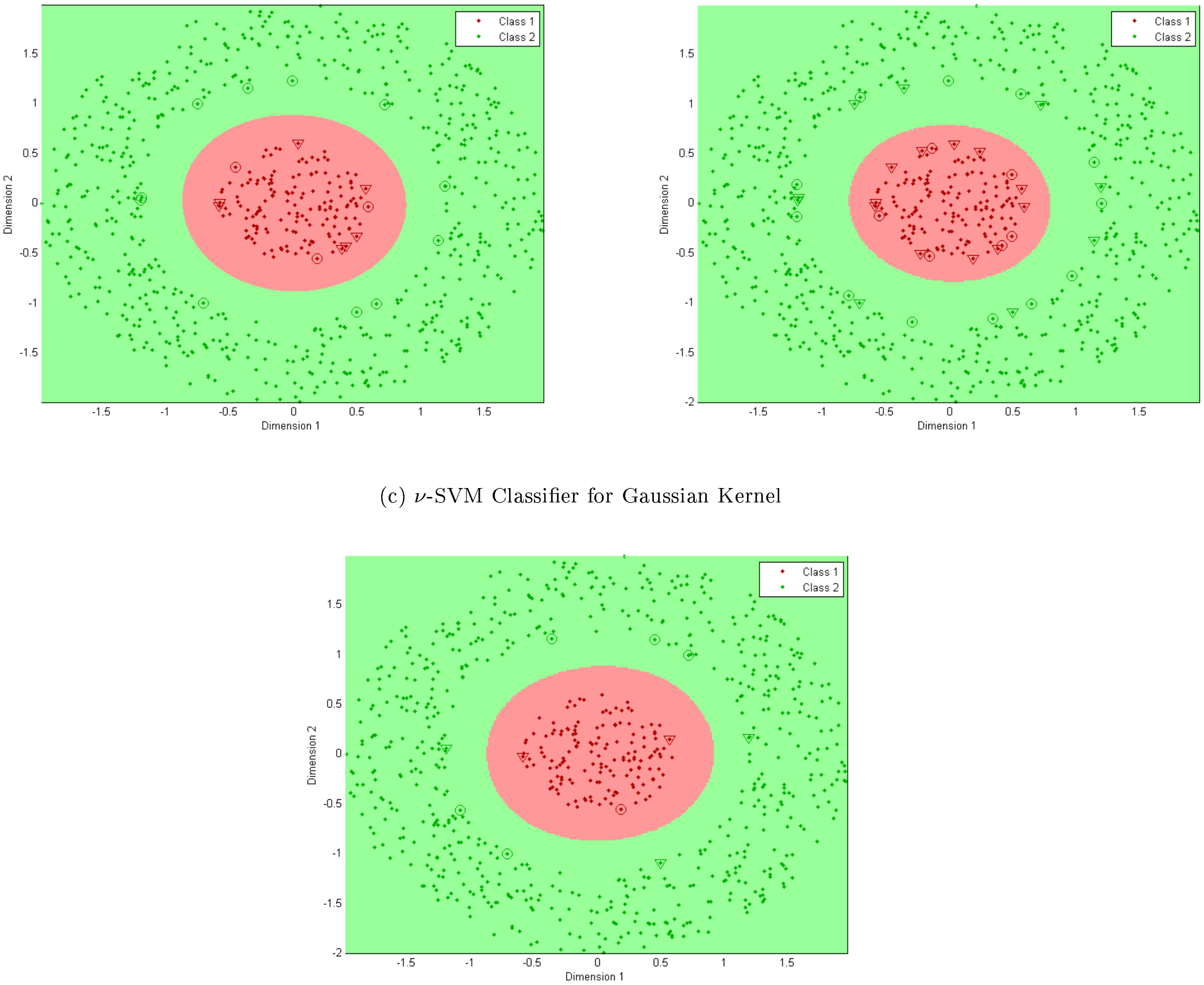

(a) Kernel Gram Matrix for Gaussian Kernel

(b) Kernel Gram Matrix for Polynomial Kernel

Figure 6: Decision region plots for different classifiers on the non-linearly separable dataset. The points in the training dataset are superimposed on the decision regions. The points marked by triangles are bounded support vectors and the ones which marked by circles are unbounded support vectors.

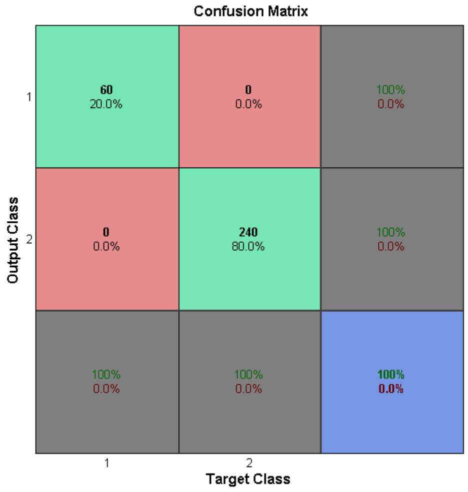
(a) C-SVM classifier for Gaussian kernel

(b) C-SVM Classifier for Polynomial Kernel



(c) ν-SVM Classifier for Gaussian Kernel

Figure 7: Confusion matrix and classification accuracy of all classifier models



Confusion Matrix

Output Class

	1	2	
1	60 20.0%	0 0.0%	100% 0.0%
2	0 0.0%	240 80.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%

Target Class

Observations

- The data is separable into 2 distinct regions and both models are able to classify the given data with 100% accuracy.
- The decision boundaries formed by the all classification models are quite similar.
- The polynomial kernel requires a larger number of support vectors as compared to the Gaussian kernel (figure 6). Also, the margin obtained for the former is smaller than the latter.

Inferences

- The Gaussian kernel has only one empirical parameter whereas the number of parameters to be determined for the polynomial kernel is more. Thus, determining the optimal parameters for the polynomial case is more difficult which can explain why the polynomial kernel model has more support vectors (figure 6).

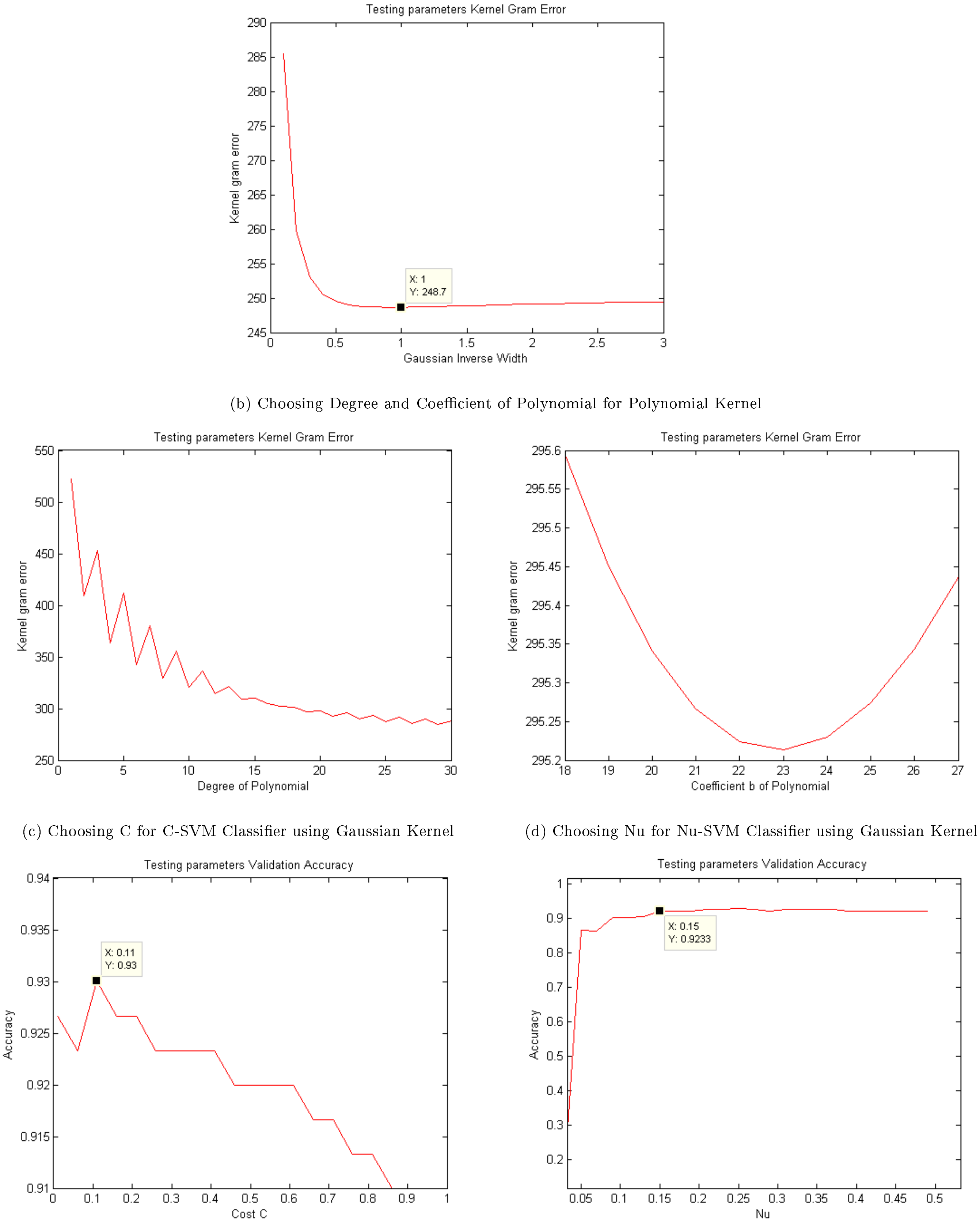
2.3 Overlapping classes

The given dataset consists of three classes. The following models were built for this dataset

- A C-SVM classifier using the linear kernel
  - Cost Parameter $C = 1.3$
- A C-SVM classifier using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 1$
  - Cost parameter $C = 0.1$
- A C-SVM classifier using the polynomial kernel with the following configuration,
  - Degree of polynomial $d = 15$
  - Constant term in the polynomial $b = 23$
  - Cost parameter $C = 0.11$
- A ν-SVM classifier using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 1$ (same as in the case of C-SVM)
  - Lower bound on fraction of support vectors $\nu = 0.15$

  The plots in figure 8 indicate why the above choices of parameters was made. The plots in figure 4 show the kernel gram matrices for different kernels on the training dataset. The decision region plots obtained from each of the three models are shown in figure 9. The confusion matrix and classification accuracy for each model is shown in figure 11.

Figure 8: Determination of different kernel parameters for different kernel functions



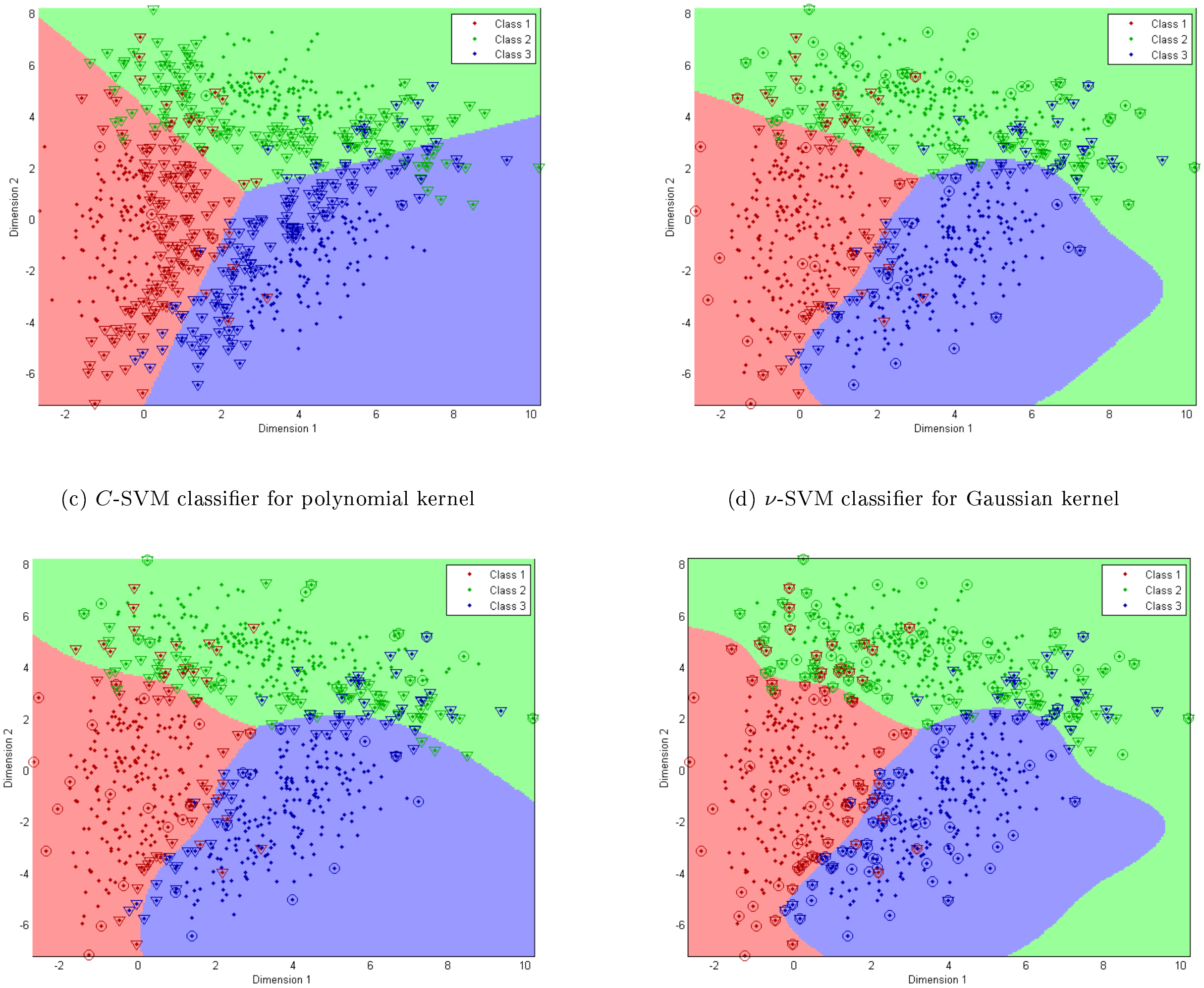
(a) Choosing Gaussian Kernel width

(b) Choosing Degree and Coefficient of Polynomial for Polynomial Kernel

(c) Choosing C for C-SVM Classifier using Gaussian Kernel

(d) Choosing Nu for Nu-SVM Classifier using Gaussian Kernel

Figure 9: Decision region plots for different classifiers on the overlapping dataset. The points in the train dataset are superimposed on the decision regions the points that are shown in triangle are bounded vectors and the ones which are shown in circle are unbounded vectors.



