2008-01-01

E-Quality Control: A Support Vector Machines Approach

Kalyan Reddy Aleti
University of Texas at El Paso, raj_aleti@yahoo.com

E-QUALITY CONTROL: A SUPPORT VECTOR MACHINES APPROACH

KALYAN REDDY ALETI

Department of Industrial Engineering

APPROVED:

Tzu-Liang Bill Tseng, Ph.D., Chair

Heidi A Taboada Jimenez, Ph.D.

Thompson Sarkodie-Gyan, Ph.D.

Patricia D. Witherspoon, Ph.D.
Dean of the Graduate School

E-QUALITY CONTROL: A SUPPORT VECTOR MACHINES APPROACH

by

KALYAN REDDY ALETI, B.E

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Industrial Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

December 2008

Acknowledgements

I am grateful to my advisor Dr. Bill Tseng for having faith in me and constantly supporting me in all means throughout the course of action for completing this work. Also, I like to thank Dr. Heidi Taboada for guiding me in the right direction whenever reached upon. I am thankful to Dr. Thompson Sarkodie-Gyan for his valuable comments and taking his time to review my work as a part of the thesis defense committee.

Last but not the least, I cannot end this without mentioning my family without whose support I could not have fulfilled all my goals.

Abstract

The web-enabled quality control process presents many benefits to industry, such as universal access, remote control capability, and integration of production equipment into information networks for improved efficiency. This capability has a great potential, since engineers can access and control the equipment anytime, anywhere as the design stages evolve. In this context, this work uses innovative methods in remote part tracking and quality control with the aid of the modern equipment and application of Support Vector machine learning approach to predict the outcome of the quality control process. The classifier equations are built on the data obtained from the experiments and analyzed with different kernel functions and a detailed analysis is presented for six different case studies. The results indicate the robustness of Support Vector classification for the experimental data with two output classes.

Table of Contents

List of Tables

List of Figures

Chapter 1: INTRODUCTION

1.1 Background

A current trend for manufacturing industry is shorter product life cycle, remote monitoring/control/diagnosis, product miniaturization, high precision, zero-defect manufacturing, and information-integrated distributed production systems for enhanced efficiency and product quality [1-6]. In tomorrow's factory, design, manufacturing, quality, and business functions will be fully integrated with the information management networks [7-9]. This new paradigm is coined with the term, e-manufacturing**.** One of the enabling tools to realize the e manufacturing is the ability to predict the variations and performance loss. Therefore, Internet-based gauging, measurement, inspection, diagnostic system, and quality control have become critical issues in the integration with e-manufacturing systems and management. For manufacturing industry, the current emphasis on quality, reliability, and the competitive state of the international/domestic markets have resulted in both greater visibility and increased responsibility for quality and inspection. According to the white correspondences from the American Society for Quality and the U.S. Department of Labor, 2004–2005 Edition on Bureau of Labor Statistics, increased emphasis has been placed on quality control in manufacturing, while inspection is more fully into production processes than in the past. Many companies have integrated teams of inspection and production workers to jointly review and improve product quality as they are seeking to implement completely automated inspection with the help of advanced sensors installed at one or several points in the production process. In manufacturing, quality control is fundamental to ascertain the conformance, and plays an important role in deciding whether the parts are being manufactured according to the design specifications and

whether the manufacturing processes are in control. The recent progress in developing new, automated production and measuring instrument has led to the 100% real-time inspection, where critical dimensions are measured and verified while parts are being produced. The immediate benefit from this approach is the reduction of manufacturing cost by preventing further processing of defective parts along the manufacturing stages. More importantly, the remote accessibility and the ability to control the equipment over the Internet/ Ethernet/LAN present unprecedented benefits to the current manufacturing environment. Designers located remotely from the production facility can carry out inspections and quality check as their design processes evolve. The quality control and the process capability analysis can be tested and adjusted according to the assembly and manufacturing specifications.

1.2 Development of internet based inspection systems

As with other manufacturing processes, robotic assembly operations require an integrated quality control routine in each stage of the assembly. This ensures that no defective parts will be propagated into the downstream, hence reducing the manufacturing costs associated with bad quality. The Yamaha YK 250X SCARA (selective compliance assembly robot arm) robot can be controlled remotely through an onboard Ethernet card, which is an optional device for connecting the robot controller over the Internet. The communications protocol utilizes TCP/IP (Transmission Control Protocol/Internet Protocol), which is a standard Internet Protocol. The unit uses 10BASE-T specifications and UTP cables (unshielded twisted-pair) or STP cables (shielded twisted-pair) can be used. PCs with Internet access can exchange data with the robot controller using Telnet (Figure 1). Once the connection is established, commands (e.g., speed, read and load program, turn motor on and off, move to points, etc.) can be sent to the controller

to achieve desired results. The Telnet procedure has been included in the Visual Basic codes to develop an application program interface (API), which improves the visualization of robot operation with the intuitive interface along with the enhanced functionality. The connection between the API and the system was established by the utilization of Winsock components and various ActiveX controls that communicate through IP addresses. The overall setting of the system is presented in the figure 1.1 shown below.

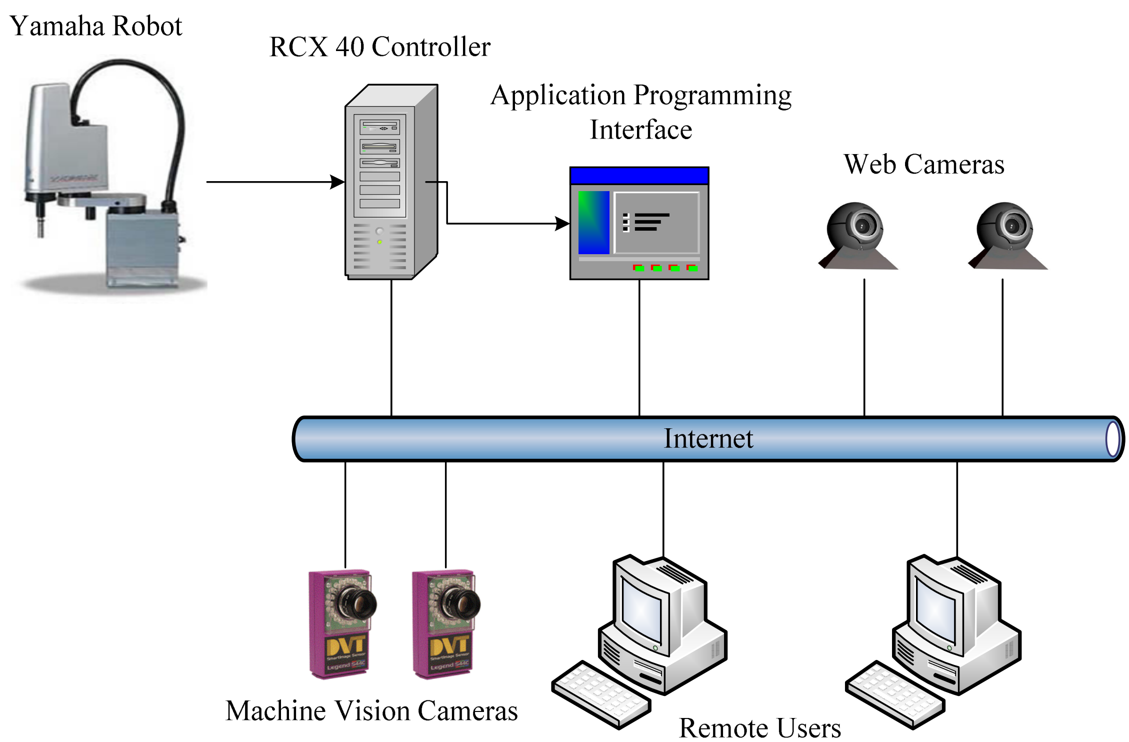


Figure 1.1: Overall Setting of the System

The captured image has 640x480 pixel size and is transferred to the API through the Internet. Any given time, 75 frames were captured and processed per second. The DVT camera acts like a server, accessible from anywhere. Locating an object is done by using a pattern matching technique. Before the actual process began, the pattern of an object was saved in the flash

memory of the camera. An image of the object was analyzed and its characteristic profile is generated. During the process, the pattern of the object is matched with the one already stored in the camera memory. If the object pattern matches, it sends a PASS event to the API through TCP/IP. When the API receives the PASS criteria, the conveyor belt stops automatically, and the object location in terms of camera coordinate system is determined by the vision algorithm. The mathematical processes in the API map this coordinate into the SCARA Cartesian coordinate system. It compiles the location coordinate into the robot command sequence and sends it to the robot. The command makes the robot to go to the position of the object, turn the vacuum gripper on, pick up the object, and place it at the desired location. This completes one cycle of quality check and the entire system waits again for another object. If the object fails the inspection, no operation is done. The object continues to be carried and stocked at the end of the conveyor belt. The quality check of remote robotic operation has been successfully tested and the entire process was monitored through the web camera. The picture of entire setup is shown in the figure 1.2.

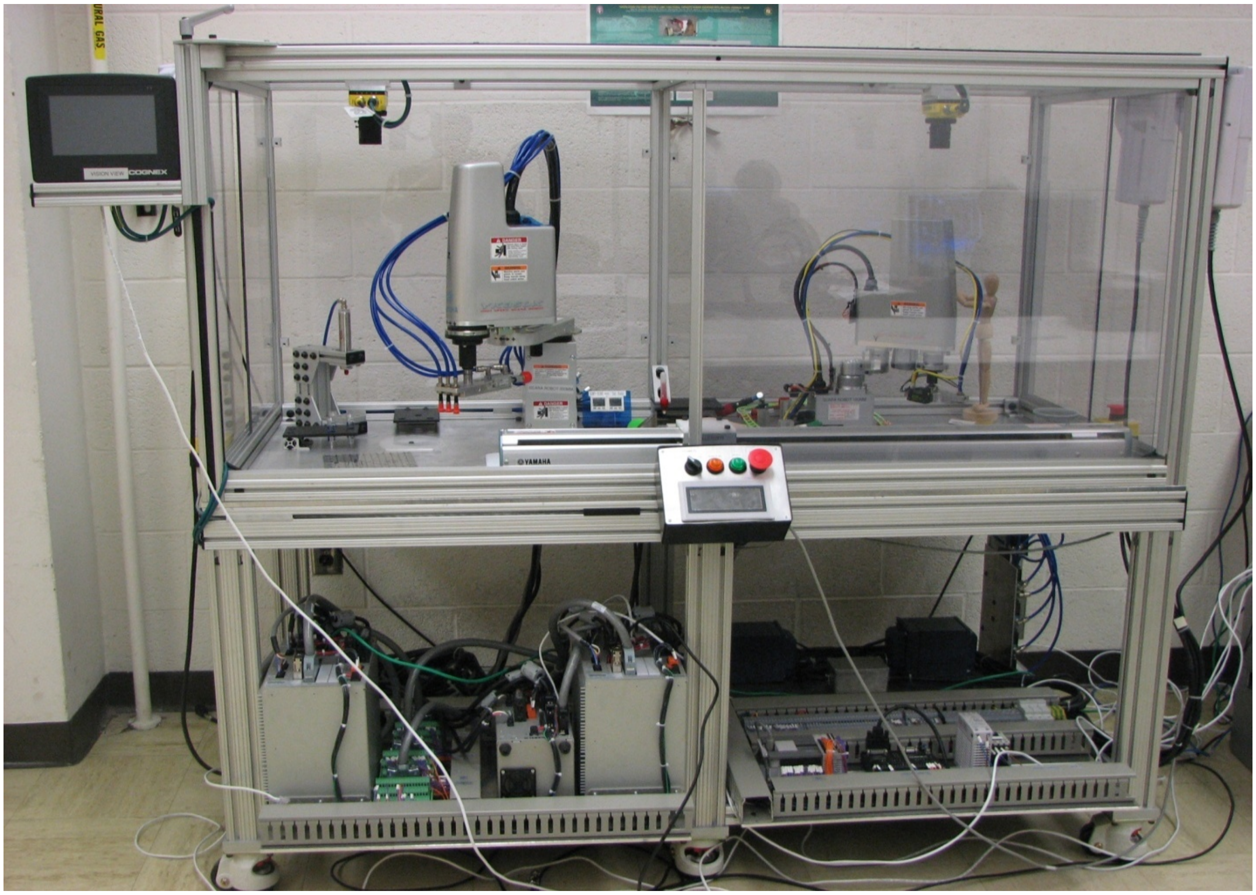


Figure 1.2: Robotic setup at ISEL lab, UTEP

1.3 Motivation of this research

The objective of this work is to apply the machine learning approach in the form of support vector machine to predict the outcome of the processes in the current domain of internet based quality control systems. Data obtained from the remote inspection experiments will be analyzed using SV classifier equations to build a model which can be used for future predictions. The motivation behind this work is to avoid the problems which might occur as the remote control of an automation process using the internet can suffer from a time lag, especially if the network is congested with heavy data traffic, which may be the greatest hurdle for using the internet for real-time control. The idea is to try and build robust classifiers which can the sort the

incoming parts based on the dimensional data into predefined groups which were used while training the algorithm.

Chapter 2: LITERATURE REVIEW

Data mining, which is also referred to as knowledge discovery in databases, means a process of nontrivial extraction of implicit, previously unknown and potentially useful information (such as knowledge rules, constraints, regularities) from data in databases [40]. In modern manufacturing environments, vast amounts of data are collected in database management systems and data warehouses from all involved areas, such as product and process design, assembly, materials planning and control, order entry and scheduling, maintenance, recycling and so on [41]. The following section illustrates the application of data mining techniques to address quality control issues.