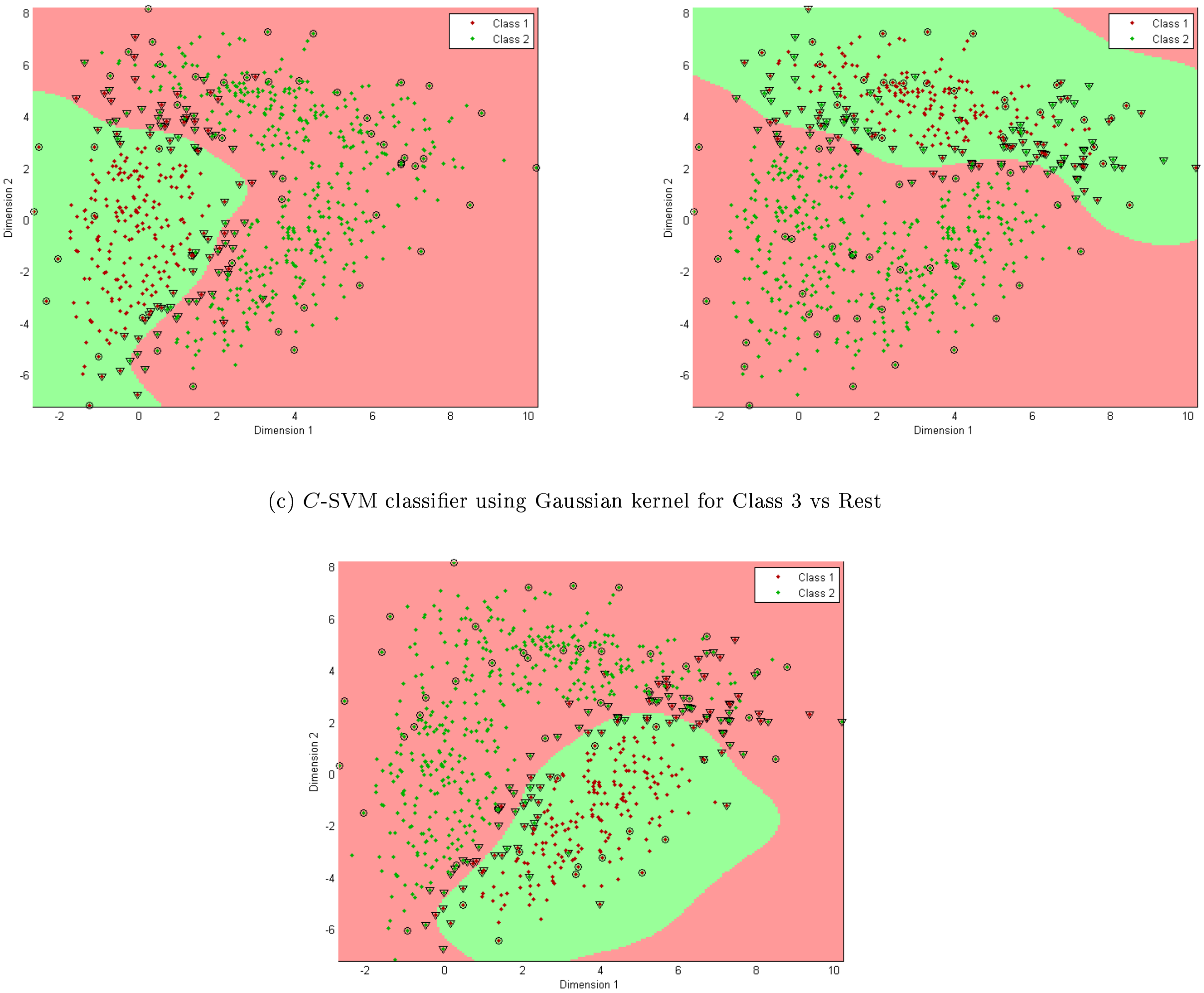
(a) C-SVM Classifier using Linear Kernel

(b) C-SVM classifier using Gaussian kernel

(c) C-SVM classifier for polynomial kernel

(d) ν-SVM classifier for Gaussian kernel

Figure 10: Decision region plots for Gaussian classifiers on the overlapping dataset in a one vs rest approach. The points in the train dataset are superimposed on the decision regions the points that are shown in triangle are bounded vectors and the ones which are shown in circle are unbounded vectors.

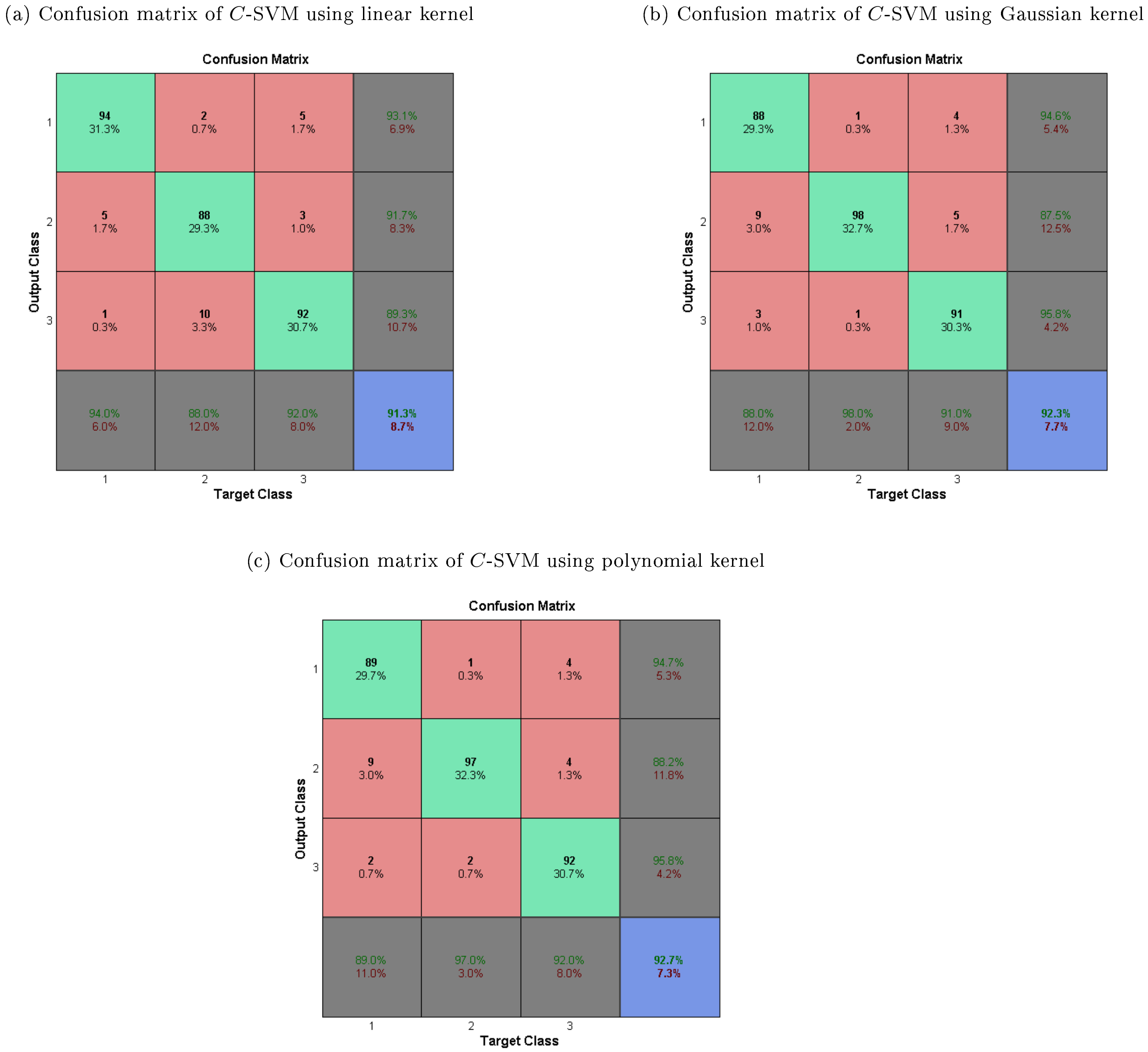


(a) C-SVM classifier using Gaussian kernel for Class 1 vs Rest

(b) C-SVM classifier using Gaussian kernel for Class 2 vs Rest

(c) C-SVM classifier using Gaussian kernel for Class 3 vs Rest

Figure 11: Confusion matrix and classification accuracy of all the classifier models



(a) Confusion matrix of C-SVM using linear kernel

(b) Confusion matrix of C-SVM using Gaussian kernel

(c) Confusion matrix of C-SVM using polynomial kernel

Observations

- The classification accuracy of the model using a polynomial or Gaussian kernel is quite similar for both C-SVM and ν-SVM. However, the linear SVM is constrained to obtain a linear decision boundary and yields a model that performs worse than the others.
- figure 10 shows the decision regions for the binary component classifiers that are combined in the one vs rest approach. It can be observed that some points are bounded support vectors for one classifier and at the same time unbounded support vectors for another classifier. The decision regions in figure 9b indicate the bounded and unbounded support vectors for all these individual component classifiers together in the complete one vs rest model. The points marked with both triangles and circles are the ones that are both bounded and unbounded.

Inferences

- Here, again we see that the linear kernel is only able to produce a linear decision boundary while the polynomial and Gaussian kernel produce non-linear boundaries.

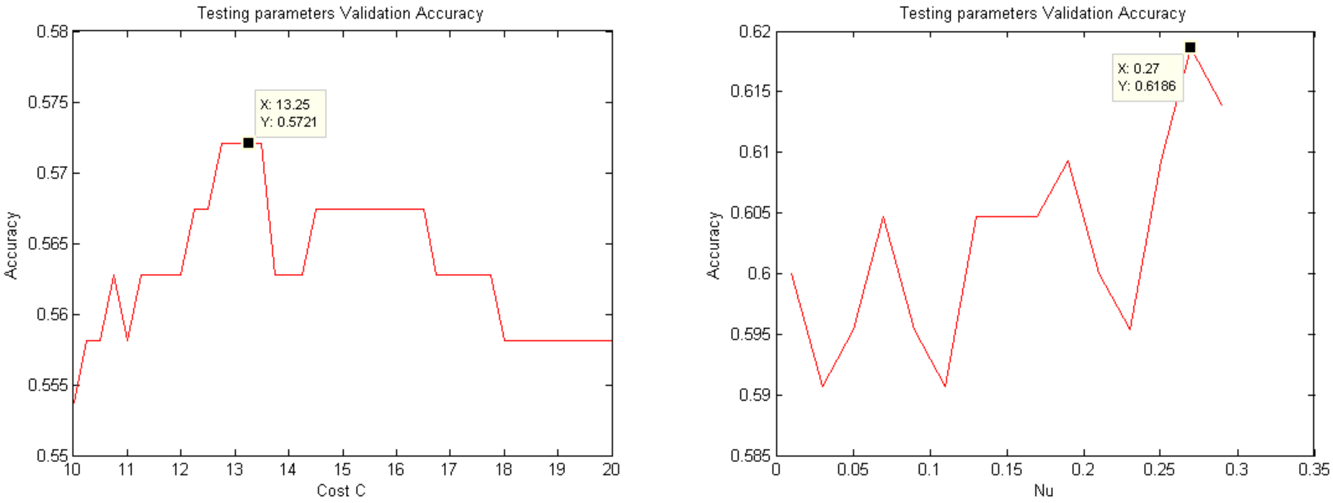
3 Image Dataset

The given dataset consists of five classes — “billiards”, “hot-tub”, “mushroom”, “bath-tub” and “leopards”. The feature vectors are 48-dimensional colour histograms which is normalized to use it for Kernel Computation. The following models were built for this dataset

- A C-SVM classifier using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 98$
  - Cost parameter $C = 1.9$
- A C-SVM classifier using the histogram intersection kernel with the following configuration,
  - Co
  - $\epsilon = 0.3$ taken from graph MSE vs ϵ.st parameter $C = 13.25$
- A ν-SVM classifier using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 98$ (same as in the case of C-SVM)
  - Lower bound on fraction of support vectors $\nu = 0.27$

The plots in figure 12 indicate why the above choices of parameters was made. The confusion matrix and classification accuracy for each of the models is shown in figure 11.

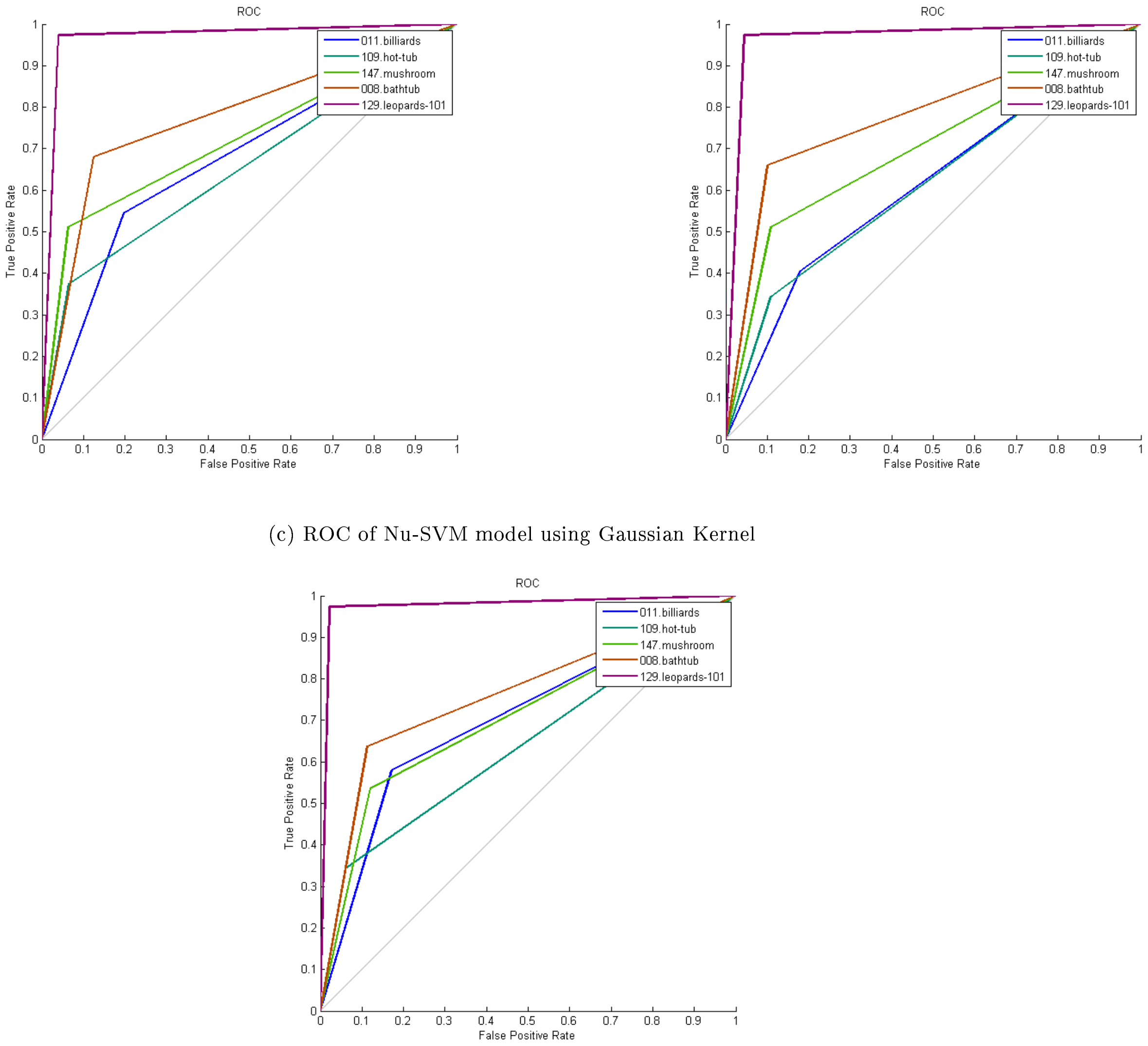
Figure 12: Determination of different kernel parameters for different kernel functions



(a) Choosing C for C-SVM classifier using histogram kernel

(b) Choosing ν for ν-SVM classifier using Gaussian kernel

Figure 13: Receiver operating characterstics of all classifier on the image data

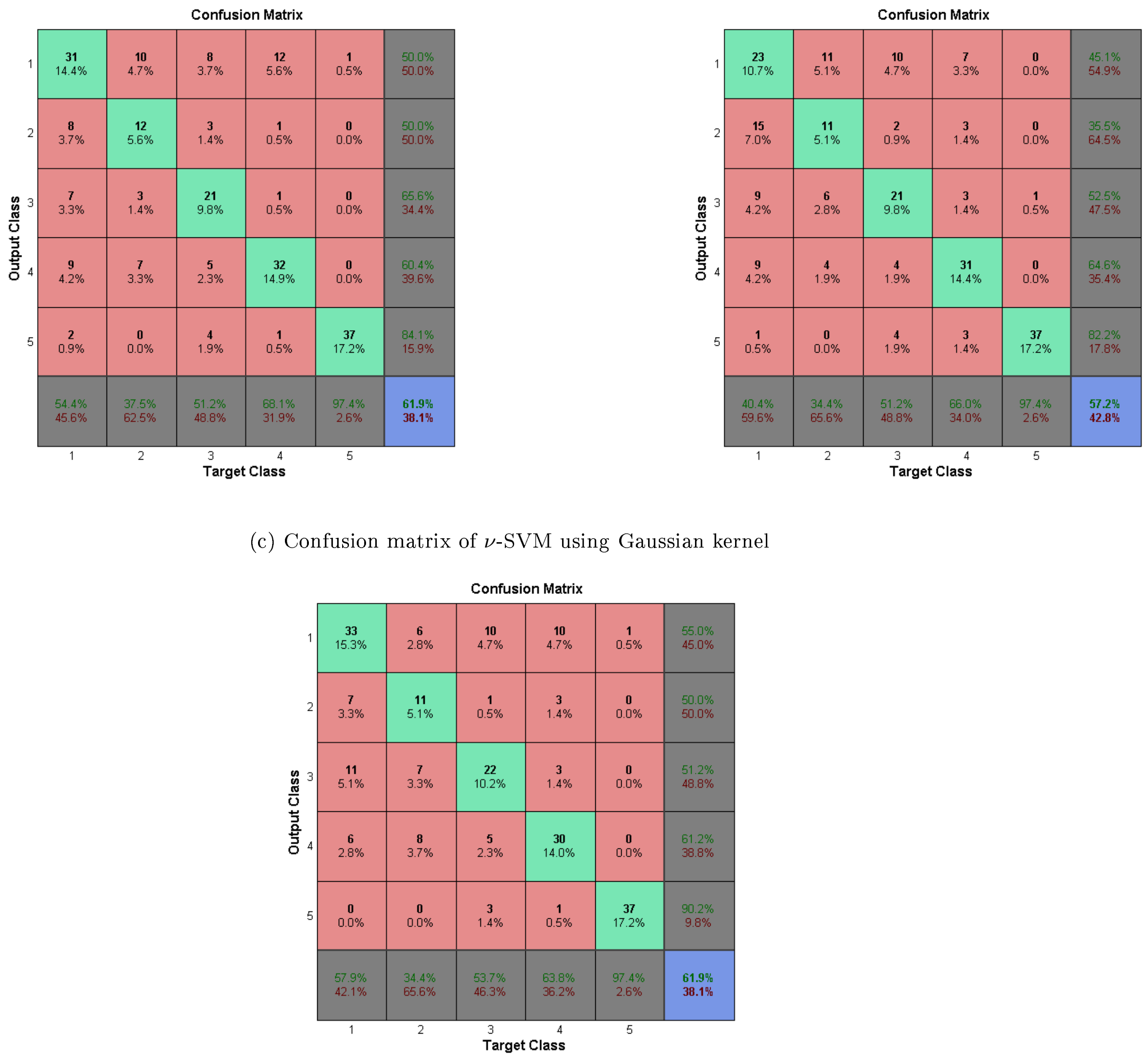


(a) ROC of C-SVM model using Gaussian Kernel

(b) ROC of C-SVM model using Histogram Kernel

(c) ROC of Nu-SVM model using Gaussian Kernel

Figure 14: Confusion matrix and classification accuracy of all classifier model



(a) Confusion matrix of C-SVM using Gaussian kernel

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	
1	31 14.4%	10 4.7%	8 3.7%	12 5.6%	1 0.5%	50.0% 50.0%
2	8 3.7%	12 5.6%	3 1.4%	1 0.5%	0 0.0%	50.0% 50.0%
3	7 3.3%	3 1.4%	21 9.8%	1 0.5%	0 0.0%	65.6% 34.4%
4	9 4.2%	7 3.3%	5 2.3%	32 14.9%	0 0.0%	60.4% 39.6%
5	2 0.9%	0 0.0%	4 1.9%	1 0.5%	37 17.2%	84.1% 15.9%
	54.4% 45.6%	37.5% 62.5%	51.2% 48.8%	68.1% 31.9%	97.4% 2.6%	61.9% 38.1%

(b) Confusion matrix of C-SVM using histogram intersection kernel

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	
1	23 10.7%	11 5.1%	10 4.7%	7 3.3%	0 0.0%	45.1% 54.9%
2	15 7.0%	11 5.1%	2 0.9%	3 1.4%	0 0.0%	35.5% 64.5%
3	9 4.2%	6 2.8%	21 9.8%	3 1.4%	1 0.5%	52.5% 47.5%
4	9 4.2%	4 1.9%	4 1.9%	31 14.4%	0 0.0%	64.6% 35.4%
5	1 0.5%	0 0.0%	4 1.9%	3 1.4%	37 17.2%	82.2% 17.8%
	40.4% 59.6%	34.4% 65.6%	51.2% 48.8%	66.0% 34.0%	97.4% 2.6%	57.2% 42.8%

(c) Confusion matrix of ν-SVM using Gaussian kernel

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	
1	33 15.3%	6 2.8%	10 4.7%	10 4.7%	1 0.5%	55.0% 45.0%
2	7 3.3%	11 5.1%	1 0.5%	3 1.4%	0 0.0%	50.0% 50.0%
3	11 5.1%	7 3.3%	22 10.2%	3 1.4%	0 0.0%	51.2% 48.8%
4	6 2.8%	8 3.7%	5 2.3%	30 14.0%	0 0.0%	61.2% 38.8%
5	0 0.0%	0 0.0%	3 1.4%	1 0.5%	37 17.2%	90.2% 9.8%
	57.9% 42.1%	34.4% 65.6%	53.7% 46.3%	63.8% 36.2%	97.4% 2.6%	61.9% 38.1%

Observations

- We found that with C-SVM model, the best classification accuracy we could obtain was only $\approx 57\%$ using the histogram intersection kernel while the Gaussian kernel gave an accuracy of $\approx 62\%$ over the test data.
- It can be seen that the "leopards" class is classified well by both the models but the predictions for the rest of the classes are much more confused.
- `libsvm` provides a mechanism for weighting the cost parameter C differently for differently classes in order to deal with class imbalance. We sought to make use of this feature by assigning classes weights proporionate to the number of examples in them. Note that in the case of the previous datasets, class imbalance was not as much of an issue but here, we have five classes for classification and hence the ratio of examples belonging to a given class and those not belonging to that class is $\approx 1 : 4$. The accuracies we obtained for the two models improved marginally in this case. The histogram intersection kernel gave an accuracy of 60.93% in this case while the Gaussian kernel gave an accuracy of 63.25%.

Inferences

- The images in the "leopard" class contain primarily dark shades of yellow and green, which are not present in the same amount in the images of other classes whereas the colour histograms for the other classes are similar. This causes this class to be distinguished more easily.
- The low classification accuracies indicate that colour histograms alone might not provide enough discriminatory information to be able to clearly distinguish between classes of images.
- We were surprised to find that the Gaussian kernel performed better than the histogram intersection kernel unlike the results reported in [BOV]. This could be an artifact of the specific choice of images we were given for this assignment. Most of the image classes we received had similar colour distributions and thus the simplistic operation used to compute the histogram intersection kernel might not suffice to distinguish between the classes.

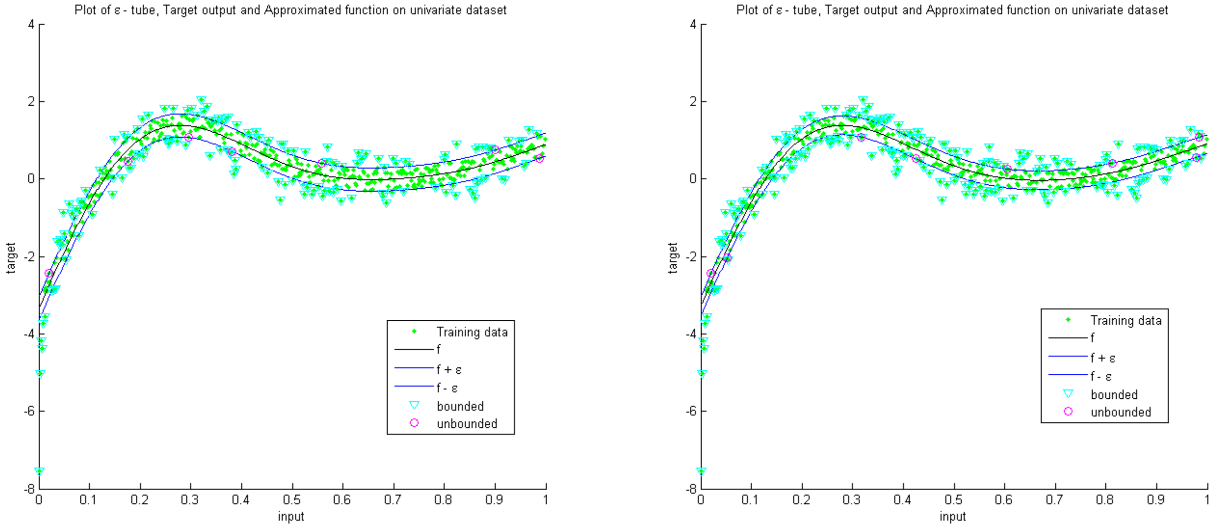
Part II
Regression

4 Univariate Data

The following models were used to perform regression on the univariate data

- $\epsilon - SVR$ model with the following configuration parameters
  - Cost parameter$C = 14$.
  - Inverse width of the Gaussian kernel$\gamma = 8$.
  - $\epsilon = 0.3$ taken from graph MSE vs ϵ (figure 16).
- $\nu - SVR$ model with the following parameters
  - Cost parameter $C = 14$.
  - Inverse width of the Gaussian kernel$\gamma = 8$.
  - $\nu = 0.4$ taken from the graph of MSE vs ν (figure 17).

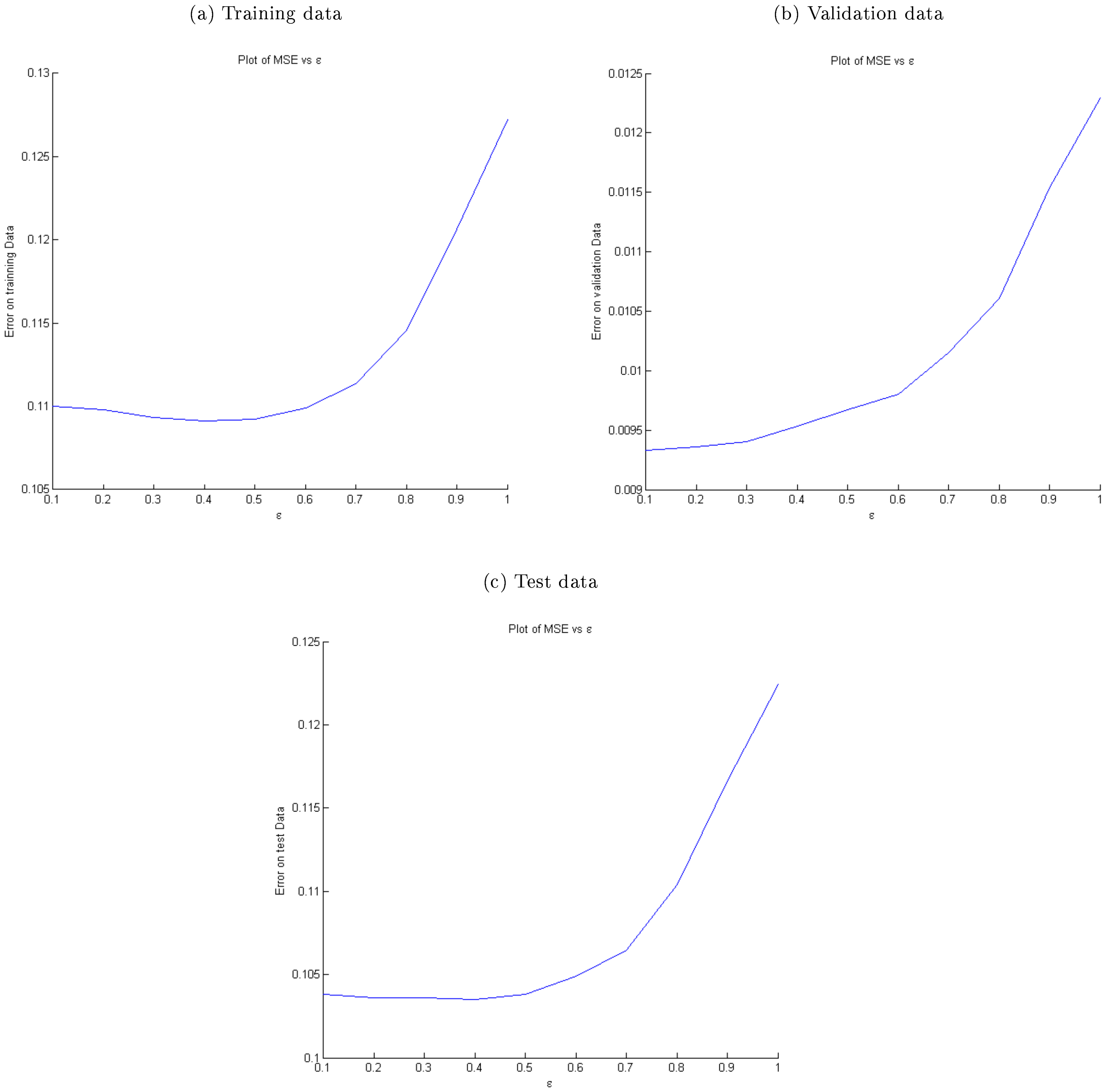
Figure 15: Plot of $\epsilon - tube$, target output and approximated function