2.1 Quality Control

Different researchers tried to solve the quality control/inspection issues using various machine learning approaches for addressing different types of problems. Automated diagnosis of sewer pipe defects was done using support vector machines (SVMs) [10] where the results showed that the diagnosis accuracy using SVMs was better than that derived by a Bayesian classifier. A combination of fuzzy logic and support vector machines was used in the form of Fuzzy support vector data description(F-SVDD) for the automatic target identification for a TFT-LCD array process [11] where the experimental results indicated that the proposed method ensemble outperformed the commonly used classifiers in terms of target defect identification rate. Independent component analysis (ICA) and SVMs were used as a combination for intelligent faults diagnosis of induction motors [12] where the results show that the SVMs achieved high performance in classification using multiclass strategy, one-against-one and one-against-all. Fault diagnosis was also done based on particle swarm optimization and support

vector machines [13] where the new method can select the best fault features in a short time and has a better real-time capacity than the method based on principal component analysis(PCA) and SVMs. Multi-class support vector machines were used for the fault diagnostics of roller bearing using kernel based neighborhood score multi-class support vector machine [14] where it was shown the multi-class SVM was effective in diagnosing the fault conditions and the results were comparable with binary SVM. Artificial neural networks were used for addressing quality control issue as a non-conventional way to detect surface faults in mechanical front seals [15] which achieved good results in comparison with the deterministic system which was already implemented. Fuzzy association rules were used to develop an intelligent quality management approach [16] with the research providing a generic methodology with knowledge discovery and the cooperative ability for monitoring the process effectively and efficiently. An automatic optical inspection was adopted for on-line measurement of small components on the eyeglasses assembly line [17] which was designed to be used at the beginning of the assembly line and is based on artificial vision, exploits two CCD cameras and an anthropomorphic robot to inspect and manipulate the objects. Fuzzy analytical hierarchy process was used to select unstable slicing machine to control wafer slicing quality [18] where the results of Exponential weighted moving average control chart demonstrated the feasibility of the proposed algorithm in effectively selecting the evaluation outcomes and evaluating the precision of the worst performing machines. Logistic Regression and PCA were the data mining algorithms used for monitoring PCB assembly quality [19] where the results demonstrated that the statistical interpretation of solder defect distributions can be enhanced by the intuitive pattern visualization for process fault identification and variation reduction. Fuzzy logic was used for the fault detection in statistical process control of industrial processes [20] and the comparative rule-based study has shown that

the developed fuzzy expert system is superior to the preceding Fuzzy rule-based algorithm. SVMs were used for an intelligent real-time vision system for surface defect detection [21] where the proposed system was found to be effective in detecting the steel surface defects based on the experimental results generated from over one thousand images. SVMs were also used as a part of the optical inspection system for the solder balls of ball grid array [22] where the system also gives the training model adjustment judgment core SVM which is efficient for the image comparison and classification. SVMs were used for quality monitoring in robotized arc welding [23] where the results show that the method can be feasible to identify the defects online in welding production. A defect classification algorithm for POSCO rolling system surface inspection system [24] was developed using Neural Networks and support vector machines with good classification ability and generalization performance. SVMs along with the wavelet feature extraction based on vector quantization and SVD techniques [25] were used for improved defect detection with the results outlining the importance of judicious selection and processing of 2D DWT wavelet coefficients for industrial pattern recognition applications as well as the generalization performance benefits obtained by involving SVM neural networks instead of other ANN models. Radial basis function (RBF) neural networks (NNS), SVMs were used for quality monitoring in a plastic injection molding process [26] where the experimental results obtained thus far indicate improved generalization with the large margin classifier as well as better performance enhancing the strength and efficacy of the chosen model for the practical case study. All the applications mentioned above are summarized in the table 2.1 given below.

Table 2.1: Support Vector Machines Applications

Applications	Approach	Researchers
Automated diagnosis	Support Vector Machines(SVMs)	Ming-Der Yang et al./2007 [10]
Automatic target defect identification	Fuzzy support vector data description (F-SVDD)	Yi-Hung Liu et al./2008 [11]
Intelligent faults diagnosis	Independent component analysis (ICA) and support vector machines (SVMs)	Achmad Widodo et al./200 [12]
Fault diagnostics	Support Vector Machine	Sheng-Fa Yuan et al./ 2006 [13]
Fault diagnostics	Multi-class support vector machine (MSVM)	V. Sugumaran et al./2008 [14]
Surface faults detection	Artificial Neural Networks	L. Barelli et al./2007 [15]
Intelligent quality management	Fuzzy association rules	H.C.W. Lau et al./2008 [16]
On-line dimensional measurement	Automatic optical inspection	G. Rosati et al./2008 [17]
Quality Control	Fuzzy analytical hierarchy process	Che-Wei Chang et al./ 2008[18]
Quality Control	Logistic Regression, principal component analysis (PCA)	Feng Zhang et al./2007 [19]
Fault Detection	Fuzzy Logic	Shendy M. El-Shal et al./2000 [20]
Surface Defect Detection	Support Vector Machine	Hongbin Jia et al./2004 [21]
Optical Inspection	Support Vector Machine	Shih-Feng Chen/2007 [22]
Quality Monitoring	Support Vector Machine	Ye Feng et al. /2002 [23]
Surface Inspection	Neural Network, Support Vector Machine	Keesug Choi et al. / 2006 [24]
Defect Detection	Support Vector Machine	D.A. Karras/2003 [25]
Quality Monitoring	Radial basis function (RBF) neural networks (NNS), support vector machines (SVMs)	Bernardete Ribeiro/ 2005 [26]

2.1.1 E-Quality Control

Although significant amount of literature is published on solving quality related issues using data mining techniques or support vector machines in particular, the concept of addressing e-quality using support vector machines remains unexplored. This is due to the fact that the whole idea of e-quality is still in its developmental stages. However, some researchers [42] developed the idea to address E-Quality for manufacturing within the framework of internet-based systems. The researchers designed the setup to perform quality control operations over the internet using Yamaha robots and machine vision cameras. The present work will be an extension to this type of work, where the data obtained from these experiments are analyzed using SVM's for future predictions. The idea behind using SVM's for this work is solely based on the fact that the performance of SVM's on binary output data is better when compared to other widely used approaches like the neural networks, principal component analysis and independent component analysis. Most of the literature review summarized above support that SVM's outperformed other methods in different cases.

2.2 Support Vector Machines

2.2.1 Historic Background

The SV algorithm is a nonlinear generalization of the Generalized Portrait algorithm developed in Russia in the sixties [27]. As such, it is firmly grounded in the framework of statistical learning theory, or VC theory, which has been developed over the last three decades by Vapnik and Chervonenki [28]. In a nutshell, VC theory characterizes properties of learning machines which enable them to generalize well to unseen data. In its present form, the SV machine was largely developed at AT&T Bell Laboratories by Vapnik and co-workers [29, 30,

31]. Due to this industrial context, SV research has up to date had a sound orientation towards real-world applications. Initial work focused on OCR (optical character recognition). Within a short period of time, SV classifiers became competitive with the best available systems for both OCR and object recognition tasks [32]. But also in regression and time series prediction applications, excellent performances were soon obtained [33]. A snapshot of the state of the art in SV learning was recently taken at the annual Neural Information Processing Systems conference [34]. SV learning has now evolved into an active area of research. Moreover, it is in the process of entering the standard methods toolbox of machine learning [35, 36, 37].

2.2.2 Support Vector Classification

The decision function for support vector classification is of the form

$$f(x) = sgn\ (\textstyle\sum_{i=1}^{m} y_i\alpha_i\langle\Phi(x),\Phi(x_i)\rangle + b)) = sgn(\sum_{i=1}^{m} y_i\alpha_i k(x,x_i) + b) \qquad (1)$$

and the following quadratic program

$$maximize_{\alpha\in\mathbb{R}^m}\ W(\alpha) = \textstyle\sum_{i=1}^{m}\alpha_i - 1/2\sum_{i,j=1}^{m}\alpha_i\alpha_j\, y_i y_j k(x_i,x_j) \qquad (2)$$

$$subject\ to\ \alpha_i \geq 0\ for\ all\ i = 1,\dots\dots,m, and\ \textstyle\sum_{i=1}^{m}\alpha_i\, y_i = o \qquad (3)$$

Soft Margin Hyperplane

In practice, a separating hyperplane may not exist, e.g., if a high noise level causes a large overlap of the classes. To allow for the possibility of examples violating, one introduces slack variables.

$$\xi_i \geq 0\ for\ all\ i = 1,\dots.,m \qquad (4)$$

in order to relax the constraints to

$$y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \;\; for\ all\ i = 1, \ldots, m \tag{5}$$

A classifier that generalizes well is then found by controlling both the classifier capacity (via $\|w\|$) and the sum of the slacks $\sum_i \xi_i$ The latter can be shown to provide an upper bound on the number of training errors.

One possible realization of such a *soft margin* classifier is obtained by minimizing the objective function

$$\tau(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \tag{6}$$

subject to the constraints (4) and (5), where the constant C > 0 determines the trade-off between margin maximization and training error minimization. Incorporating a kernel, and rewriting it in terms of Lagrange multipliers, this again leads to the problem of maximizing (2), subject to the constraints

$$0 \leq \alpha_i \leq C\ for\ all\ i = 1, \ldots, m, and\ \sum_{i=1}^{m} \alpha_i\, y_i = o \tag{7}$$

The only difference from the separable case is the upper bound C on the Lagrange multipliersα_i. This way, the influence of the individual patterns (which could be outliers) gets limited. As above, the solution takes the form (1). The threshold *b* can be computed by exploiting the fact that for all SVs x_i with α_i< C, the slack variable ξ_i is zero (this again follows from the KKT conditions), and hence

$$\sum_{j=1}^{m} \alpha_j y_j k(x_i, x_j) + b = y_i \tag{8}$$

Geometrically speaking, choosing *b* amounts to shifting the hyperplane, and [38] states that we have to shift the hyperplane such that the SVs with zero slack variables lie on the ±l lines.

2.3 Example

The following example demonstrates the usage of support vector classification methodology by using ten data points.

Suppose we have ten one dimensional data points $x_1 = 1,\ x_2 = 2,\ x_3 = 3,\ x_4 = 4,\ x_5 = 5,$ $x_6 = 6,\ x_7 = 7,\ x_8 = 8,\ \ x_9 = 9,\ x_{10} = 10$

Let us consider points 1, 2, 3, 4, 5 as class 1 and 6, 7, 8, 9, 10 as class 2.

Therefore, we can consider $y_1 = +1,\ y_2 = +1,\ y_3 = +1,\ y_4 = +1,\ y_5 = +1,\ y_6 = -1,$ $y_7 = -1,\ y_8 = -1,\ y_9 = -1,\ y_{10} = -1$ (Class 1 = +1, Class 2 = -1)

In this case, let us choose polynomial kernel of degree 2 which can be of the form

$K(x, y) = (x * y + 1)^2$ and assume C = 100

We can solve for support vectors by using the equation

$$\max \sum_{i=1}^{10} \alpha_i - 1/2 \sum_{i=1}^{10} \sum_{i=1}^{10} \alpha_i \alpha_j y_i y_j (x_i x_j + 1)^2$$

$$subject\ to\ 100 \geq \alpha_i \geq 0, \sum_{i=1}^{10} \alpha_i y_i = 0$$

By using a Quadratic Programming solver, $\alpha_1 = 0, \alpha_2 = 0, \alpha_3 = 0, \alpha_4 = 0, \alpha_5 = 0.016, \alpha_6 = 0.016, \alpha_7 = 0, \alpha_8 = 0, \alpha_9 = 0, \alpha_{10} = 0$

So, by definition the non-zero values of alpha are classified as the support vectors. Here, the support vectors are points 5 and 6 ($x_4 = 5,\ x_6 = 6$)

The discriminant function will be

$$f(y) = 0.016(+1)(5y+1)^2 + 0.016(-1)(6y+1)^2 + b$$

$$f(y) = -0.179y^2 - 0.032y + b$$

'b' can be recovered by solving f(5) = 1 or f(6) = -1, as all these points lie on

$y_i(w^T\phi(z) + b) = 1$ and both give $b = 5.63$

Therefore, the classifier equation is $f(y) = -0.179y^2 - 0.032y + 5.63$

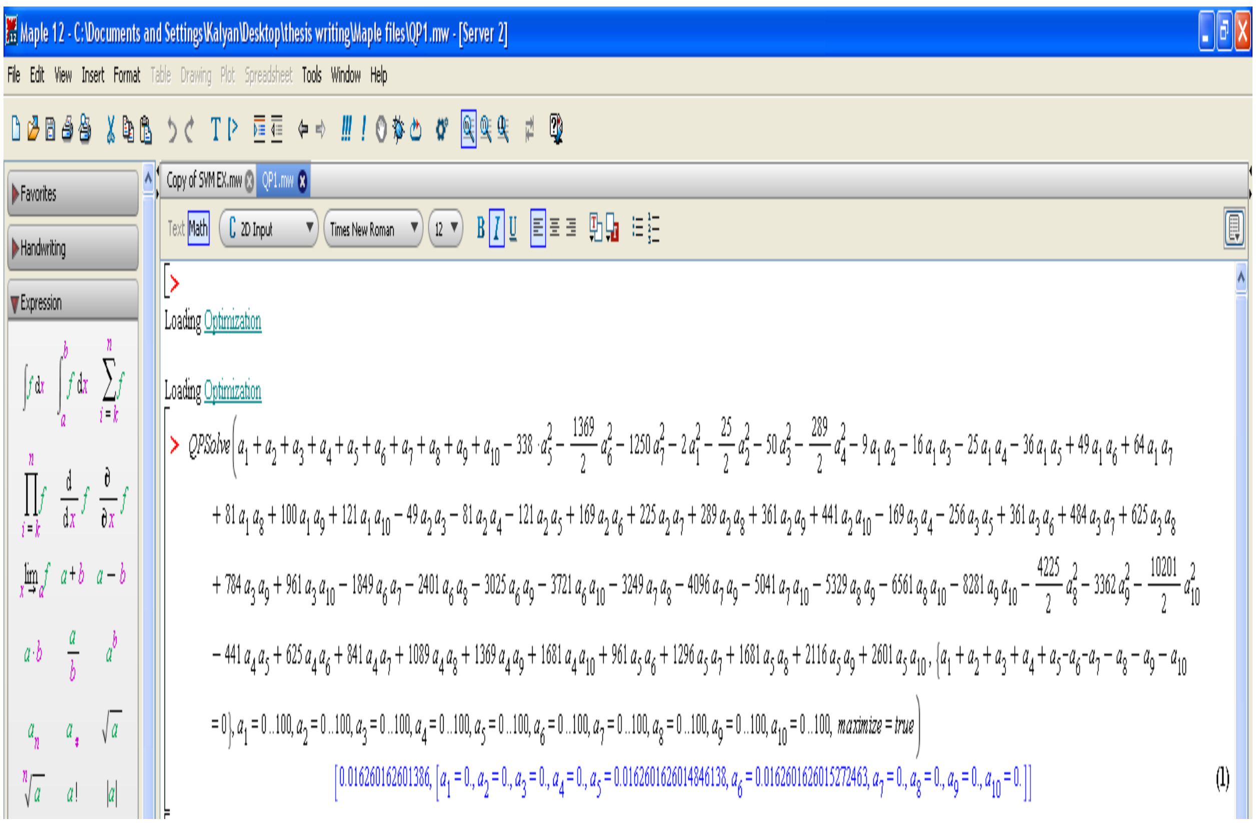


Figure 2.1: A screenshot of the Maple 12 Software used for solving the quadratic programming problem.