(a) $\epsilon - SVR$ model

(b) $\nu - SVR$ model

Figure 16: MSE vs ϵ for training, test and validaton data for $\epsilon - SVR$

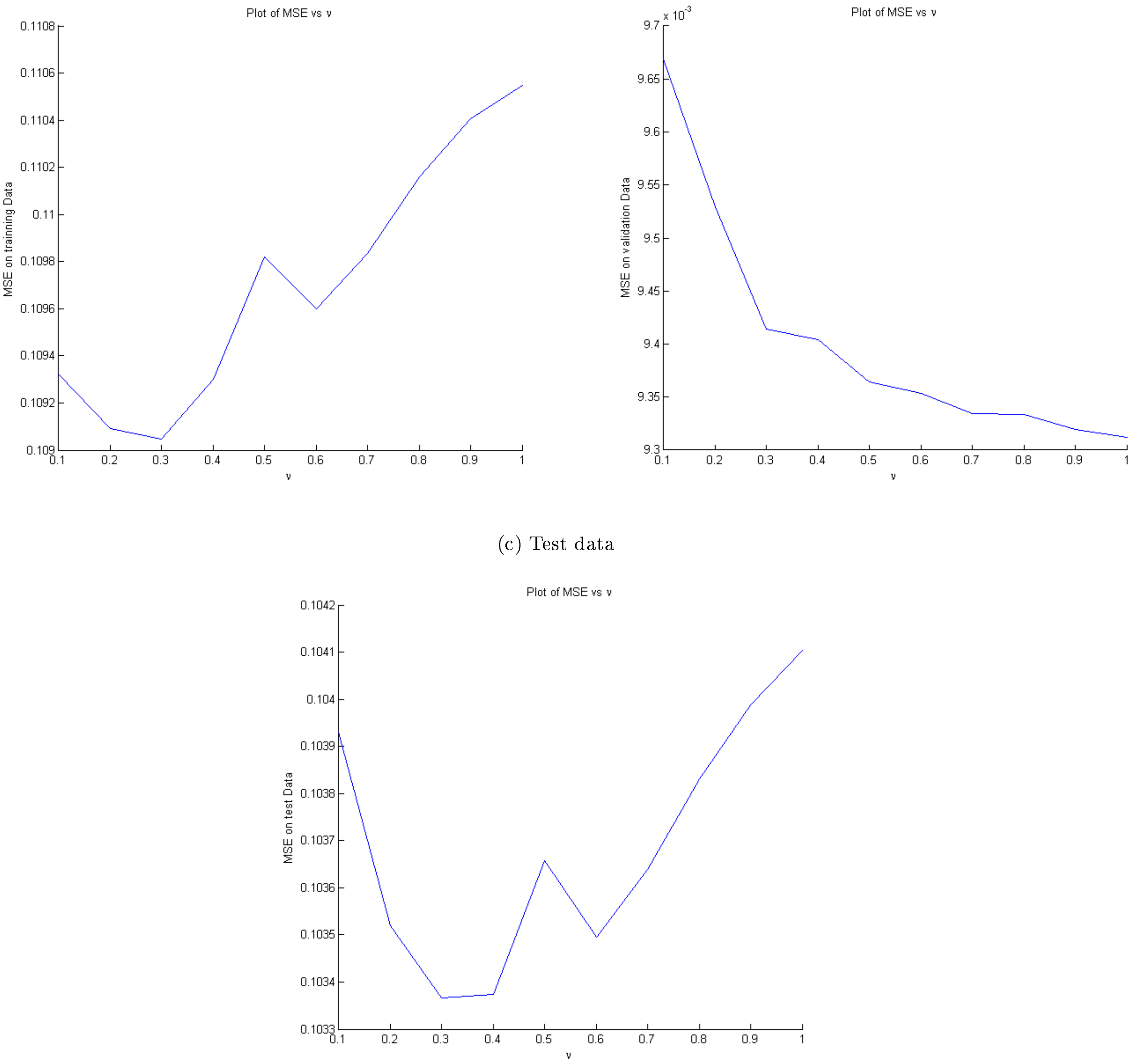


(a) Training data

(b) Validation data

(c) Test data

Figure 17: MSE vs ν for training, test and validaton data for $\nu - SVR$

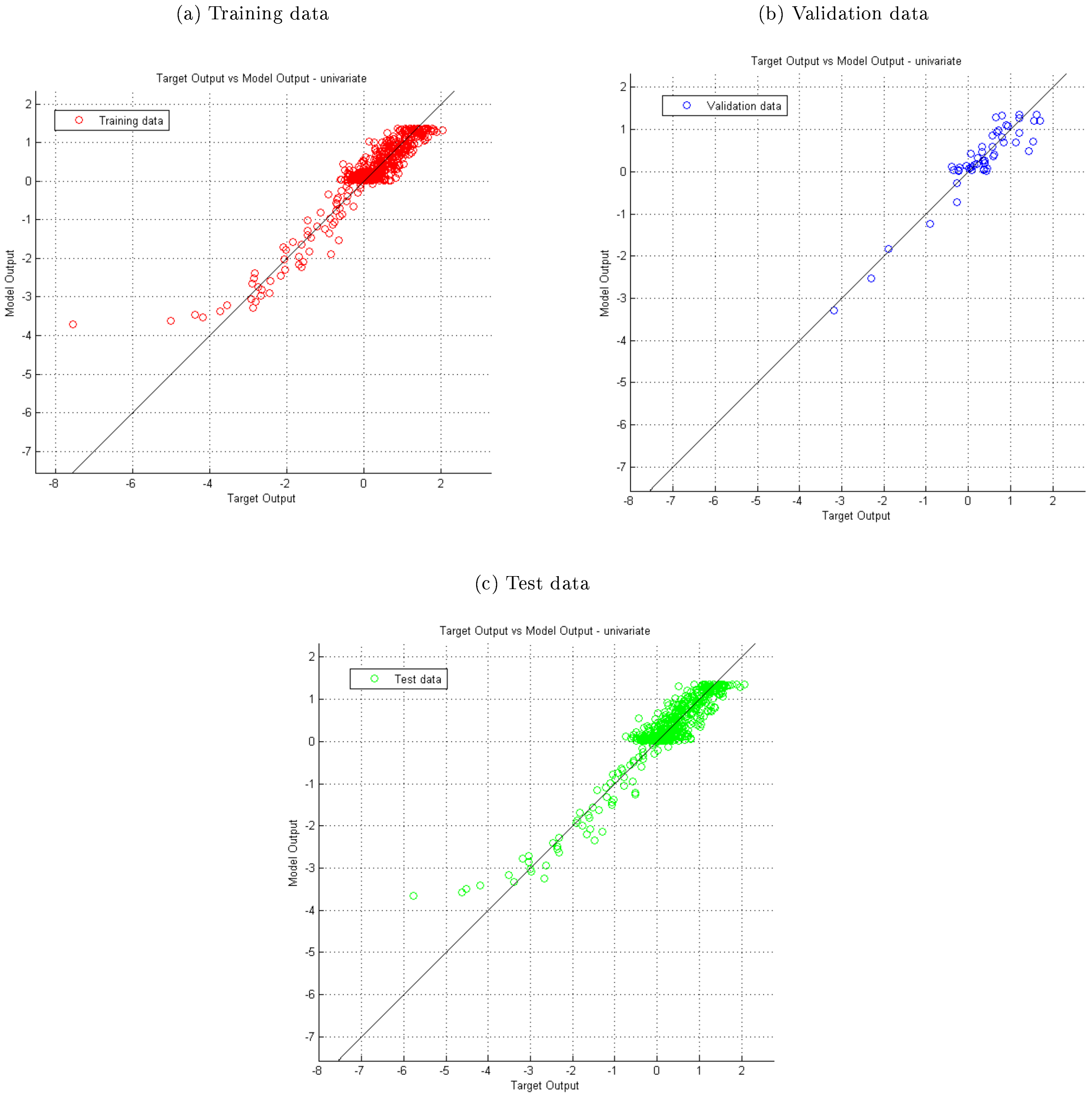


(a) Training data

(b) Validation data

(c) Test data

Figure 18: Scatter plot of model output vs target output for training, test and validaton data for $\epsilon - SVR$

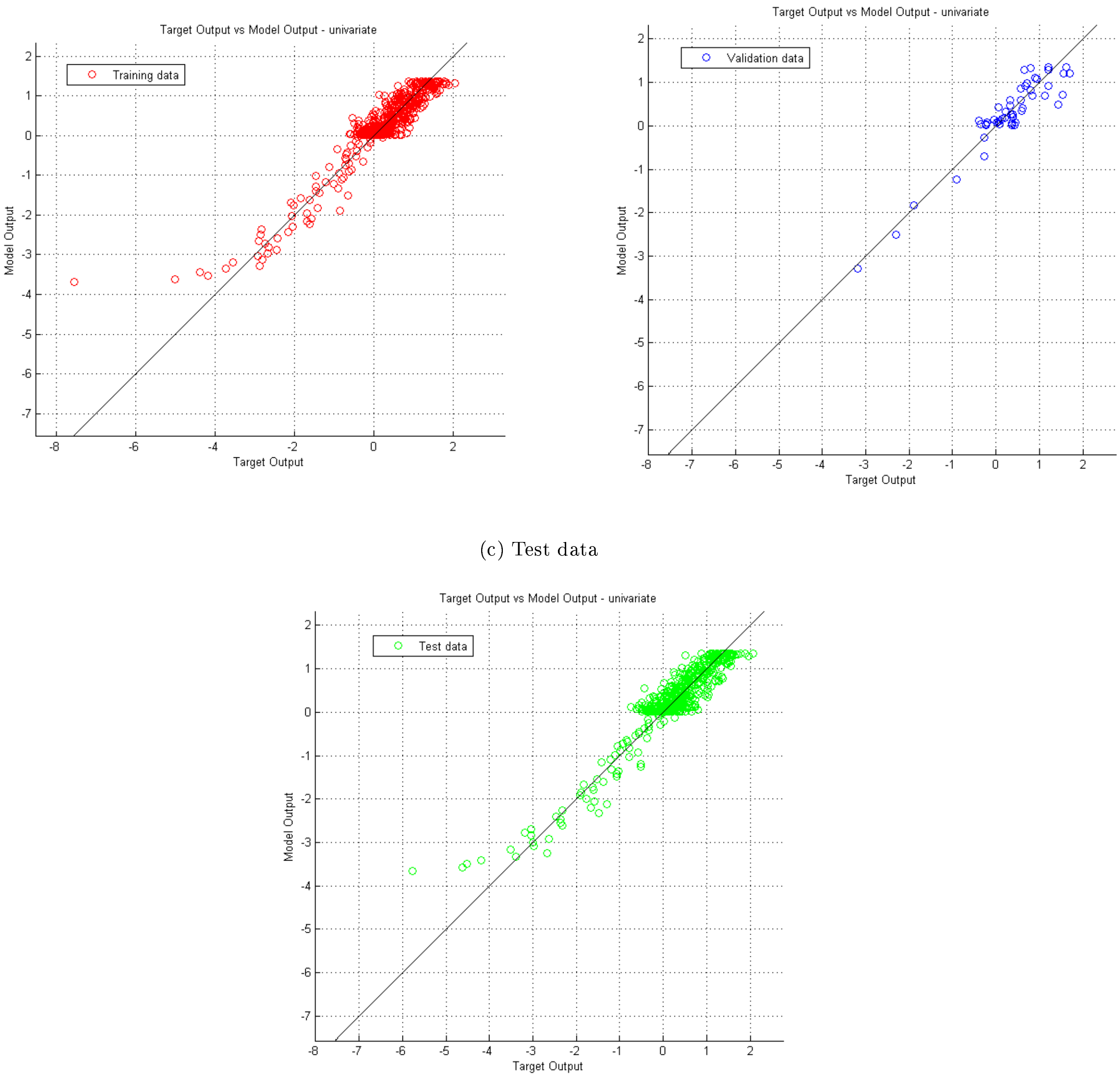


(a) Training data

(b) Validation data

(c) Test data

Figure 19: Scatter plot of model output vs target output for training, test and validaton data for $\nu - SVR$ model


(a) Training data

(b) Validation data

(c) Test data

Observations

- For the $\epsilon - SVR$ model, the mean squared error (MSE) vs ϵ in figure 16 indicates that choosing ϵ around 0.3 will give a good approximation as well as good generalization. Hence the given model was chosen.
- For the $\nu - SVR$ model, the mean squared error (MSE) vs ν in figure 17 indicates that choosing ν around 0.4 will give a good approximation as well as good generalization. Hence the given model was chosen.
- It can be seen from the scatter plots (figure 18 and figure 19) that most of the points lie in the range $[0, 1.5]$. This can easliy be explained by looking at Figure 15 where it is clear that the y-coordinate values of a large majority of the points lie in that range.

- Result of $\nu - SVR$ and $\epsilon - SVR$ are similar. This can be seen from the plots figure 15, figure 18 and figure 19.

Inferences

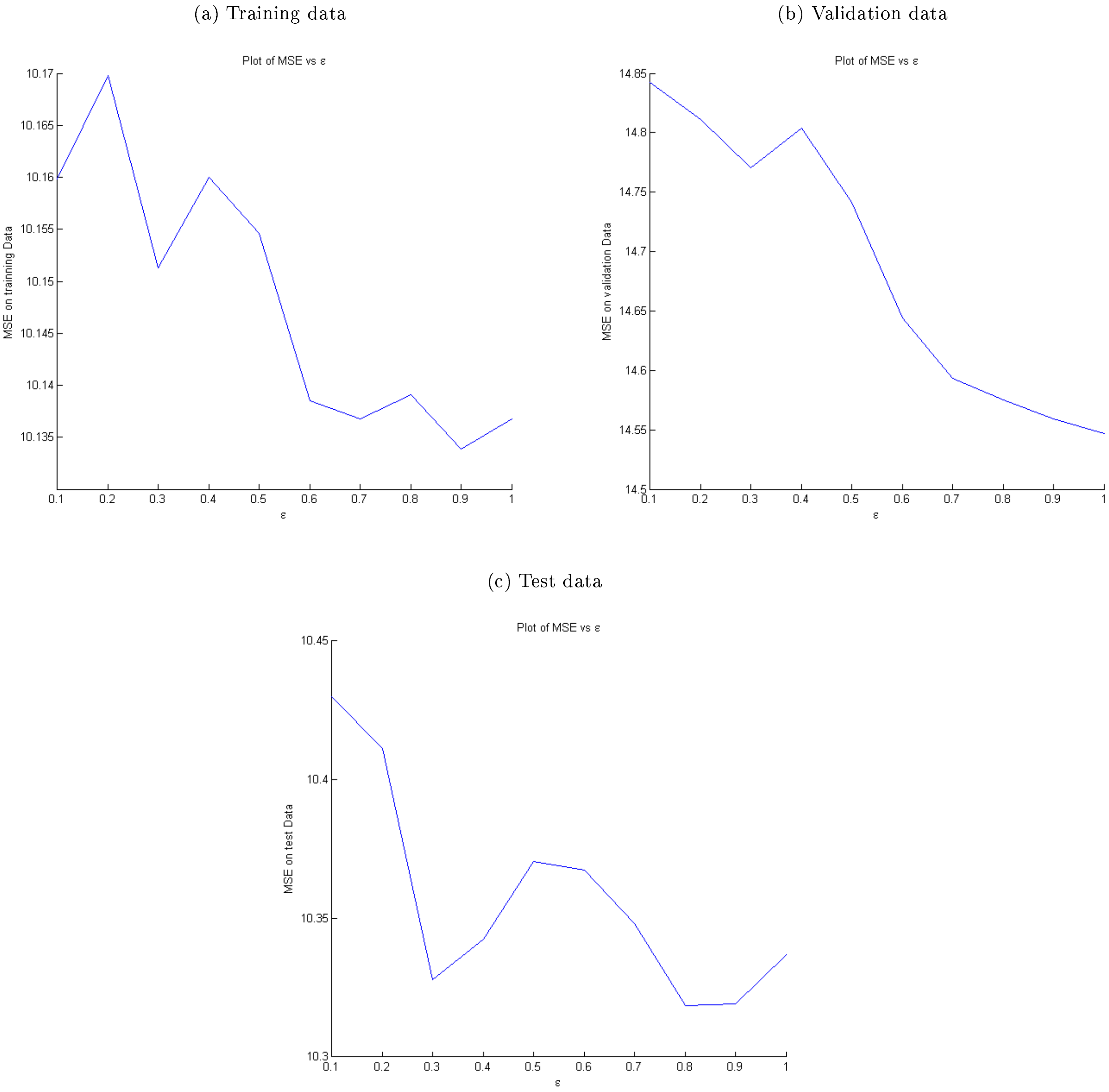
- The plots of the model and target outputs in figure 15 shows that both models performed quite well on the unseen test data. This is also borne out by the scatter plot in figure 18 and figure 19.
- Result of $\nu - SVR$ and $\epsilon - SVR$ are similar. Because the optimal solution for ϵ obtained by $\nu - SVR$ is equal to the value of ϵ set during $\epsilon - SVR$ for same cost parameter C and same inverse width of gaussian kernelγ.