Chapter 3: METHODOLOGY

This chapter explains the model selection for running the experiments using the support vector classifiers, the values of training parameters selected and values of the parameters used for different kernel functions. The following figure 3.1 shows the conceptual framework used as a part of this work.

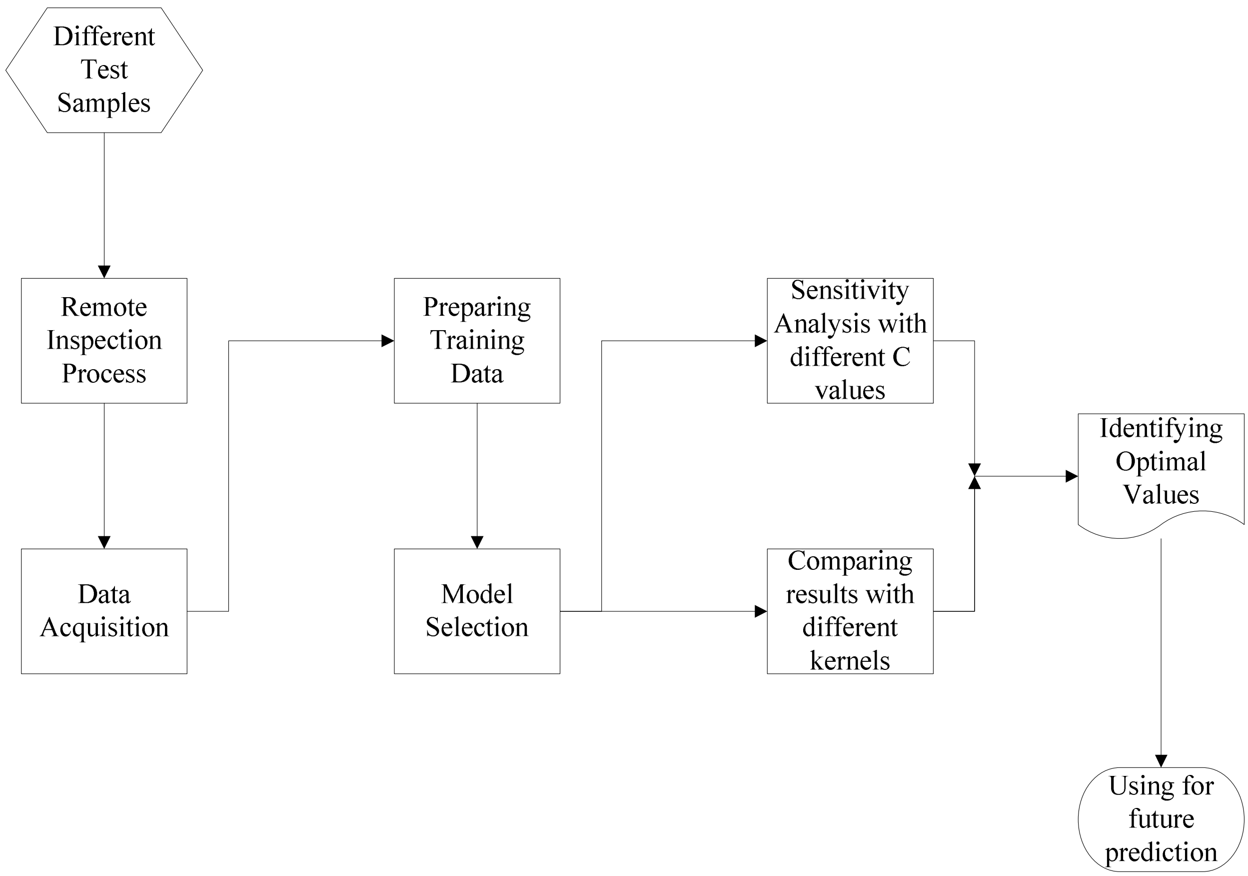


Figure 3.1: Conceptual framework

3.1 Model Selection

In training a support vector machine we need to select a kernel and set a value to the margin parameter C. Thus to develop the optimal classifier, we need to determine the optimal kernel parameter and the optimal value of C.

Estimating the training parameter C

A k-fold (value of k = 10) [39] cross validation approach is adopted for estimating the value of the training parameter. The minimum value considered was 0.1 and the maximum value was 500. Then, the value of C for which the testing accuracy percentage level was high was identified as the optimal value (highlighted in bold characters).

Performance Evaluation using different kernels

The performance of the classifiers will be evaluated by using different kernel functions in terms of testing accuracy, training accuracy, number of support vectors and validation accuracy.

Four different kernel functions are identified for this research based on the knowledge gained from the literature review. Polynomial and RBF kernels are by far the commonly used kernels in the research world.

1. Linear Kernel

2. Polynomial Kernel

3. Radial Basis Function (RBF) Kernel

4. Sigmoid Kernel

The following section identifies the different parameters involved in all the kernels which are presented above and also discusses the range for each parameter.

1. Linear Kernel

$$k(x_i, x_j) = \langle x_i , x_j \rangle \tag{9}$$

It is the inner product of $\langle x_i , x_j \rangle$, so there is no gamma, no bias.

2. Polynomial Kernel

$$k(x_i, x_j) = (\gamma x_i \, x_j \; + coefficient)^{degree} \tag{10}$$

Where ($degree \in \mathbb{N}, coefficient \geq 0, \; \gamma > 0$)

Two cases are designed based on this kernel which includes degree 2 and degree 3. Gamma as 2, coefficient as 1 are chosen based on the data. Trial and error method was adopted to find the optimal values which give high testing data classification accuracy.

The values of gamma

3. Radial Basis Function (RBF) Kernel

$$k(x_i, x_j) = exp\left(-\gamma \left|x_i - x_j\right|^2\right) \tag{11}$$

Where ($\gamma > 0$)

Two cases are also designed based on this kernel with gamma values 0.5 and 2. Trial and error method was adopted to find the optimal values which give high testing data classification accuracy.

4. Sigmoid Kernel

$$k(x_i, x_j) = tanh(\gamma x_i\, x_j \ + coefficient) \tag{12}$$

Where ($\gamma \ > 0, coefficient \ \geq 0$)

Values of gamma 0.2 and coefficient 0.1 were chosen for this work based on the data.

3.2 Training the Support Vector Classifiers

After the acquisition of the data from the experiments, the following characteristics were identified.

Total number of cases = 138

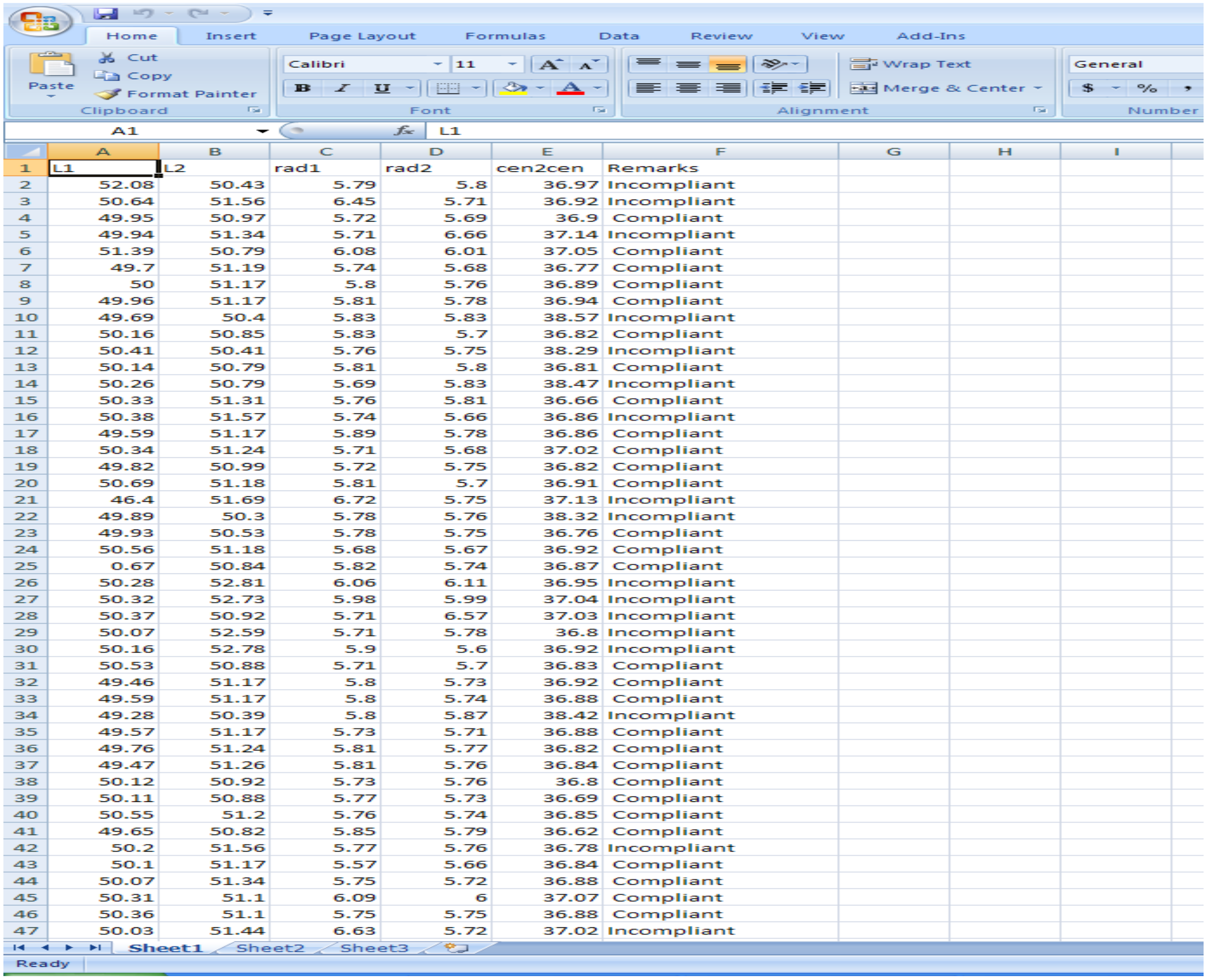
Note: Total number of times that the experiment was run including ‘auto’ and ‘manual’ modes.

Number of input features = 5

Note: Five features include the five different dimensions of the test piece.

Number of output features = 2

Note: Two outputs include the cases where the test piece is ‘compliant’ or ‘non-compliant.



	L1	L2	rad1	rad2	cen2cen	Remarks
2	52.08	50.43	5.79	5.8	36.97	Incompliant
3	50.64	51.56	6.45	5.71	36.92	Incompliant
4	49.95	50.97	5.72	5.69	36.9	Compliant
5	49.94	51.34	5.71	6.66	37.14	Incompliant
6	51.39	50.79	6.08	6.01	37.05	Compliant
7	49.7	51.19	5.74	5.68	36.77	Compliant
8	50	51.17	5.8	5.76	36.89	Compliant
9	49.96	51.17	5.81	5.78	36.94	Compliant
10	49.69	50.4	5.83	5.83	38.57	Incompliant
11	50.16	50.85	5.83	5.7	36.82	Compliant
12	50.41	50.41	5.76	5.75	38.29	Incompliant
13	50.14	50.79	5.81	5.8	36.81	Compliant
14	50.26	50.79	5.69	5.83	38.47	Incompliant
15	50.33	51.31	5.76	5.81	36.66	Compliant
16	50.38	51.57	5.74	5.66	36.86	Incompliant
17	49.59	51.17	5.89	5.78	36.86	Compliant
18	50.34	51.24	5.71	5.68	37.02	Compliant
19	49.82	50.99	5.72	5.75	36.82	Compliant
20	50.69	51.18	5.81	5.7	36.91	Compliant
21	46.4	51.69	6.72	5.75	37.13	Incompliant
22	49.89	50.3	5.78	5.76	38.32	Incompliant
23	49.93	50.53	5.78	5.75	36.76	Compliant
24	50.56	51.18	5.68	5.67	36.92	Compliant
25	0.67	50.84	5.82	5.74	36.87	Compliant
26	50.28	52.81	6.06	6.11	36.95	Incompliant
27	50.32	52.73	5.98	5.99	37.04	Incompliant
28	50.37	50.92	5.71	6.57	37.03	Incompliant
29	50.07	52.59	5.71	5.78	36.8	Incompliant
30	50.16	52.78	5.9	5.6	36.92	Incompliant
31	50.53	50.88	5.71	5.7	36.83	Compliant
32	49.46	51.17	5.8	5.73	36.92	Compliant
33	49.59	51.17	5.8	5.74	36.88	Compliant
34	49.28	50.39	5.8	5.87	38.42	Incompliant
35	49.57	51.17	5.73	5.71	36.88	Compliant
36	49.76	51.24	5.81	5.77	36.82	Compliant
37	49.47	51.26	5.81	5.76	36.84	Compliant
38	50.12	50.92	5.73	5.76	36.8	Compliant
39	50.11	50.88	5.77	5.73	36.69	Compliant
40	50.55	51.2	5.76	5.74	36.85	Compliant
41	49.65	50.82	5.85	5.79	36.62	Compliant
42	50.2	51.56	5.77	5.76	36.78	Incompliant
43	50.1	51.17	5.57	5.66	36.84	Compliant
44	50.07	51.34	5.75	5.72	36.88	Compliant
45	50.31	51.1	6.09	6	37.07	Compliant
46	50.36	51.1	5.75	5.75	36.88	Compliant
47	50.03	51.44	6.63	5.72	37.02	Incompliant

Figure 3.2: A screen shot of the excel file containing the data

Training the Algorithm:

As Support Vector Machines are part of the supervised learning methods in data mining, the data is to be divided into training set and the testing set. Going with the standard approach, two-thirds of the data is divided into training set and the remaining one thirds into testing set. Accordingly,

Training Data Sample size = 92

Testing Data Sample size = 46

According to the literature review explained in the previous chapter, for obtaining the support vectors we need to solve the equation

$$maximize_{\alpha \in \mathbb{R}^m} \;\; W(\alpha) = \sum_{i=1}^{m} \alpha_i - 1/2 \sum_{i,j=1}^{m} \alpha_i \alpha_j \, y_i y_j k(x_i, x_j) \tag{13}$$

subject to the constraints

$$0 \leq \alpha_i \leq C \; for \; all \; i = 1, \dots., m, and \; \sum_{i=1}^{m} \alpha_i \, y_i = o \tag{14}$$

Accordingly, we have to estimate the value of the training parameter C. After closely following the literature and going through the published material, it is decided that using k-fold cross validation method will be used to estimate the value of the training parameter C.