5 Bivariate Data

The following models were used to perform regression on the bivariate data

- $\epsilon - SVR$ model with the following configuration parameters
  - Cost parameter $C = 15$.
  - Inverse width of the Gaussian kernel $\gamma = 9$.
  - $\epsilon = 0.74$ taken from graph MSE vs ϵ.
- $\nu - SVR$ model with the following parameters
  - Cost parameter $C = 15$.
  - Inverse width of the Gaussian kernel $\gamma = 9$.
  - $\nu = 0.7$ taken from graph MSE vs ν.

Figure 20: MSE vs ϵ for training, test and validaton data for $\epsilon - SVR$

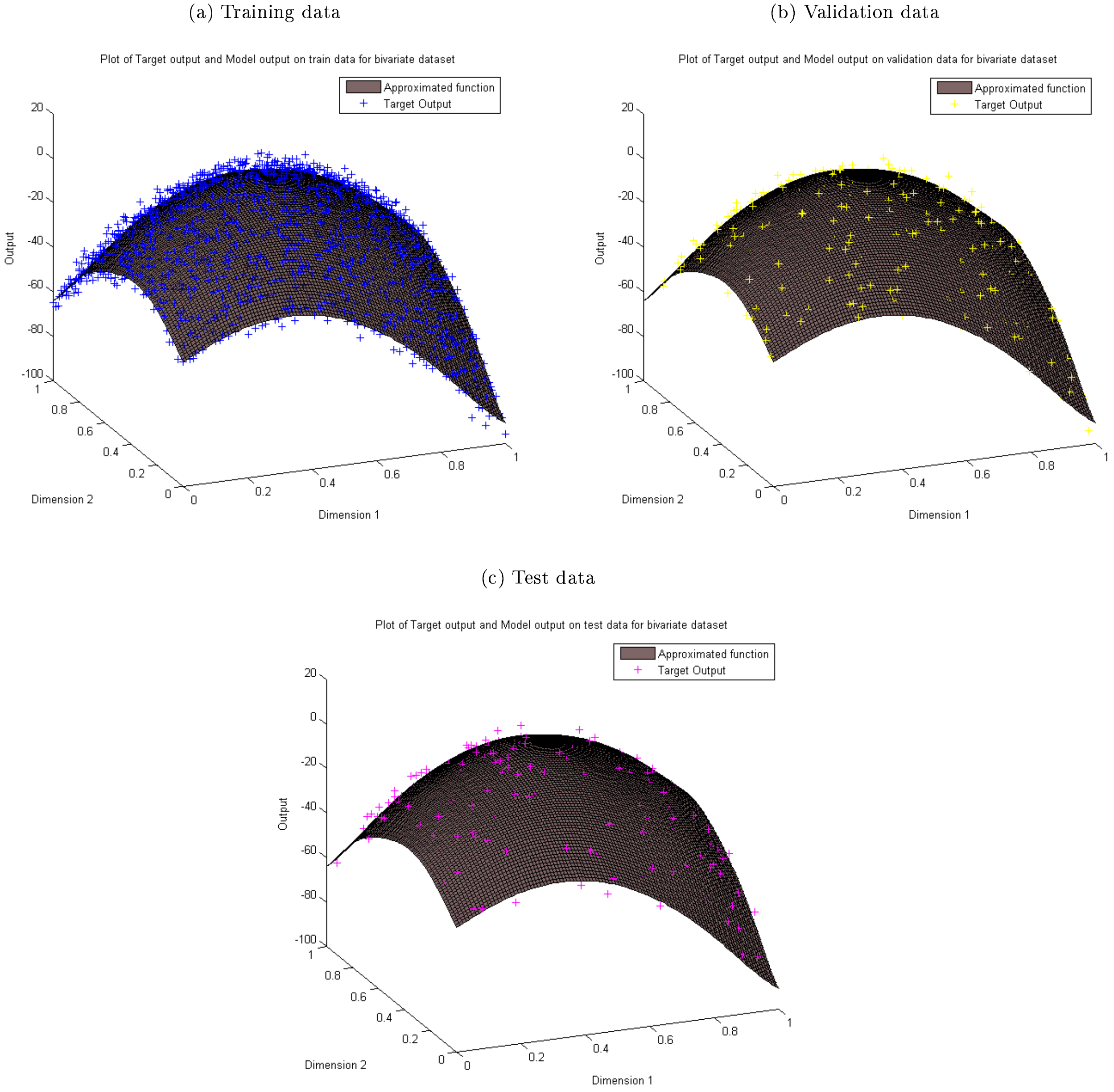


(a) Training data

(b) Validation data

(c) Test data

Figure 21: Model output and target output for training, test and validaton data for $\epsilon - SVR$

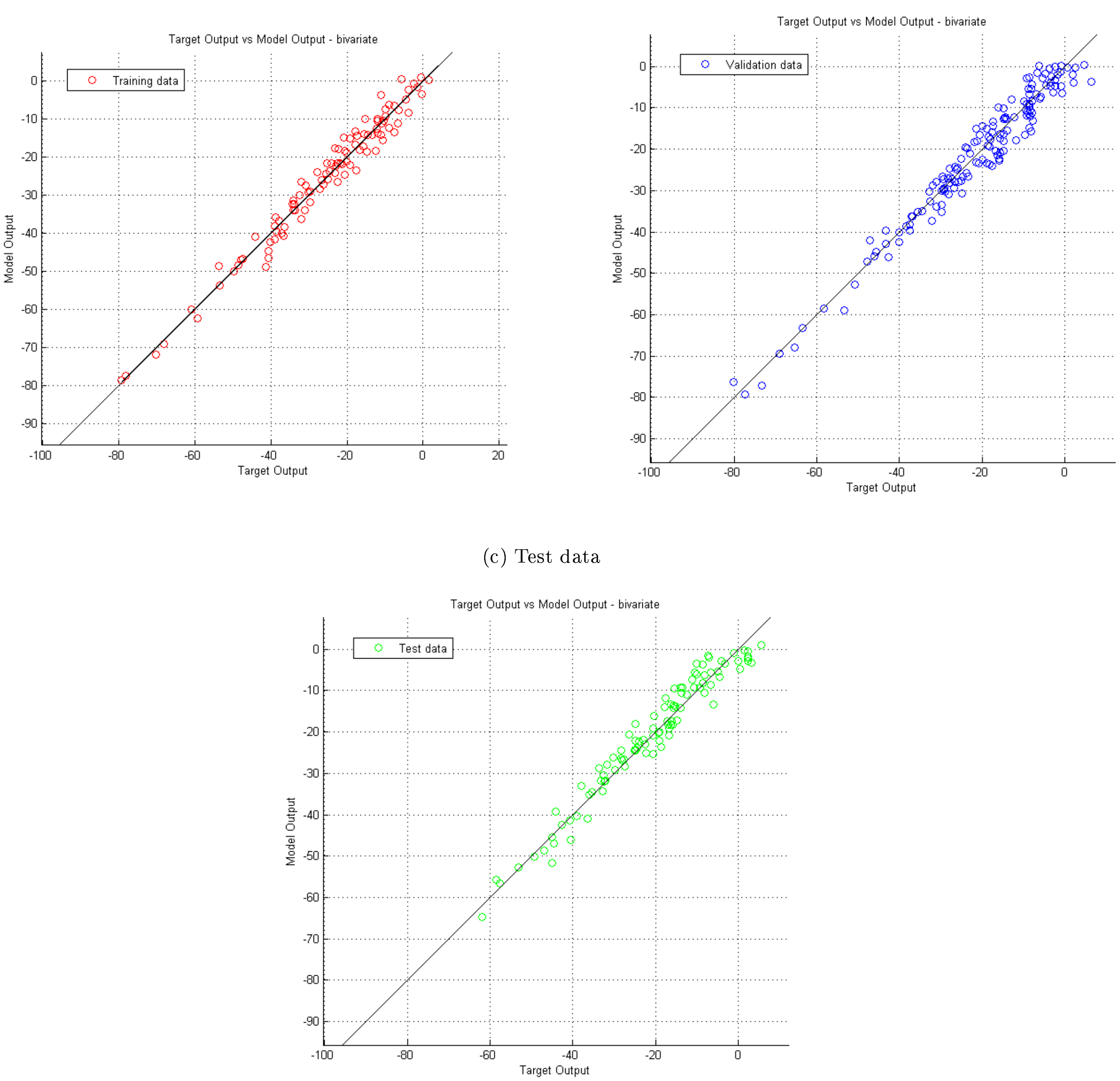


(a) Training data

(b) Validation data

(c) Test data

Figure 22: Scatter plot of model output vs target output for training, test and validaton data for $\epsilon - SVR$

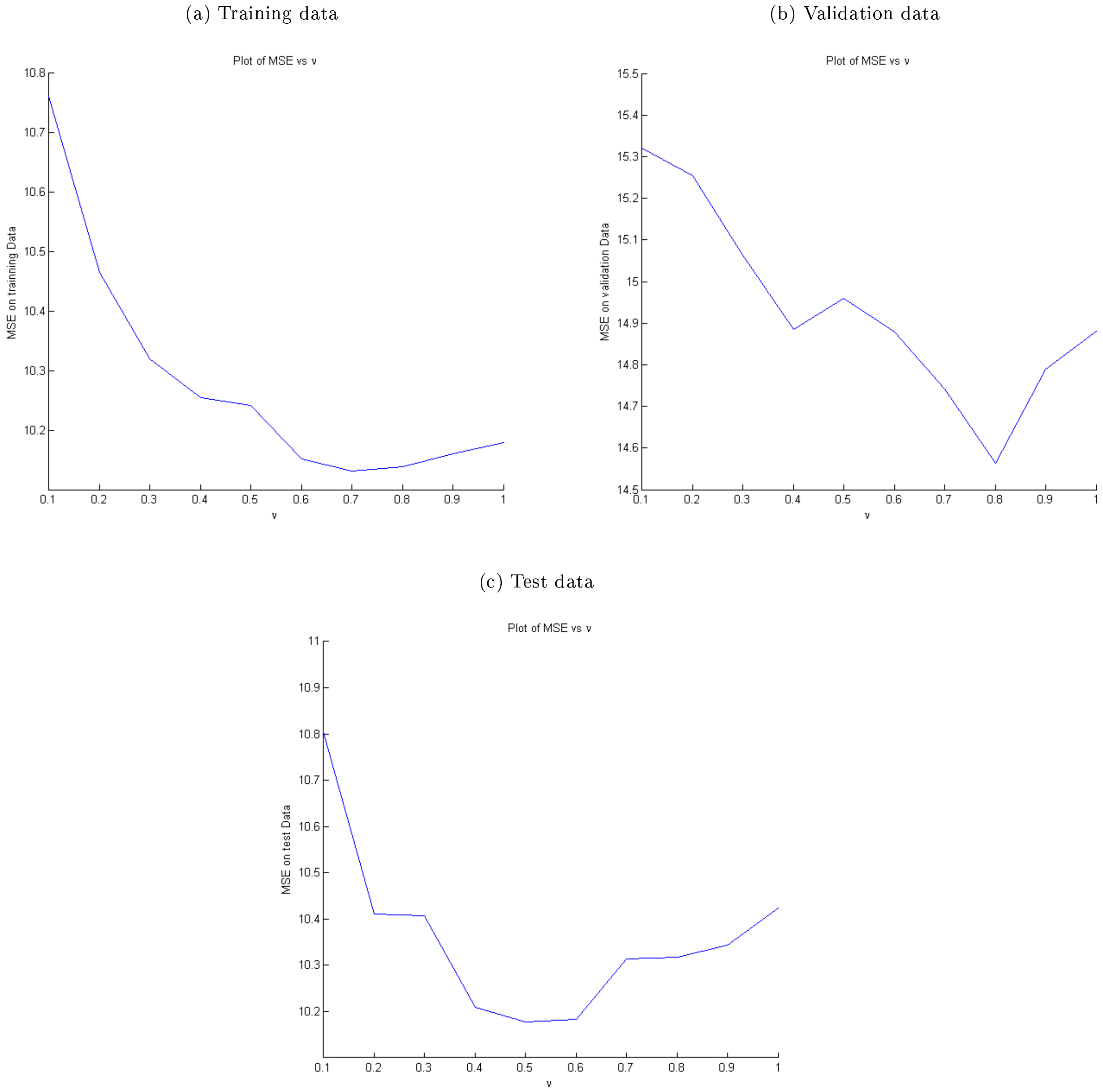


(a) Training data

(b) Validation data

(c) Test data

Figure 23: MSE vs ν for training, test and validaton data for $\nu - SVR$

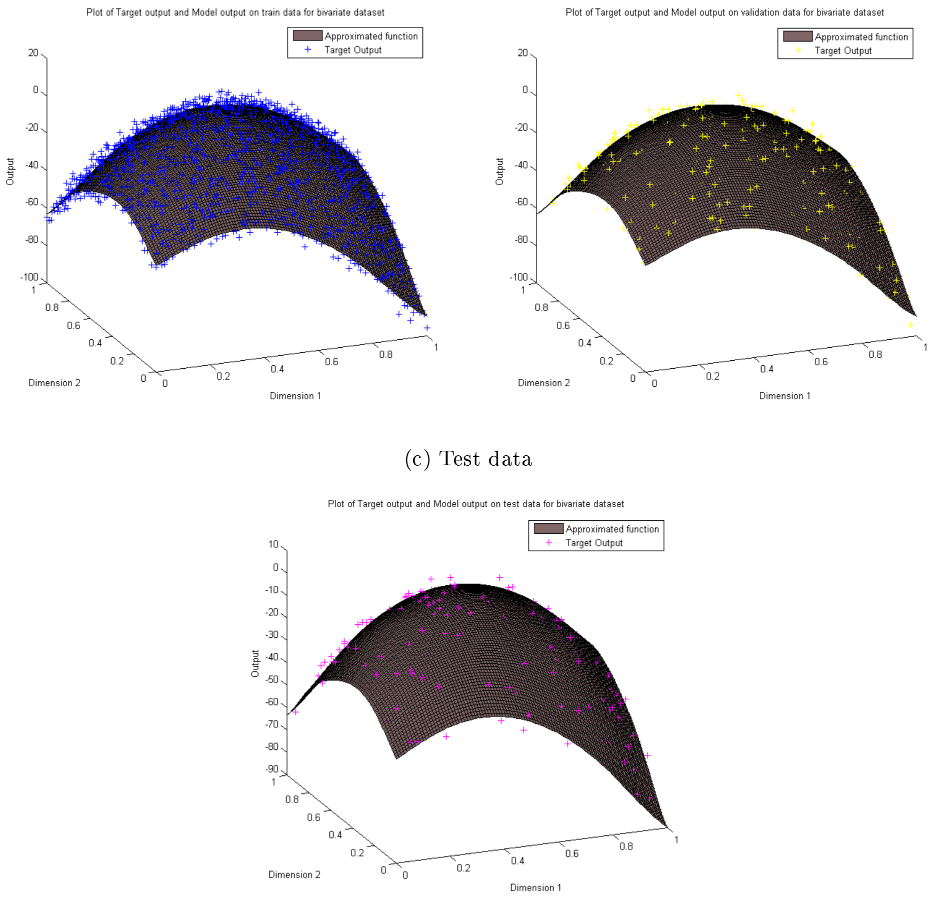


(a) Training data

(b) Validation data

(c) Test data

Figure 24: Model output and target output for training, test and validaton data for $\nu - SVR$

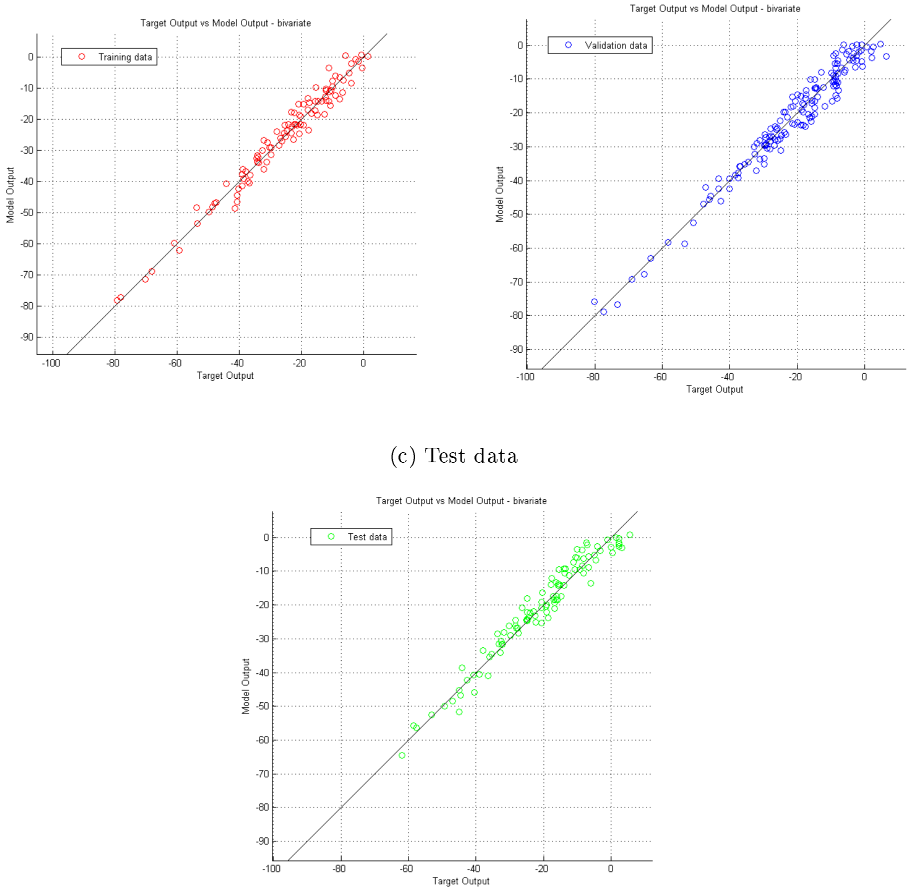


(a) Training data

(b) Validation data

(c) Test data

Figure 25: Scatter plot of model output vs target output for training, test and validaton data for $\nu - SVR$


(a) Training data

(b) Validation data

(c) Test data

Observations

- For the $\epsilon - SVR$ model, the mean squared error (MSE) vs ϵ in figure 20 indicates that choosing ϵ around 0.74 will give a good approximation as well as good generalization. Hence the given model was chosen.
- For the $\nu - SVR$ model, the mean squared error (MSE) vs ν in figure 23 indicates that choosing ν around 0.7 will give a good approximation as well as good generalization. Hence the given model was chosen.
- Result of $\nu - SVR$ and $\epsilon - SVR$ are similar. As we can see it from plots figure 21, figure 22, figure 24 and figure 25.

Inferences

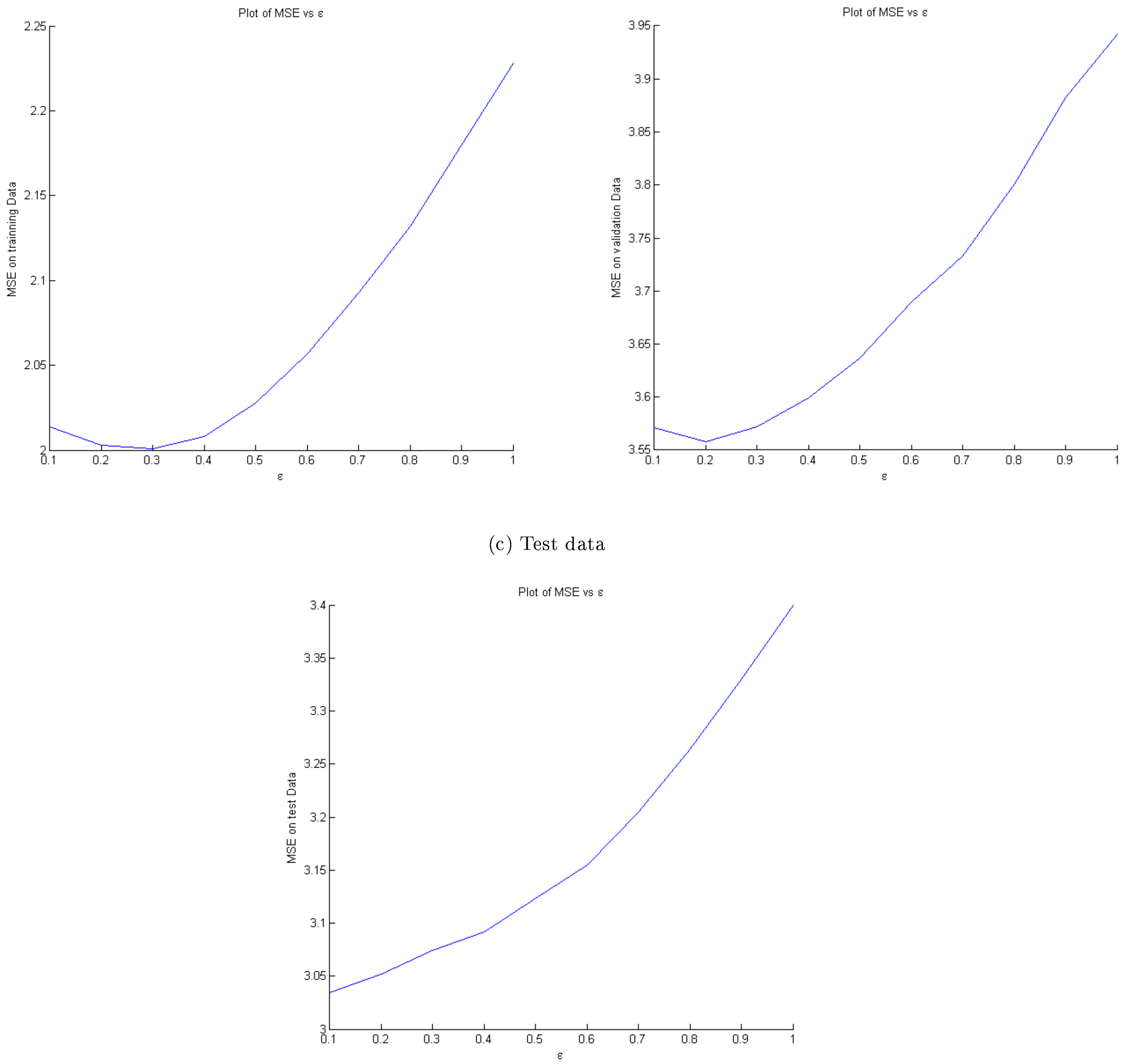
- The plots of the model and target outputs in figure 21 and figure 24 shows that both models performed quite well on the unseen test data. This is also borne out by the scatter plots in figure 22 and figure 25.
- Result of $\nu - SVR$ and $\epsilon - SVR$ are similar. Because the optimal solution for ϵ obtained by $\nu - SVR$ is equal to the value of ϵ set during $\epsilon - SVR$ for same cost parameter$C$ and same inverse width of gaussian kernelγ.

6 Multivariate Data

The following models were used to perform regression on the bivariate data

- $\epsilon - SVR$ model with the following configuration parameters
  - Cost parameter $C = 4$.
  - Inverse width of the Gaussian kernel $\gamma = 7$.
  - $\epsilon = 0.3$ taken from graph MSE vs ϵ.
- $\nu - SVR$ model with the following parameters
  - Cost parameter $C = 4$.
  - $\nu = 0.6$ taken from graph MSE vs ν.