Chapter 4: RESULTS AND COMPARISONS

In this chapter, the various steps involved in the process of conducting experiments are presented and the results generated from using the classifiers in six different cases are also shown in the form of tables. Later, the comparison between the performances of different kernel functions on the given data is made and the findings are reported.

4.1 Preparing Test Samples

The experimental setup considered for this research is in the development stage and as of now it cannot be completely commercialized and used for dealing real world problems. So, a similar scenario is designed which will depict the real world cases, where instead of the parts dealt in a typical production line, small test pieces are designed to perform this research. These pieces are smaller in size, less complicated in dimensions and shapes when compared to the parts used in various types of industries. The following figure 4.1 shows the geometry of the test piece used in this experiment.

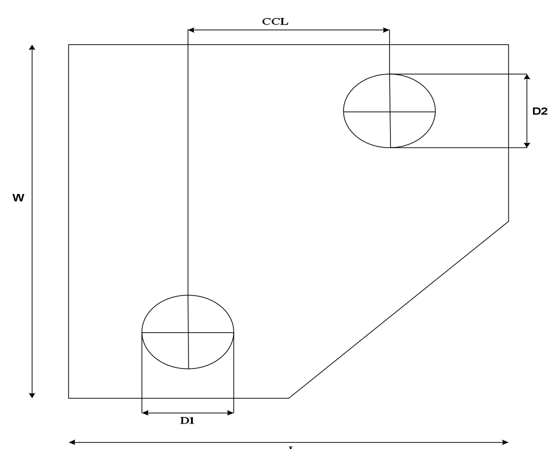


Figure 4.1: Test Piece Geometry

L – Length of the piece

W – Width of the piece

D_1, D_2 - Diameters of the two circles

CC_L – Distance between the centers of two circles

There are about twenty pieces made with same shape but each one of them differs from the original in atleast one dimension so that it depicts the real world scenario where there are potential defective products in the production line.

4.2 Performing the Inspection process

The purpose of this experiment is to become familiar with automated quality control process in a real-time industry like settings. The experiment includes recording real-time measurements on sample work pieces (products) that are passed around on a conveyor belt, compare these dimensions to the required specifications, make a decision on the quality of the product (i.e. if it is compliant or non-compliant) and take an appropriate action on the product.

Experimental Apparatus:

1. Yamaha YK 250X SCARA
2. Variable speed Dorner 6100 conveyor system
3. Cognex DVT 540 vision sensor
4. Sample test pieces

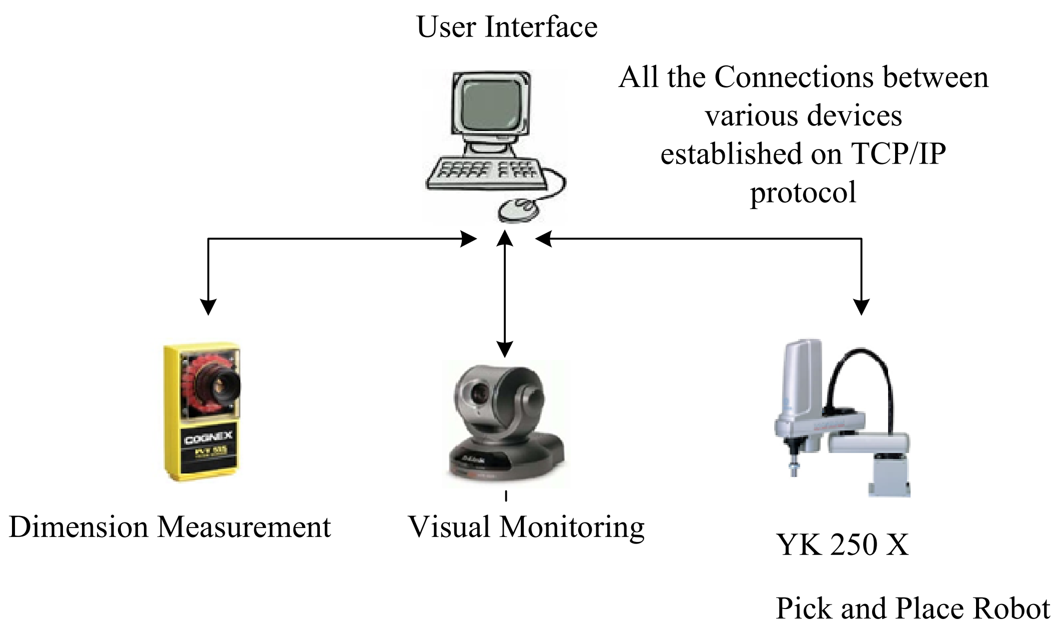


Figure 4.2: Schematic Diagram of System Setup

We use a Cognex DVT 540 vision sensor for making inspections and measurements on the object under test. We have set the camera to have an image resolution of 640 x 480 bits with an exposure time of 4□s and a frequency of 2 snapshots per second. The camera is initially trained to learn the profile of the object being tested and make the required measurements on it. Once trained, it can detect the presence of the same kind of object under different orientations. The camera can be addressed using an IP address and is capable of exchanging information with other entities over a data network. Subsequently during later inspections the camera makes measurements on the objects passing on the conveyor belt and reports it back along with the objects' position to an application server over the network through a TCP/IP connection.

The Yamaha YK 250X SCARA robot is specifically configured to be suitable for pick and place or assembly operations with a high degree of accuracy and speed, having the repeatability along horizontal planes of +/- 0.01 mm (+/- 0.0004 in.). A variable speed Dorner

6100 conveyor system is connected with the robot's I/O device ports in order to synchronize the conveyor with the motion of the robot. The robot can also be addressed using an IP address and is remotely operated over the Internet by the application server through a TCP/IP connection.

The application is software written in VB6 and runs in a PC. It communicates with the camera over the network and receives the measurement and position information about the object. It then uses this information to decide on if the product adheres to the required specifications. Once a decision is made, it communicates with the robot over the network instructs it to it stop the belt and do the proper pick and place operation on the object. The robot places the compliant and non-compliant objects on to two different stacks.

The user can start and stop the inspection process, observe and record the measurements, and override the application server to manually instruct the robot to coordinate the pick and place operations. The camera placed at the inspection site also allows for visual monitoring of the ongoing process from a remote location.

A drawing of the work-piece used for this experiment is already shown in figure 4.2. The camera is programmed to measure the key dimensions as shown in the figure. Table 4.1 summarizes the specifications for the different type of work-pieces that you will be using for your experiment. Type 1 constitutes objects that adhere to the required specifications. The other types deviate from these specifications in one dimension.

Table 4.1: Work piece Specifications (All units are in mm)

Type	L	W	D_1	D_2	CC_L	Remarks
1	50	50	12	12	37	Correct Dimensions
2	52	50	12	12	37	Wrong Length
3	50	52	12	12	37	Wrong Width
4	50	50	14	12	37	Wrong Radius
5	50	50	12	14	37	Wrong Radius
6	50	50	12	12	39	Wrong Center – Center Horizontal Distance
7	50	50	12	12	39	Wrong Center – Center Vertical Distance

4.3 Data Acquisition

Based on the previous phase, the experiment is conducted allowing the user to test the different samples. The user has two options to perform this type of experiment which include dealing with the robot in manual and auto modes. The data used for this experiment consists of data recorded in two modes.

Manual Mode: After the test piece is placed on the conveyor system, it stops as soon as it comes right below the DVT vision sensor camera which records the dimensions of the piece and through a VB interface, the user is able to see the values and has to make a judgment whether if the test piece is compliant or non-compliant. After the user takes a decision the robot is programmed to pick up the object and place it in the respective stack. The process continues until all the test samples are put on the conveyor. The significance of this mode lies on the users' ability to make the correct decision to classify the object.

Auto Mode: The process is almost similar to the manual mode except after the dimensions are recorded by the DVT vision sensor camera, the application itself takes the decision whether the piece is compliant or non-compliant.

The VB application has the option to write all the data recorded along with the action taken into an excel file. The input data used for analysis using Support Vector Machines is the product of these experiments.

Working with Statistica

Most of the analysis part is done using a statistics and analytics software package developed by StatSoft Inc. called STATISTICA which provides a selection of data analysis, data management, data mining, and data visualization procedures.

The following figure 4.3 shows the screen shot of the input data file used for the analysis in the STATISTICA 8.0 data mining module.

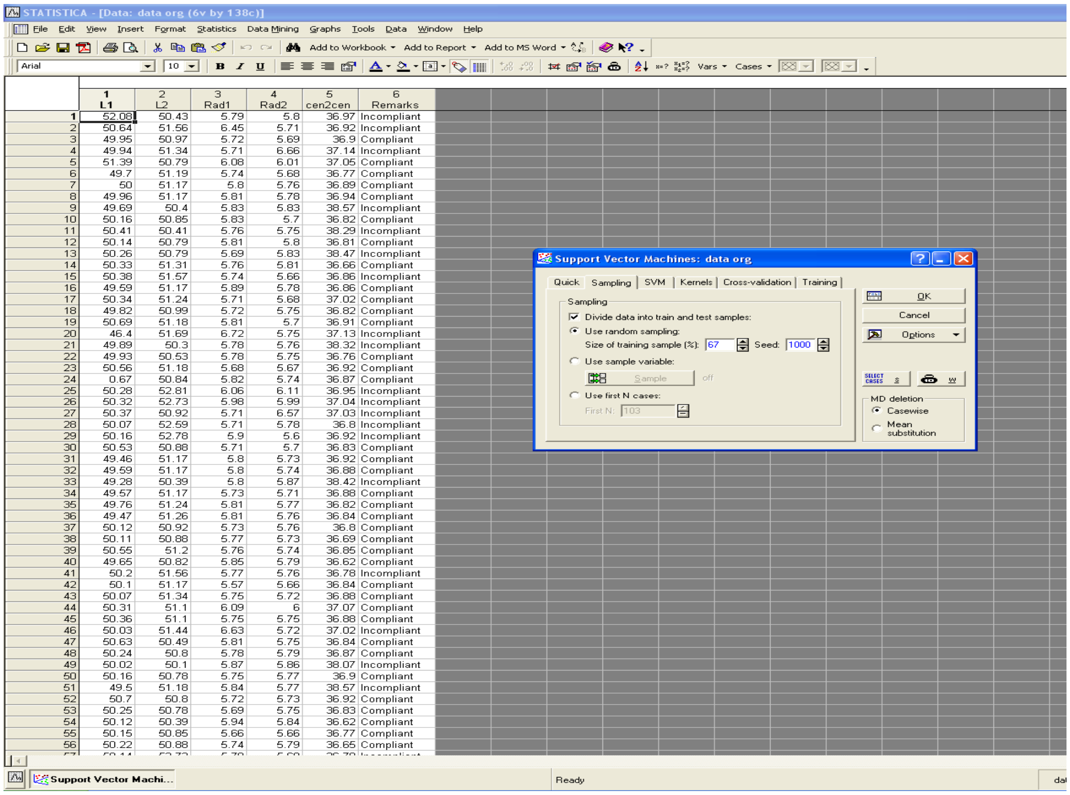


Figure 4.3: Screen shot of the STATISTICA 8.0 interface

4.4 Case Studies

Results obtained from six different case studies are presented in this section, all the cases varying in the type of kernel function used for classification.

Case 1: Radial Basis Function Kernel (gamma = 0.5)

Case 2: Radial Basis Function Kernel (gamma = 2.0)

Case 3: Polynomial Kernel (degree = 2, gamma = 2, coefficient = 1)

Case 4: Polynomial Kernel (degree = 3, gamma = 2, coefficient = 1)

Case 5: Linear Kernel

Case 6: Sigmoid Kernel (gamma = 0.2, coefficient = 0.1)

Case 1: Radial Basis Function Kernel (gamma = 0.5)

The classifier equations are tested with radial basis function kernel using a gamma value of 5 and different runs are made using different values of the training parameter. For each run, the number of support vectors generated, the accuracy of classifying the data in the training set and testing set are noted. The summary of all these observations are presented in the table 4.2.

Table 4.2: Summary table for case 1

Training Parameter ‘C’	Number of Support Vectors	Training Accuracy (%)	Testing Accuracy (%)
0.1	72	72	72
1	39	90	89
2	31	92	87
10	22	93	87
20	22	93	89
40	21	98	89
60	19	99	89
80	20	99	93
100	19	99	93
165	**16**	**99**	**93**
200	17	99	93
300	10	99	91
400	10	99	91
500	10	99	91

Values from the table are plotted in the form of a graph shown in figure 4.4 using Statistica with training parameter values on the X-axis and the accuracy percentages on the Y-axis. Optimal cross validation value is found and highlighted in the graph.