Figure 26: Plot of MSE vs ϵ for $\epsilon - SVR$

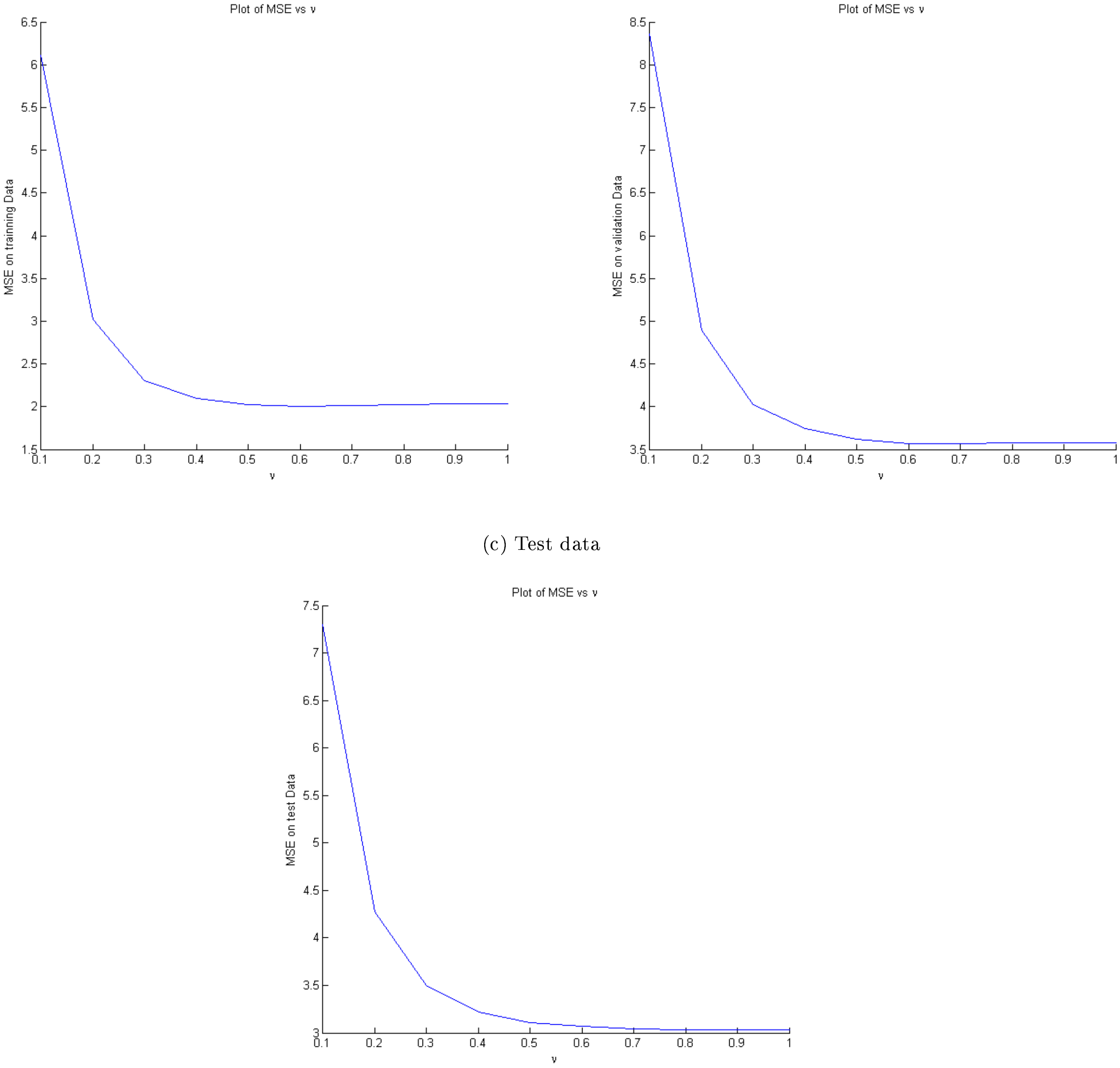


(a) Training data

(b) Validation data

(c) Test data

Figure 27: Plot of MSE vs ν for $\nu - SVR$

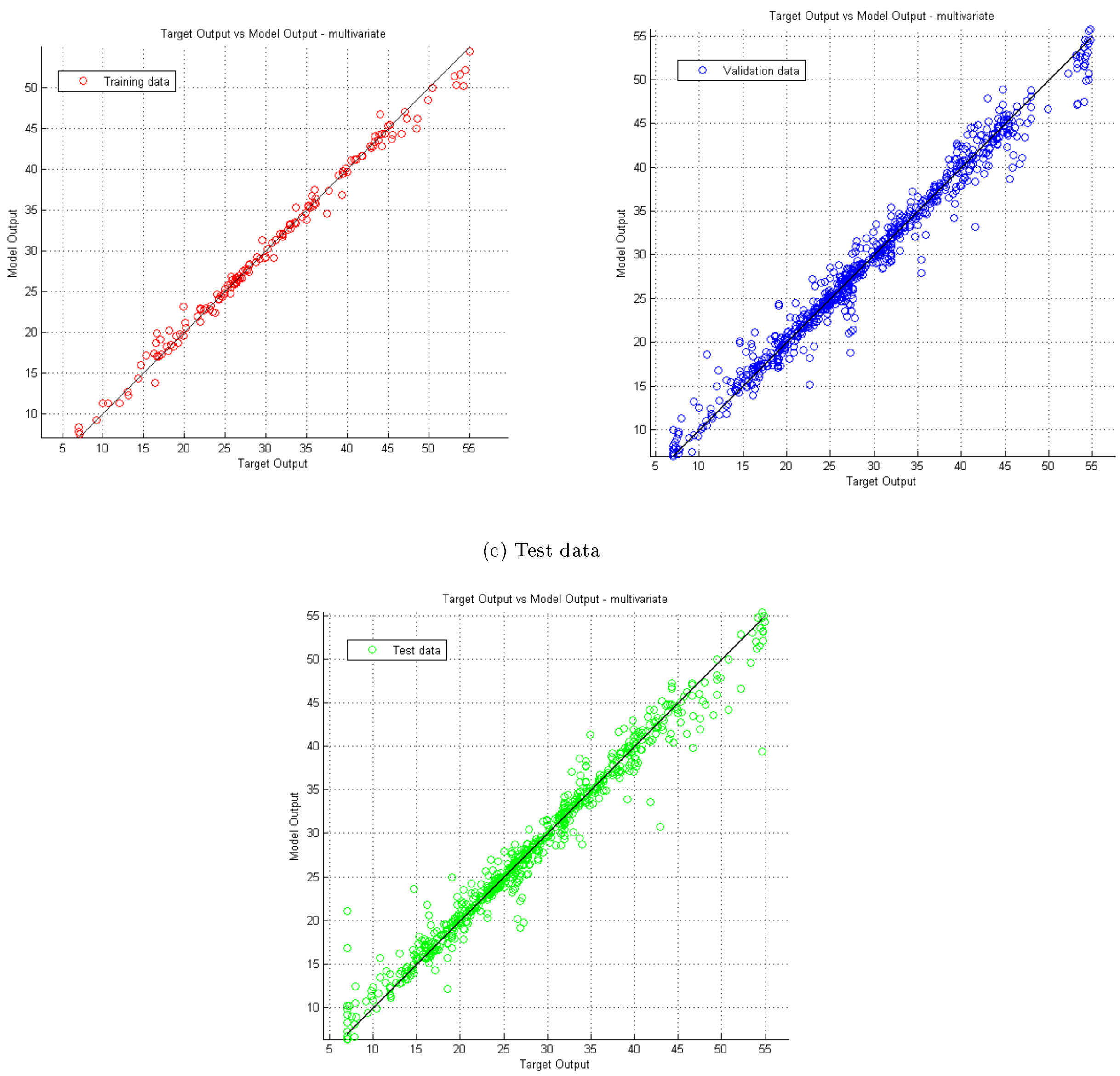


(a) Training data

(b) Validation data

(c) Test data

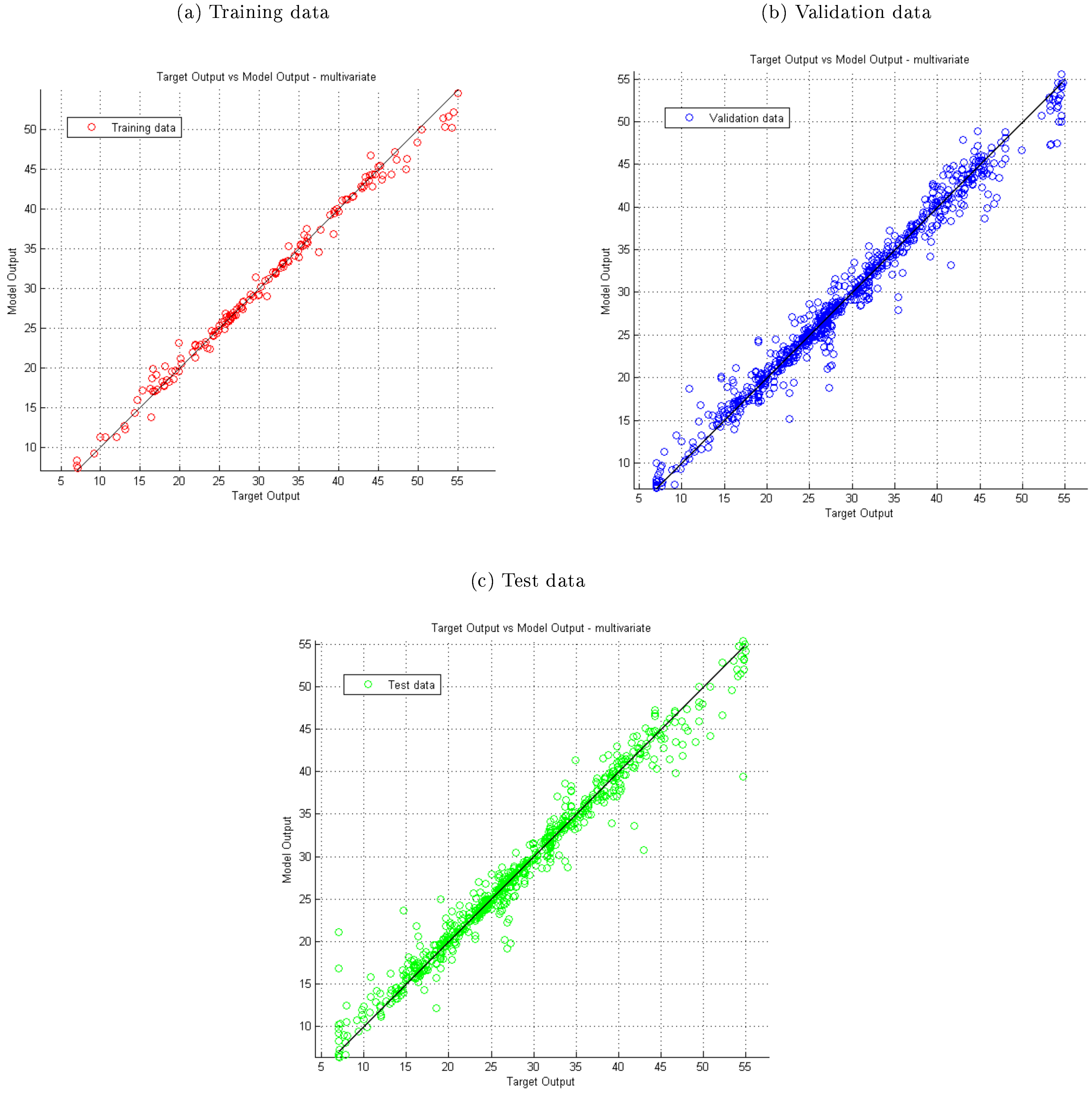
Figure 28: Scatter plot of model output vs target output for training, test and validaton data for $\epsilon - SVR$ model


(a) Training data

(b) Validation data

(c) Test data

Figure 29: Scatter plot of model output vs target output for training, test and validaton data for $\nu - SVR$ model


(a) Training data

(b) Validation data

(c) Test data

Observations

- For the $\epsilon - SVR$ model, the mean squared error (MSE) vs ϵ in figure 26 indicates that choosing ϵ around 0.3 will give a good approximation as well as good generalization. Hence the given model was chosen.
- For the $\nu - SVR$ model, the mean squared error (MSE) vs ν in figure 27 indicates that choosing ν around 0.6 will give a good approximation as well as good generalization. Hence the given model was chosen.
- Result of $\nu - SVR$ and $\epsilon - SVR$ are similar. As we can see it from plots figure 28 and figure 29.

Inferences

- The scatter plots of the model and target outputs in figure 28 and figure 29 shows that both models performed quite well on the unseen test data.
- Result of $\nu - SVR$ and $\epsilon - SVR$ are similar. Because the optimal solution for ϵ obtained by $\nu - SVR$ is equal to the value of ϵ set during $\epsilon - SVR$ for same cost parameter$C$ and same inverse width of gaussian kernelγ.

Part III

Novelty Detection

For the novelty detection tasks, after data pre-processing, we trained the model using only the data for the normal class. Since validation data was not explicitly provided, we used a 70-30 split of the train data to get the data for validation.

7 Bivariate overlapping dataset

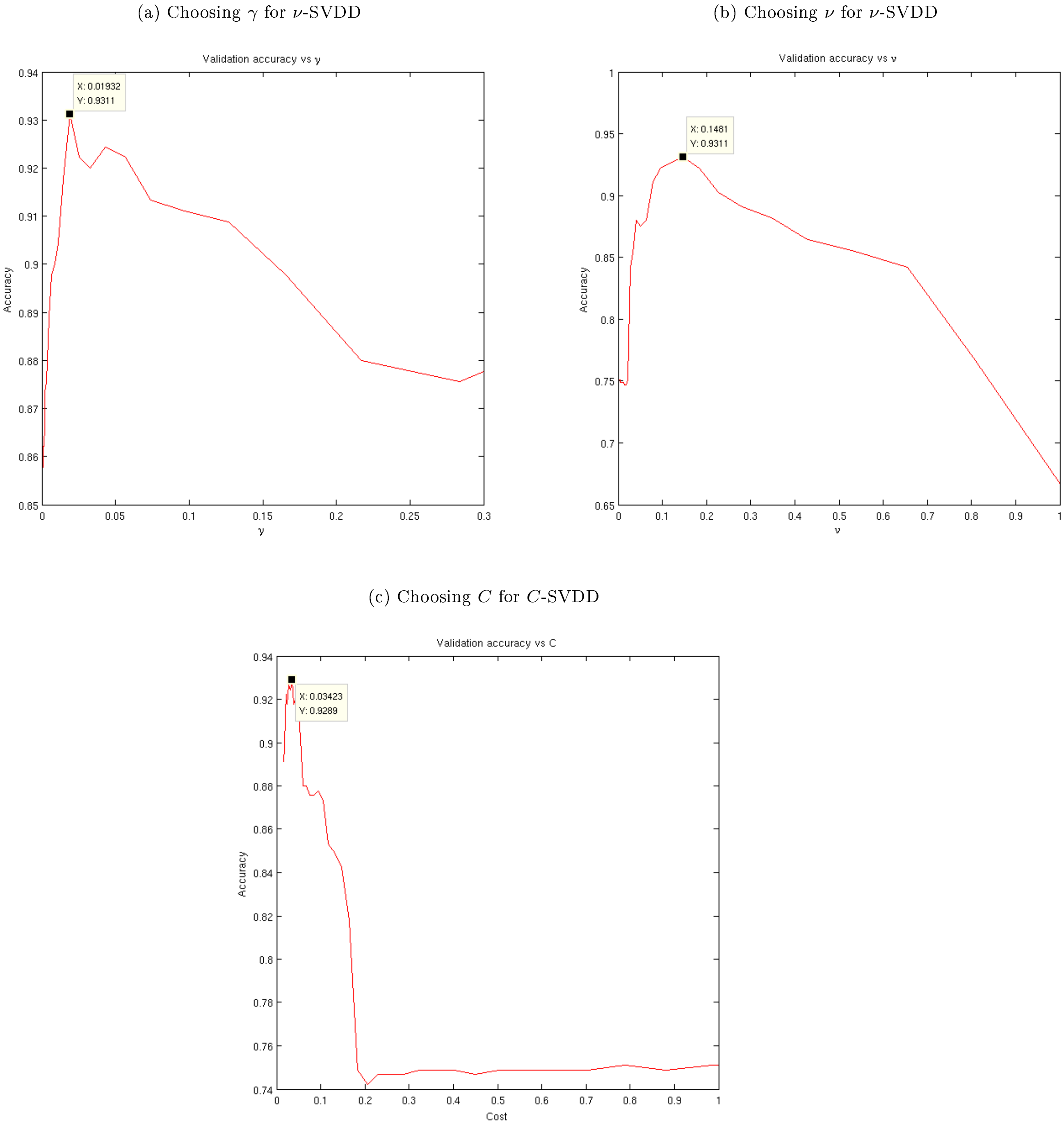
The given data consisted of three classes. Although, we experimented by treating each of the three classes as the "normal" class in turn and using only its data for training, we present here only the results obtained for one of the classes.

The following models were built for this dataset

- A C-SVDD model using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 0.0193$
  - Cost parameter $C = 0.0342$
- A ν-SVDD model using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 0.0193$ (same as in the case of C-SVM)
  - Lower bound on fraction of support vectors $\nu = 0.148$

The plots in figure 30 indicate why the above choices of parameters was made. The decision region plots obtained from each of the models are shown in figure 31. The confusion matrices for the models on test data are shown in figure 32. The ROC curves on the test data are shown in figure 33. The true positive rate and false alarm rate on training, test and validation data and some other details about the best solution obtained are tabulated in table 1.

Figure 30: Determination of different kernel parameters for different kernel functions

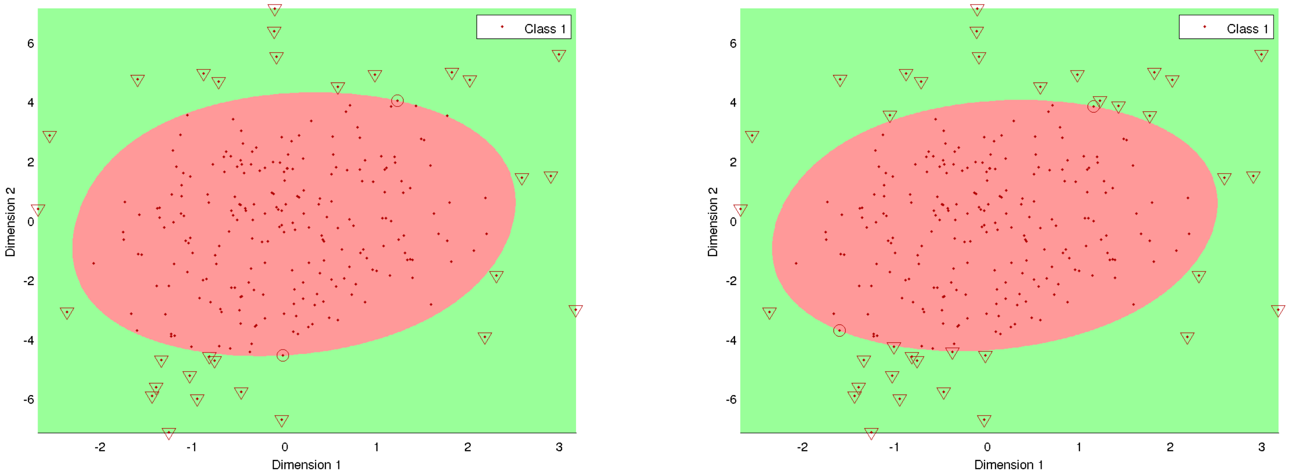


(a) Choosing γ for ν-SVDD

(b) Choosing ν for ν-SVDD

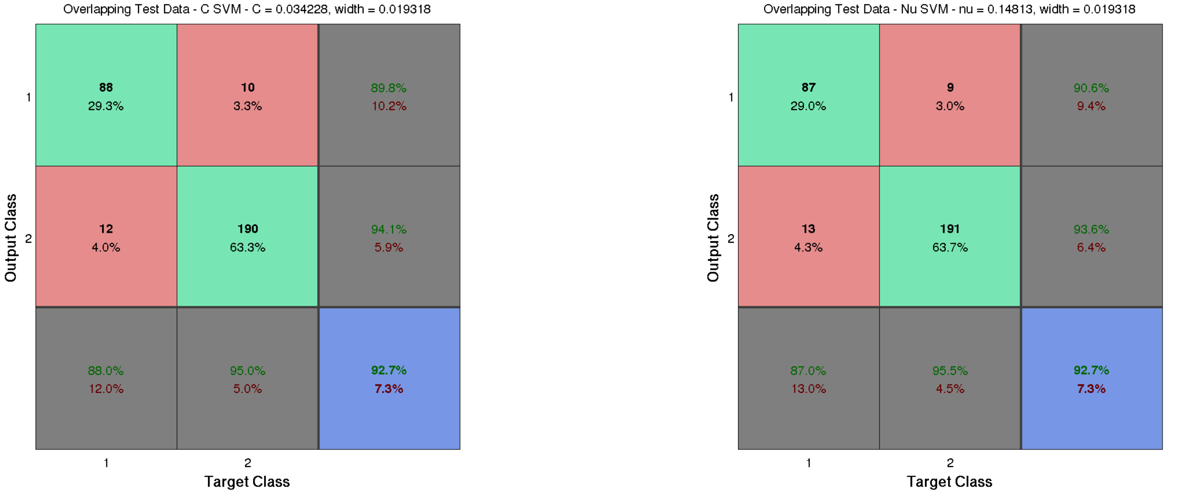
(c) Choosing C for C-SVDD

Figure 31: Decision region plots of both the models on the overlapping data


(a) Decision region for C-SVDD model

(b) Decision region for ν-SVDD model

Figure 32: Confusion matrices of both the models on the overlapping data


(a) Confusion matrix for C-SVDD model (test data)