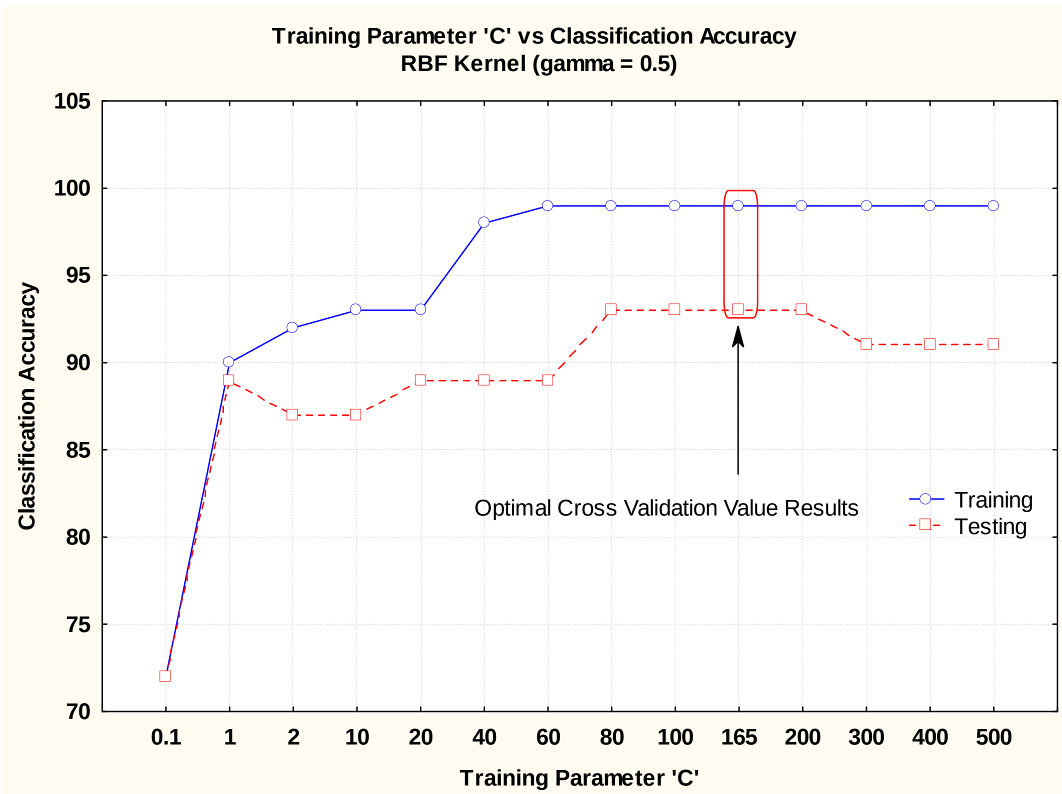


Figure 4.4: Graph showing different ‘C’ values plotted against accuracy levels

The number of support vectors identified for each case is also presented using a graph (figure 4.5) where the training parameter values are plotted on the X-axis and number of support vectors on Y-axis. A screen shot of the Statistics results box displaying the results for the best case is also presented in figure 4.6.

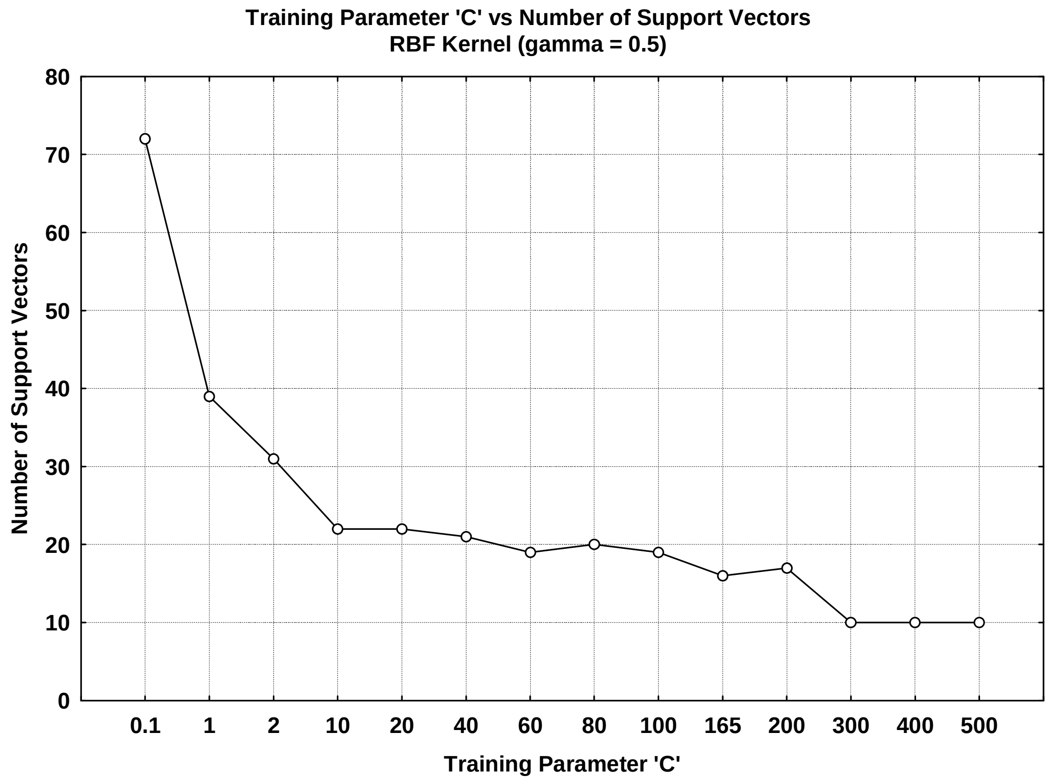


Figure 4.5: Graph showing different ‘C’ values plotted against number of support vectors.

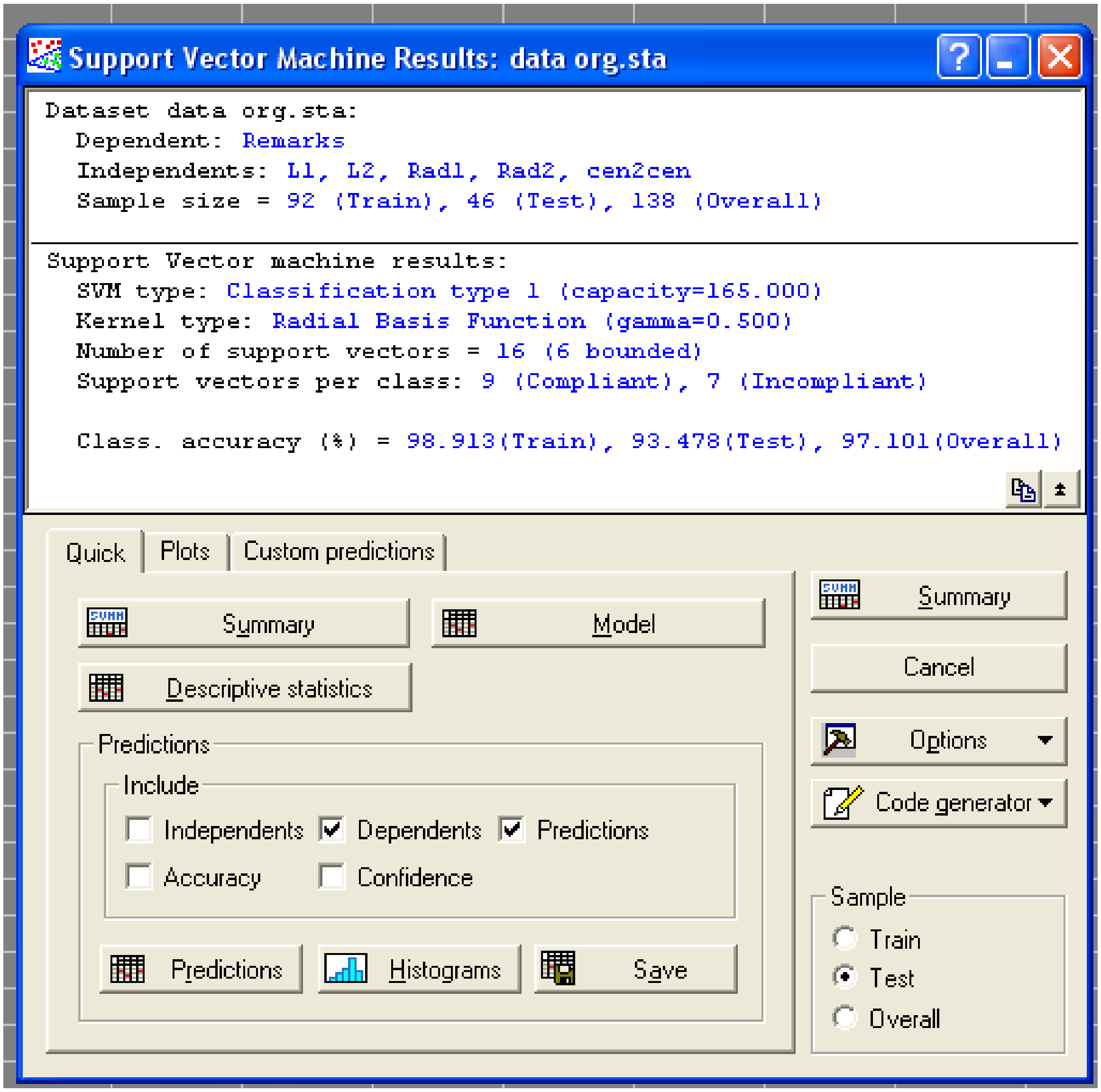


Figure 4.6: A screenshot of the Statistica results window.

Case 2: Radial Basis Function Kernel (gamma = 2.0)

The classifier equations are tested with radial basis function kernel using a gamma value of 2 and different runs are made using different values of the training parameter. For each run, the number of support vectors generated, the accuracy of classifying the data in the training set and testing set are noted. The summary of all these observations are presented in the table 4.3.

Table 4.3: Summary table for case 2

Training Parameter 'C'	Number of Support Vectors	Training Accuracy (%)	Testing Accuracy (%)
0.1	61	89	87
1	29	91	89
2	24	95	89
10	21	97	89
20	20	99	91
40	16	99	91
60	15	99	91
80	12	99	91
100	12	99	91
200	13	99	91
300	13	99	91
376	**13**	**99**	**93**
400	13	99	93
500	13	99	93

The values from the table are plotted in the form of a graph shown in figure 4.7 using Statistica with training parameter values on the X-axis and the accuracy percentages on the Y-axis. Optimal cross validation value is found and highlighted in the graph.

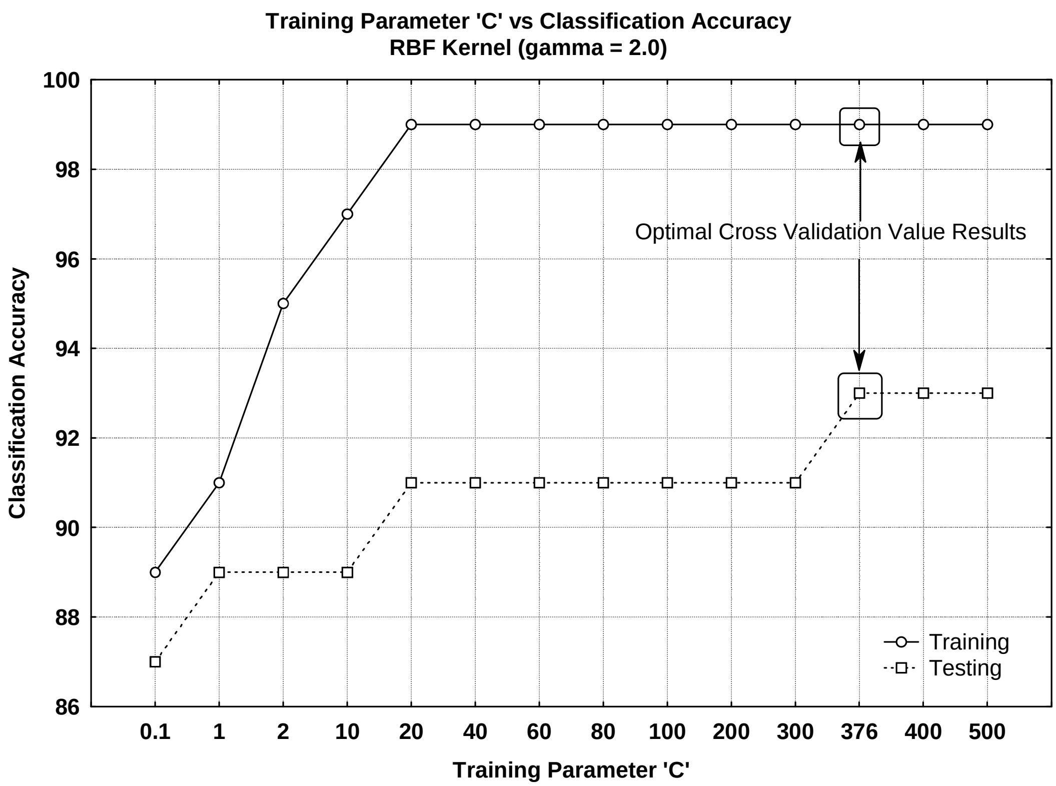


Figure 4.7: Graph showing different ‘C’ values plotted against accuracy levels

The number of support vectors identified for each case is also presented using a graph (figure 4.8) where the training parameter values are plotted on the X-axis and number of support vectors on Y-axis. A screen shot of the Statistics results box displaying the results for the best case is also presented in figure 4.9.

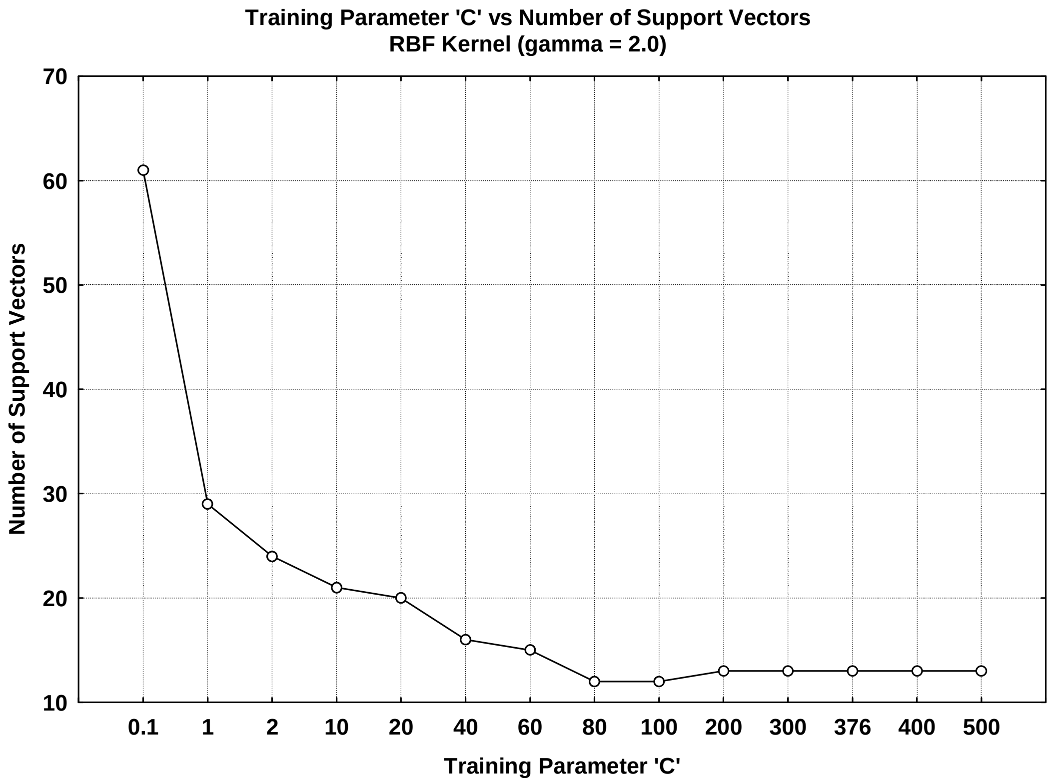


Figure 4.8: Graph showing different ‘C’ values plotted against number of support vectors.

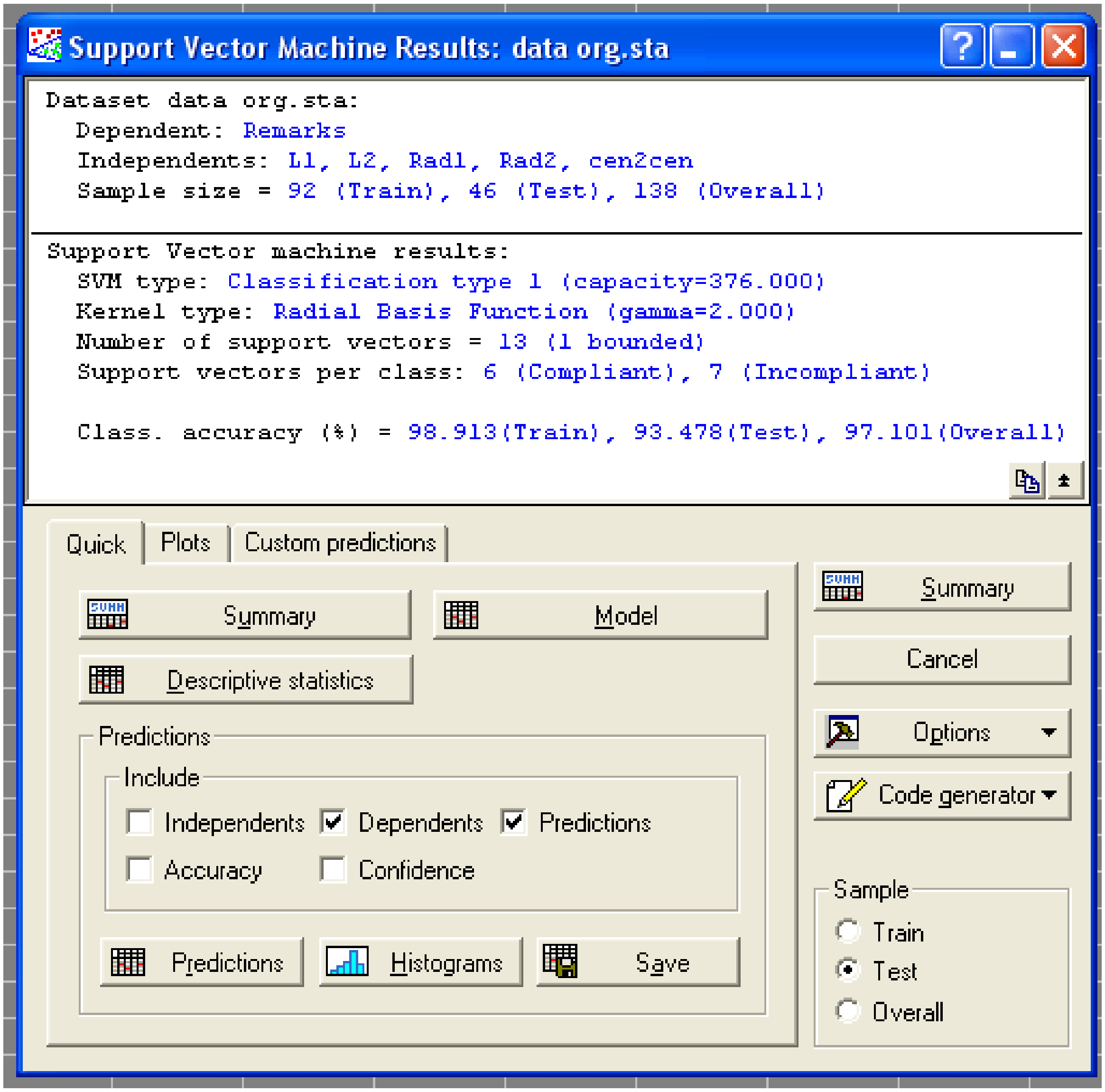


Figure 4.9: A screenshot of Statistica results window.

Case 3: Polynomial Kernel (degree = 2, gamma = 2, coefficient = 1)

The classifier equations are tested with polynomial kernel and different runs are made using different values of the training parameter. For each run, the number of support vectors generated, the accuracy of classifying the data in the training set and testing set are noted. The summary of all these observations are presented in the table 4.4.

Table 4.4: Summary table for case 3

Training Parameter 'C'	Number of Support Vectors	Training Accuracy (%)	Testing Accuracy (%)
0.1	28	93	85
1	19	93	87
2	17	97	89
6	**14**	**99**	**93**
10	14	99	93
20	13	99	91
40	11	99	91
60	12	99	91
80	12	99	91
100	11	99	91
200	10	99	91
300	11	100	89
400	9	100	91
500	11	100	89

The values from the table are plotted in the form of a graph shown in figure 4.10 using Statistica with training parameter values on the X-axis and the accuracy percentages on the Y-axis. Optimal cross validation value is found and highlighted in the graph.

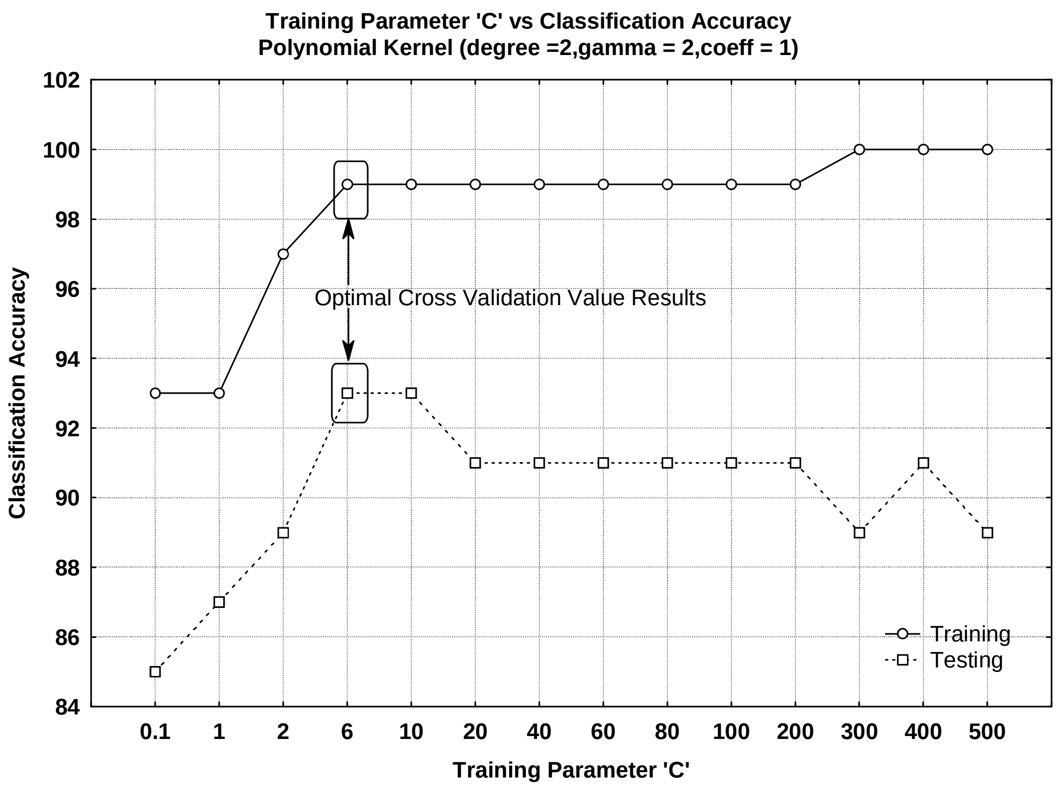


Figure 4.10: Graph showing different ‘C’ values plotted against accuracy levels

The number of support vectors identified for each case is also presented using a graph (figure 4.11) where the training parameter values are plotted on the X-axis and number of support vectors on Y-axis. A screen shot of the Statistics results box displaying the results for the best case is also presented in figure 4.12.

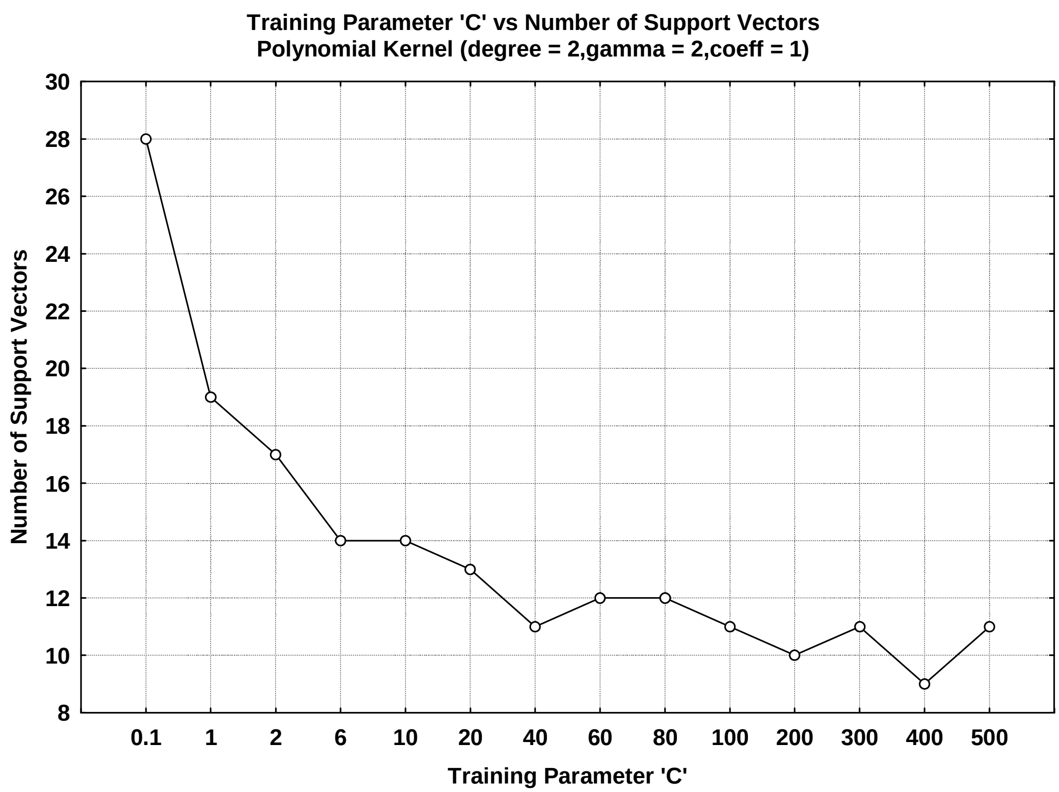


Figure 4.11: Graph showing different ‘C’ values plotted against number of support vectors.

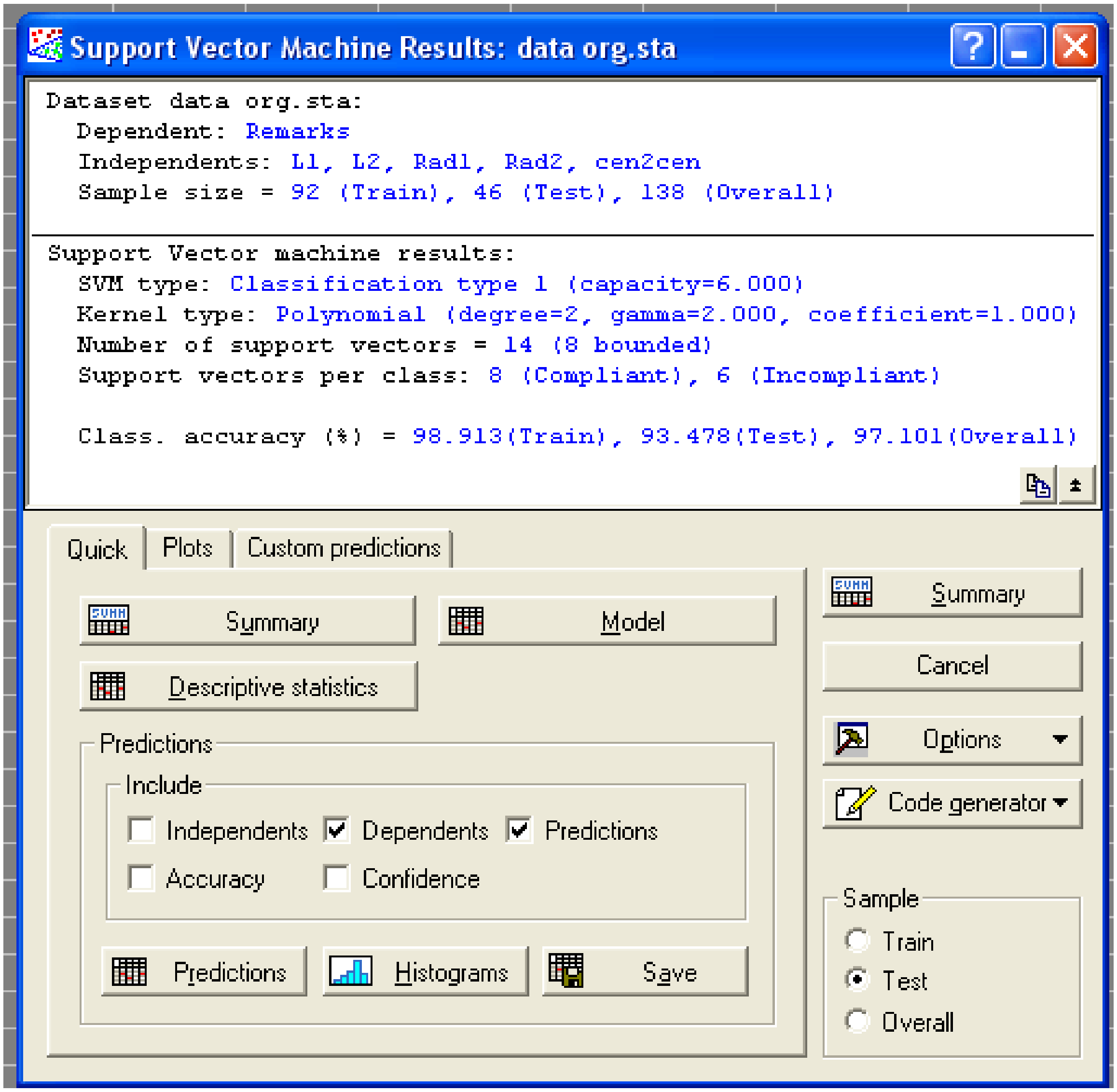


Figure 4.12: A screenshot of Statistica results window

Case 4: Polynomial Kernel (degree = 3, gamma = 2, coefficient = 1)

The classifier equations are tested with polynomial kernel and different runs are made using different values of the training parameter. For each run, the number of support vectors generated, the accuracy of classifying the data in the training set and testing set are noted. The summary of all these observations are presented in the table 4.5.