(b) Confusion matrix for ν-SVDD model (test data)

Overlapping Test Data - C SVM - C = 0.034228, width = 0.019318

Output Class	1	2	
1	88 29.3%	10 3.3%	89.8% 10.2%
2	12 4.0%	190 63.3%	94.1% 5.9%
	88.0% 12.0%	95.0% 5.0%	92.7% 7.3%

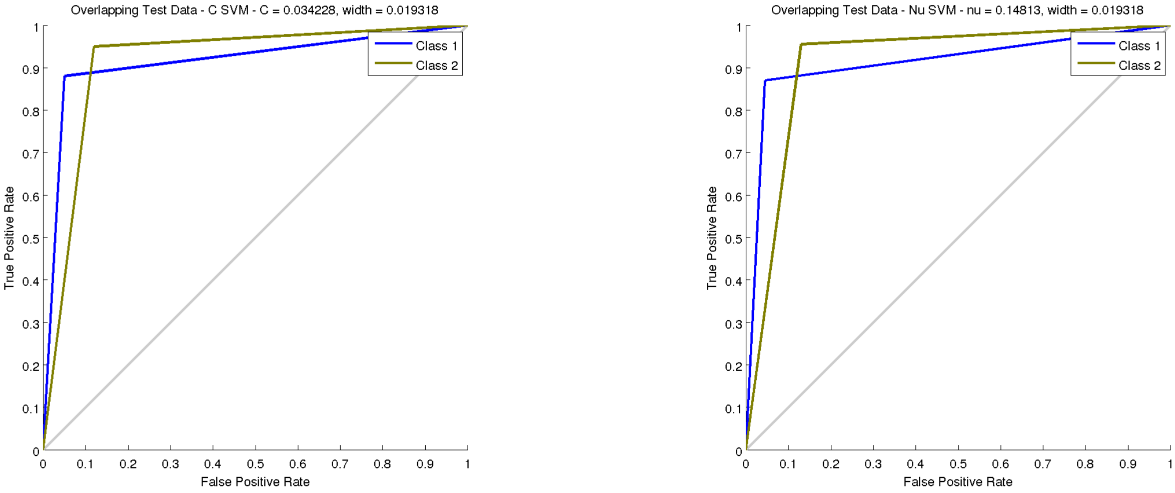
Target Class

Overlapping Test Data - Nu SVM - nu = 0.14813, width = 0.019318

Output Class	1	2	
1	87 29.0%	9 3.0%	90.6% 9.4%
2	13 4.3%	191 63.7%	93.6% 6.4%
	87.0% 13.0%	95.5% 4.5%	92.7% 7.3%

Target Class

Figure 33: Receiver operating characterstics of both the models on the overlapping data



(a) ROC of C-SVDD model (test data)

(b) ROC of ν-SVDD model (test data)

Table 1: Results and number of support vectors for the best solutions obtained using the two models on the overlapping dataset

		C-SVDD	ν-SVDD
Train data	TPR	87.20	84.80
	FAR	5.00	3.60
	Accuracy	92.40	92.53
Validation data	TPR	88.00	86.67
	FAR	4.67	3.67
	Accuracy	92.89	93.11
Test data	TPR	88.00	87.00
	FAR	5.00	4.50
	Accuracy	92.67	92.67
# bounded support vectors		29	36
# unbounded support vectors		2	2

Observations

- Both the models essentially give equivalent performance. The decision region plots (figure 31) for the two models are nearly identical. Any minor differences that remain can be removed by tuning the model parameters further.
- It was observed that tuning the parameters of the ν-SVDD model is easier than for the C-SVDD model.

Inferences

- The one-class classifier models are able to define the boundaries of the given class very well. Thus, they are able to identify outliers in the data. However, the model parameters must be chosen carefully to ensure good generalization ability.

8 Multivariate dataset

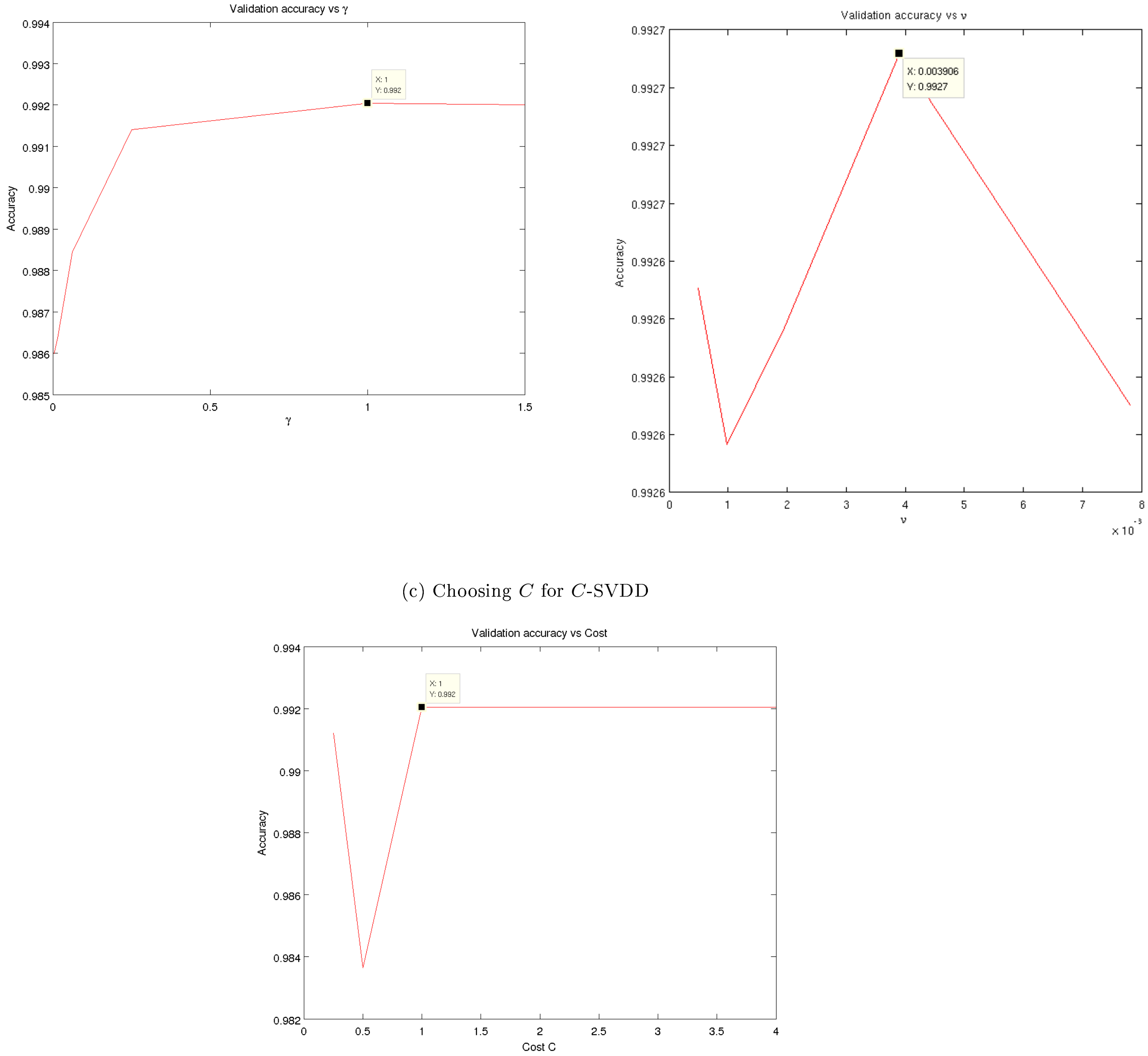
For the novelty detection tasks, after data pre-processing, we trained the model only using the data for the normal class. Since validation data was not explicitly provided, we used a 70-30 split of the train data to get the data for validation. Initially, we used all the data of the abnormal classes provided to us in the "train" dataset for validation. However, this slowed the program down and did not give appreciably better results and therefore, we created a 70-30 split of the abnormal class data as well, discarding the 70% and using only the 30% for validation.

The following models were built for this dataset,

- A C-SVDD model using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 1$
  - Cost parameter $C = 1$
- A ν-SVDD model using the Gaussian kernel with the following configuration,
  - Inverse width of the Gaussian kernel $\gamma = 1$ (same as in the case of C-SVM)
  - Lower bound on fraction of support vectors $\nu = 0.0039$

The plots in figure 34 indicate why the above choices of parameters was made. The confusion matrices for the models on test data are shown in figure 36. The ROC curves on the test data are shown in figure 35. The true positive rate and false alarm rate on training, test and validation data and some other details about the best solution obtained are tabulated in table 2.

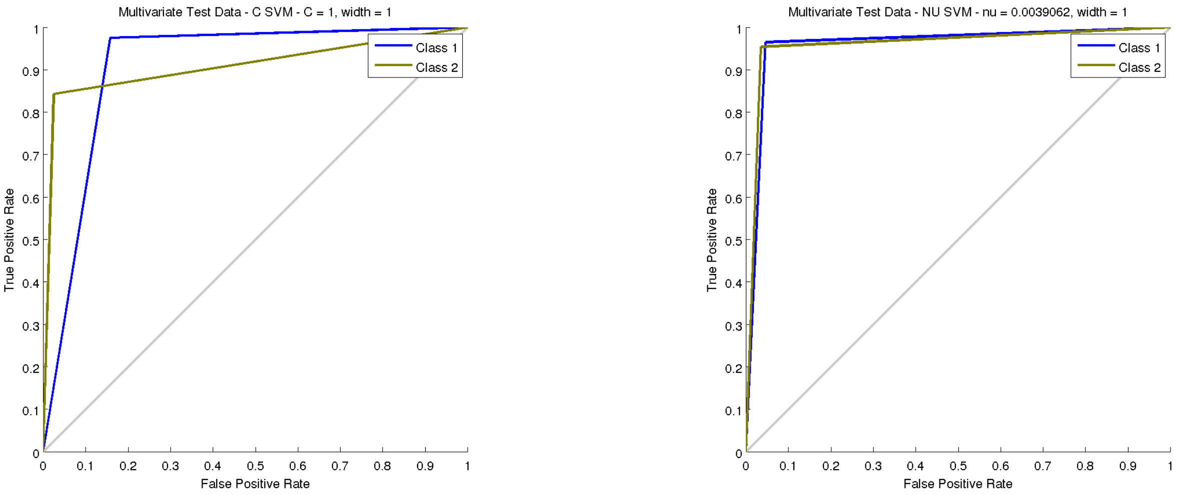
Figure 34: Determination of different kernel parameters for different kernel functions


(a) Choosing γ for C-SVDD

(b) Choosing ν for ν-SVDD

(c) Choosing C for C-SVDD

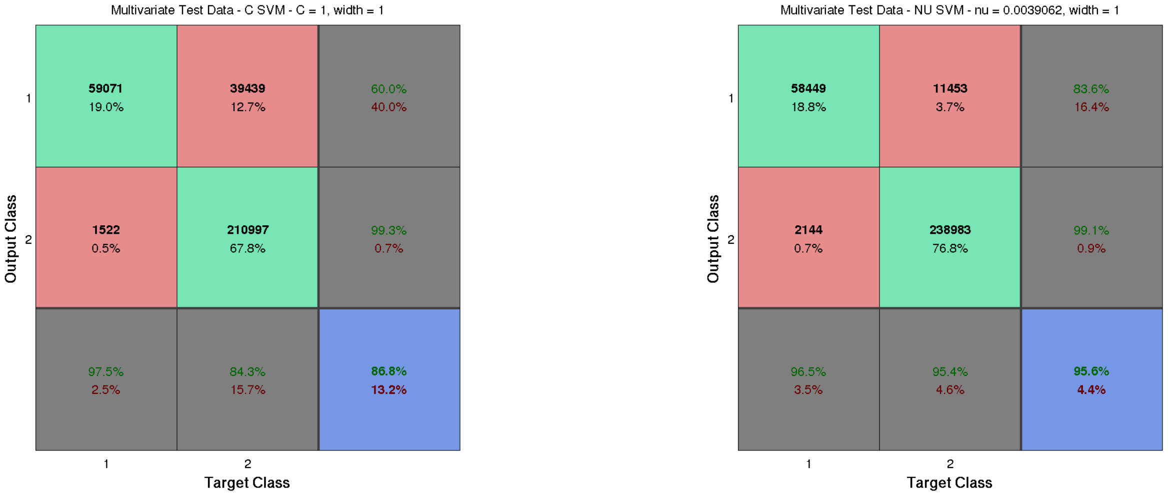
Figure 35: Receiver operating characterstics of both the models on the multivariate data



(a) ROC of C-SVDD model (test data)

(b) ROC of ν-SVDD model (test data)

Figure 36: Confusion matrices of both the models on the multivariate data



(a) Confusion matrix for C-SVDD model (test data)

(b) Confusion matrix for ν-SVDD model (test data)

Table 2: Results and number of support vectors for the best solutions obtained using the two models on the multivariate dataset

		C-SVDD	ν-SVDD
Train data	TPR	99.29	99.57
	FAR	0.79	0.79
	Accuracy	99.22	99.28
Validation data	TPR	99.10	99.45
	FAR	0.77	0.78
	Accuracy	99.20	99.27
Test data	TPR	97.49	96.46
	FAR	15.75	4.57
	Accuracy	86.83	95.63
# bounded support vectors		0	113
# unbounded support vectors		320	311

Observations

- The one-class classifiers in the case of both the C-SVDD and the ν-SVDD give fairly good results. The ν-SVDD, in particular, performs extremely well with the appropriate choice of parameters.
- While the C-SVM is able to get nearly all examples of the normal class right (very few false negatives), the false positive rate is quite high. It should be noted that since the number of examples of the normal class is smaller than all the examples of the abnormal classes combined a false positive rate of 15% implies that the number of false positives and the number of true positives is of the same order.
- The false positive rate on both the train and validation data is very low but that on the test data is much higher.
- Choosing parameters in case of the ν-SVDD was seen to be easier than in the case of the C-SVDD.

Inferences

- Since the models use the data of only one class, they do not have access to discriminatory information between the different classes. The only place where data of the other class, if available, is made use of is to tune parameters during validation. This is evident in the very good performance of the model on the train and validation data in both the cases.
- It should be noted that determining the parameters that give the best performance on the validation data does not result in equivalent preformance on the test data. This seems to reflect the fact that the test data has some characteristics that are not present in the training data (and the validation data, which was drawn from the training data). Thus, even though the validation data is used only for tuning parameters and not for training, very good performance on the validation data does not translate into as good performance on the test data. This is particularly true in the case of the C-SVM.
- It was seen that there is a trade-off between the true positive rate and the false alarm rate. For example, setting $\gamma = 0.8$ in case of the C-SVM instead of 1 improves the false alarm rate to 14% but at cost of reducing the true positive rate to 88%.

References

[BOV] Annalisa Barla, Francesca Odone, Alessandro Verri, *Histogram Intersection Kernel for Image Classification.*

[DH] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification,* 2ed*,* Wiley Student Edition

[LIBSVM] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin *A Practical Guide to Support Vector Classication,* http://www.csie.ntu.edu.tw/∼cjlin/papers/guide/guide.pdf