Table 4.5: Summary table for case 4

Training Parameter ‘C’	Number of Support Vectors	Training Accuracy (%)	Testing Accuracy (%)
0.1	20	93	87
1	**13**	**99**	**91**
2	12	99	91
10	8	99	91
20	9	99	91
40	8	100	89
60	8	100	85
80	8	100	85
100	8	100	85
200	8	100	85
300	8	100	85
400	8	100	85
500	8	100	85

The values from the table are plotted in the form of a graph shown in figure 4.13 using Statistica with training parameter values on the X-axis and the accuracy percentages on the Y-axis. Optimal cross validation value is found and highlighted in the graph.

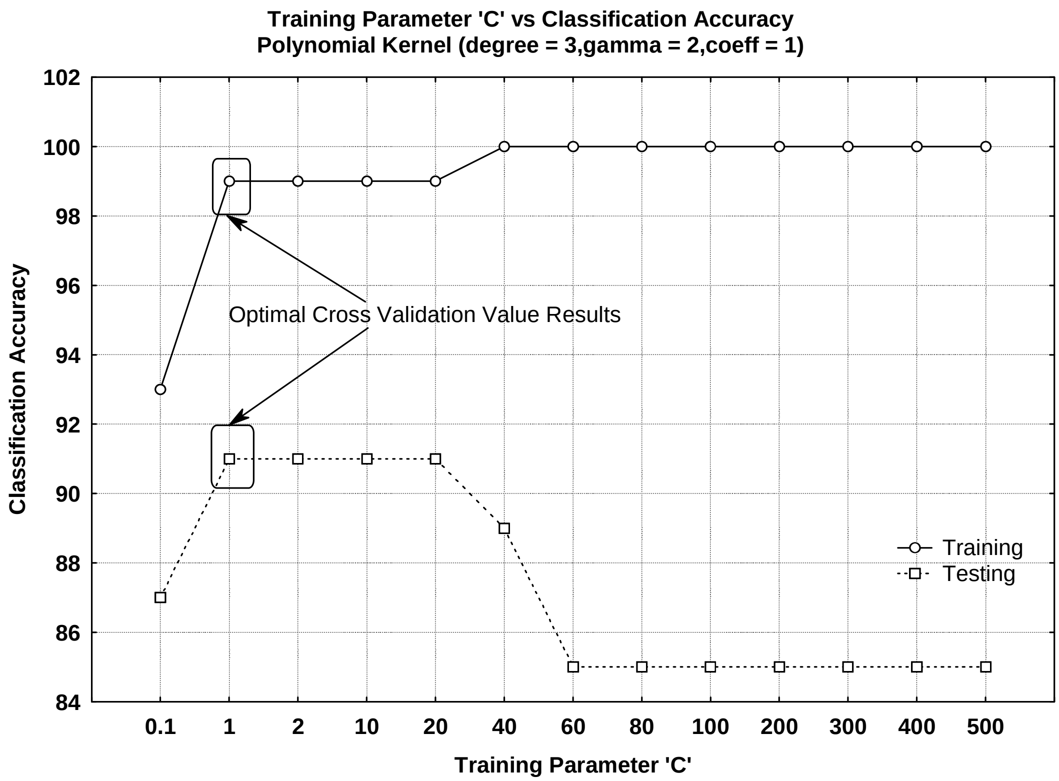


Figure 4.13: Graph showing different ‘C’ values plotted against accuracy levels

The number of support vectors identified for each case is also presented using a graph (figure 4.14) where the training parameter values are plotted on the X-axis and number of support vectors on Y-axis. A screen shot of the Statistics results box displaying the results for the best case is also presented in figure 4.15.

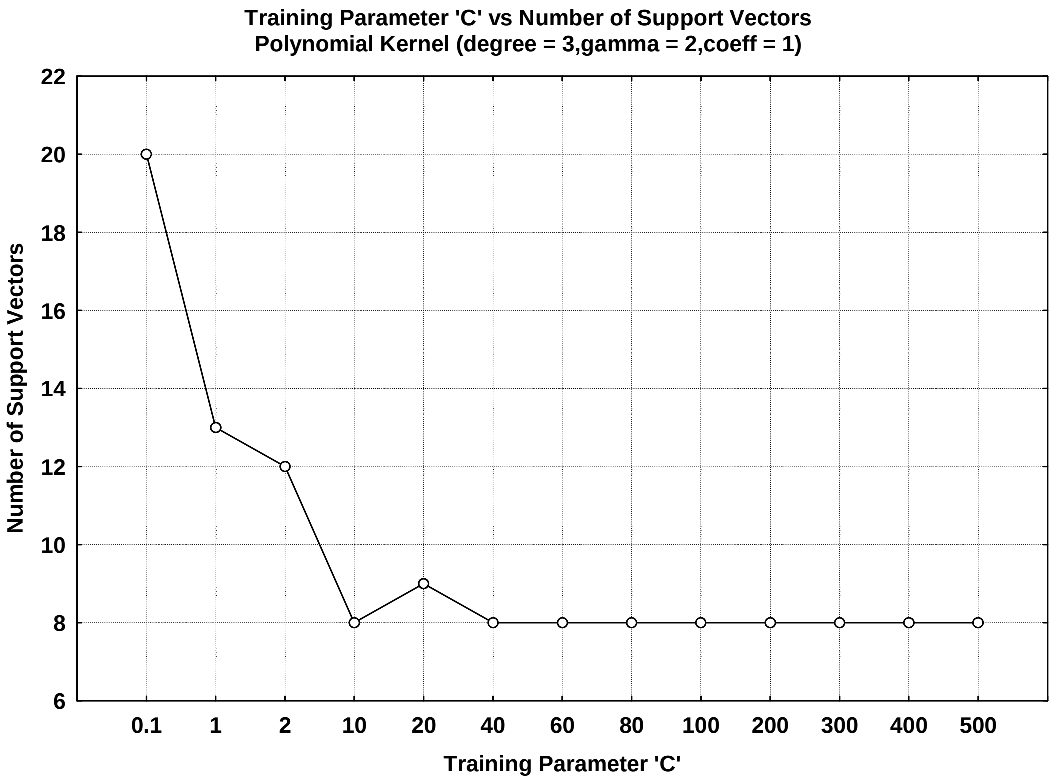


Figure 4.14: Graph showing different ‘C’ values plotted against number of support vectors.

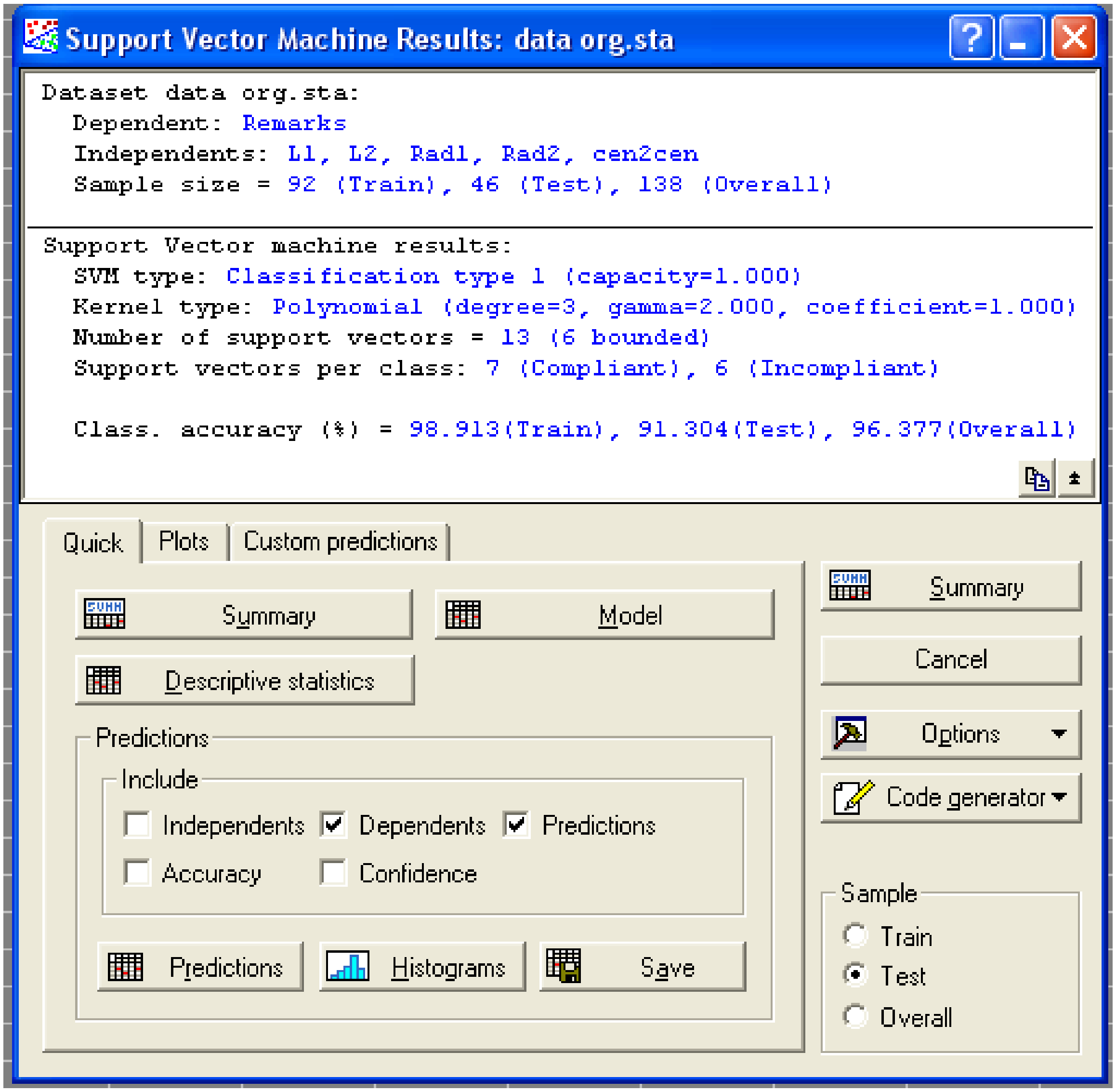


Figure 4.15: A screenshot of Statistica results window

Case 5: Linear Kernel

The classifier equations are tested with linear kernel and different runs are made using different values of the training parameter. For each run, the number of support vectors generated, the accuracy of classifying the data in the training set and testing set are noted. The summary of all these observations are presented in the table 4.6.

Table 4.6: Summary table for Case 5

Training Parameter ‘C’	Number of Support Vectors	Training Accuracy (%)	Testing Accuracy (%)
0.1	73	78	76
1	39	93	85
2	30	93	85
10	22	93	85
20	20	93	85
40	20	93	87
60	18	93	85
80	17	93	87
100	16	93	87
200	16	96	93
300	15	96	93
400	15	96	93
476	**18**	**97**	**93**
500	19	96	93

The values from the table are plotted in the form of a graph shown in figure 4.16 using Statistica with training parameter values on the X-axis and the accuracy percentages on the Y-axis. Optimal cross validation value is found and highlighted in the graph.

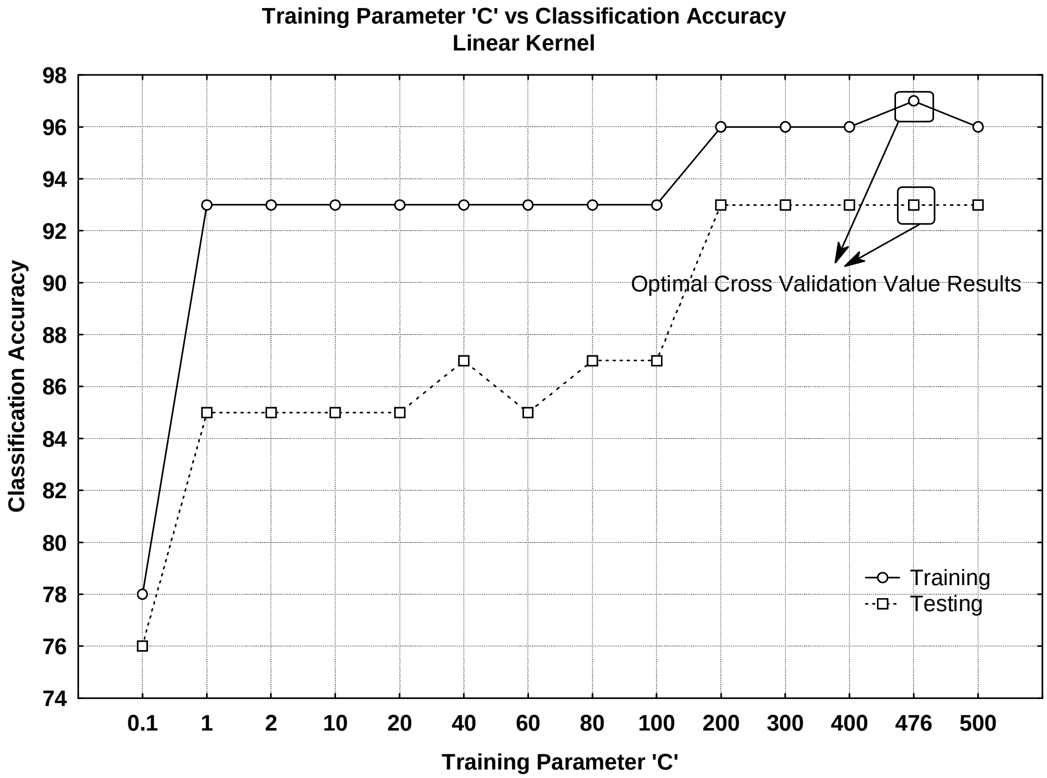


Figure 4.16: Graph showing different 'C' values plotted against accuracy levels

The number of support vectors identified for each case is also presented using a graph (figure 4.17) where the training parameter values are plotted on the X-axis and number of support vectors on Y-axis. A screen shot of the Statistics results box displaying the results for the best case is also presented in figure 4.18.

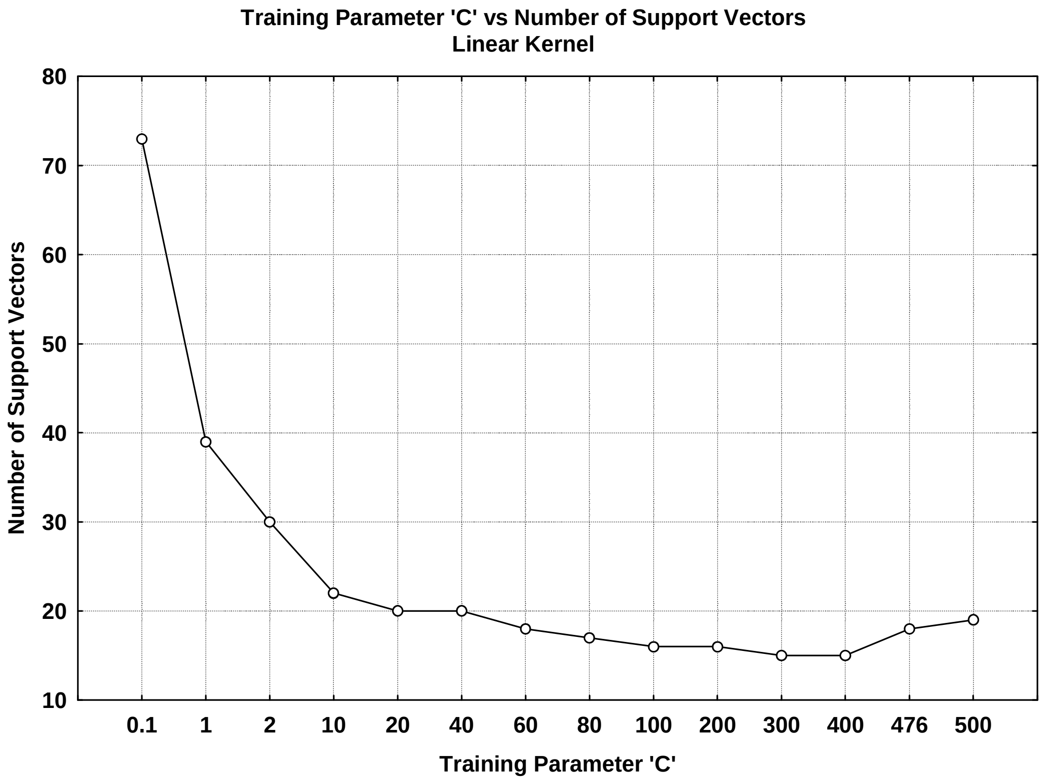


Figure 4.17: Graph showing different ‘C’ values plotted against number of support vectors.

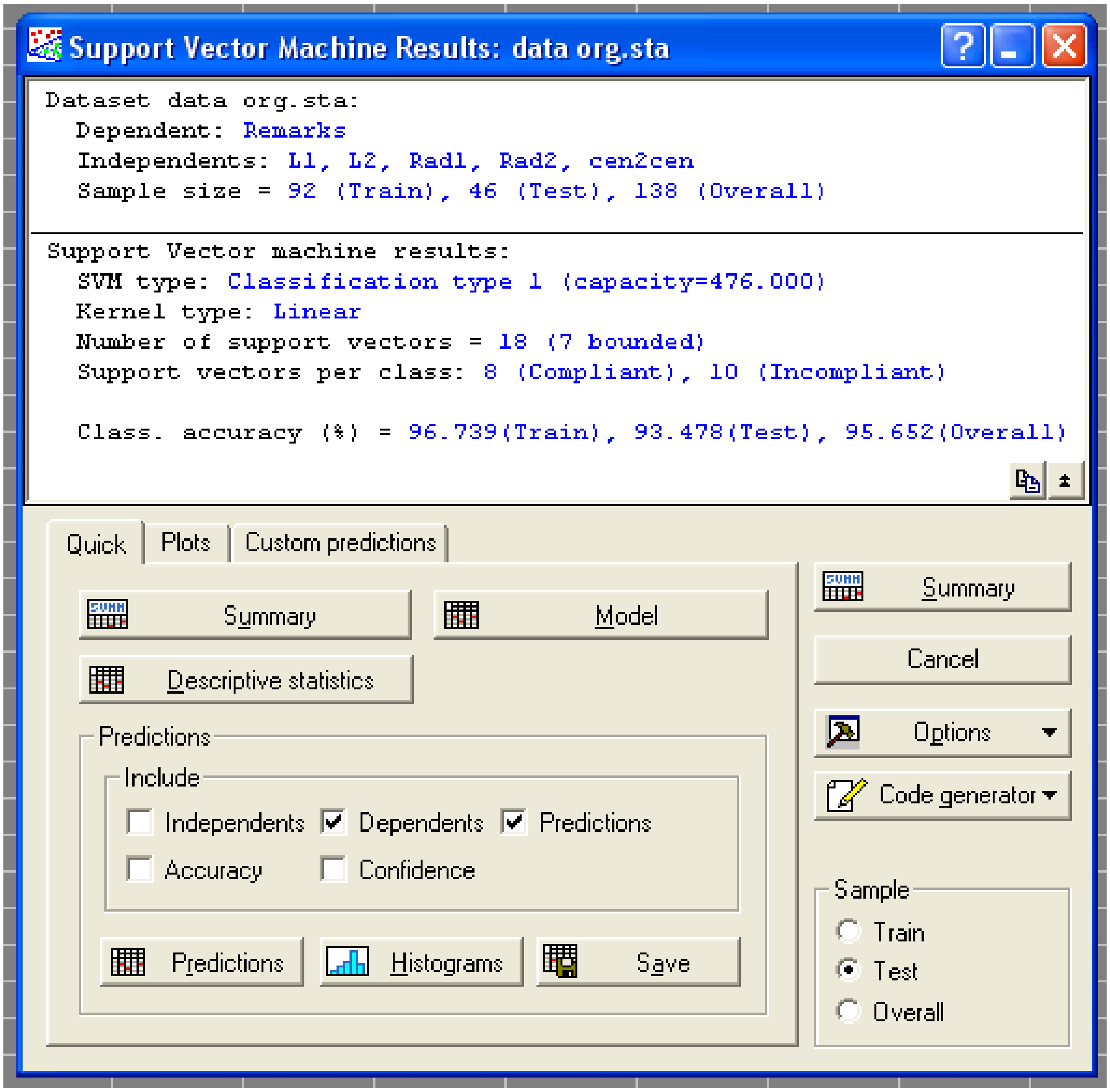


Figure 4.18: A screenshot of Statistica results window

Case 6: Sigmoid Kernel (gamma = 0.2, coefficient = 0.1)

The classifier equations are tested with sigmoid kernel and different runs are made using different values of the training parameter. For each run, the number of support vectors generated, the accuracy of classifying the data in the training set and testing set are noted. The summary of all these observations are presented in the table 4.7.

Table 4.7: Summary table for Case 6

Training Parameter ‘C’	Number of Support Vectors	Training Accuracy (%)	Testing Accuracy (%)
0.1	72	61	67
1	72	86	80
2	58	86	80
10	35	93	85
16	**30**	**93**	**85**
20	28	93	85
40	24	93	85
60	24	93	85
80	22	93	85
100	22	93	85
200	18	92	85
300	17	92	85
400	17	92	85
500	16	92	85

The values from the table are plotted in the form of a graph shown in figure 4.19 using Statistica with training parameter values on the X-axis and the accuracy percentages on the Y-axis. Optimal cross validation value is found and highlighted in the graph.

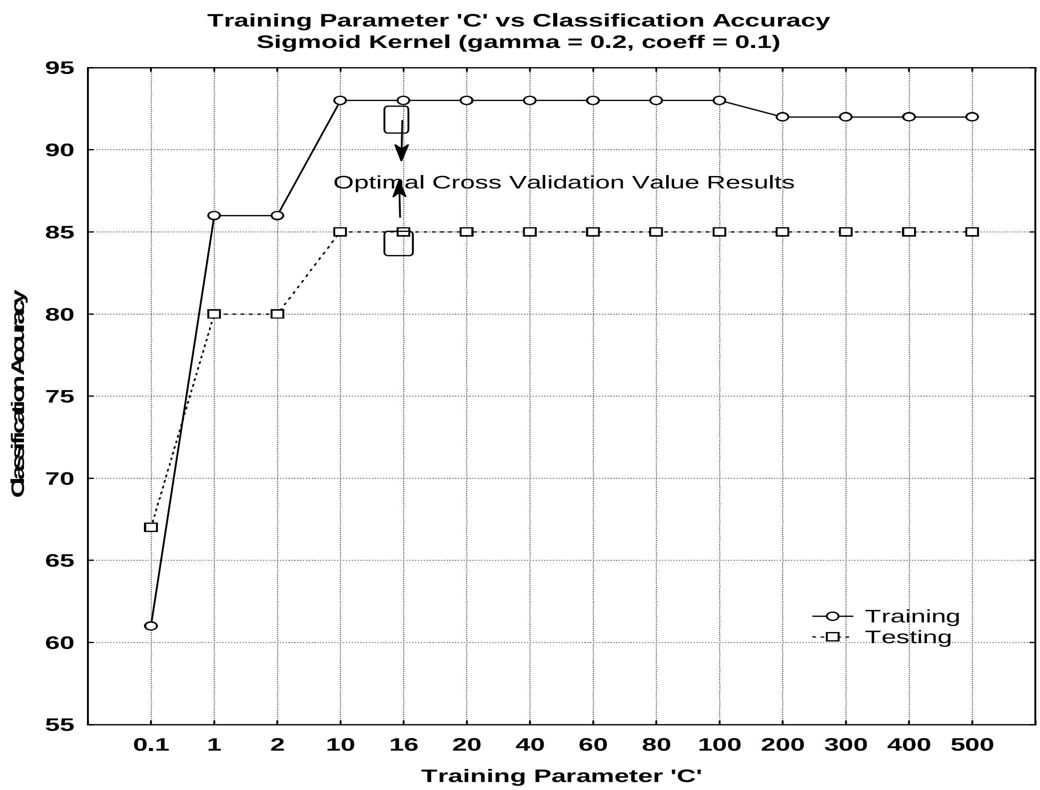


Figure 4.19: Graph showing different ‘C’ values plotted against accuracy levels

The number of support vectors identified for each case is also presented using a graph (figure 4.20) where the training parameter values are plotted on the X-axis and number of support vectors on Y-axis. A screen shot of the Statistics results box displaying the results for the best case is also presented in figure 4.21.

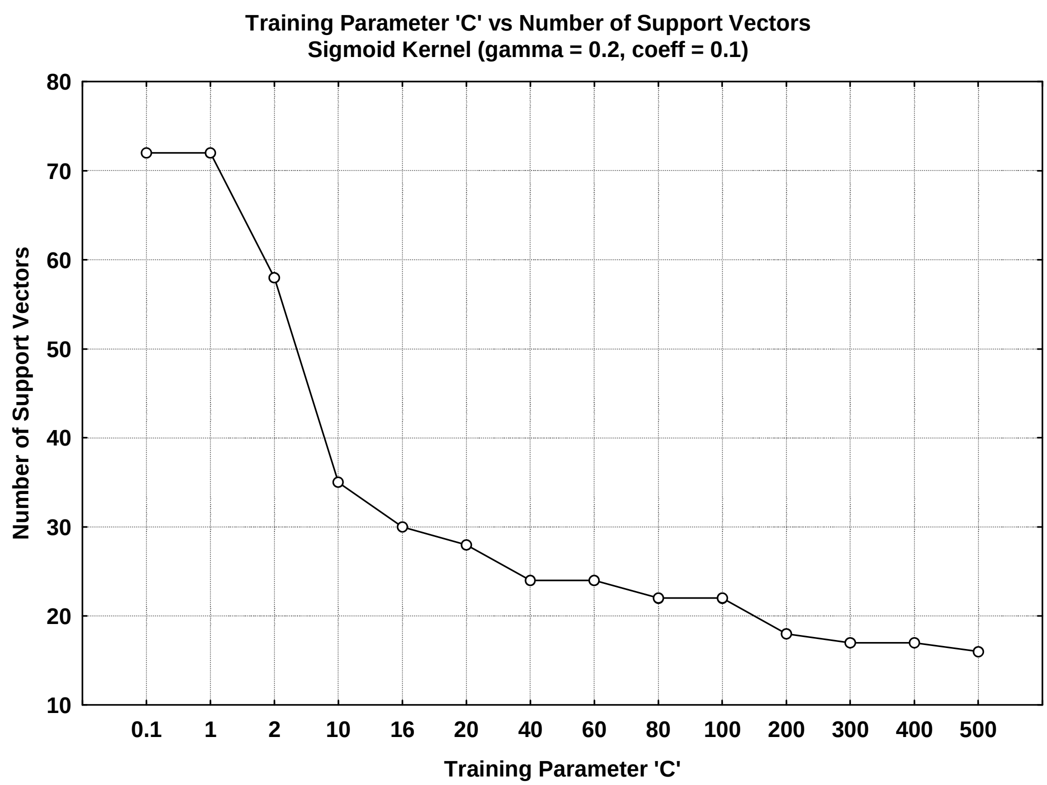


Figure 4.20: Graph showing different ‘C’ values plotted against number of support vectors.

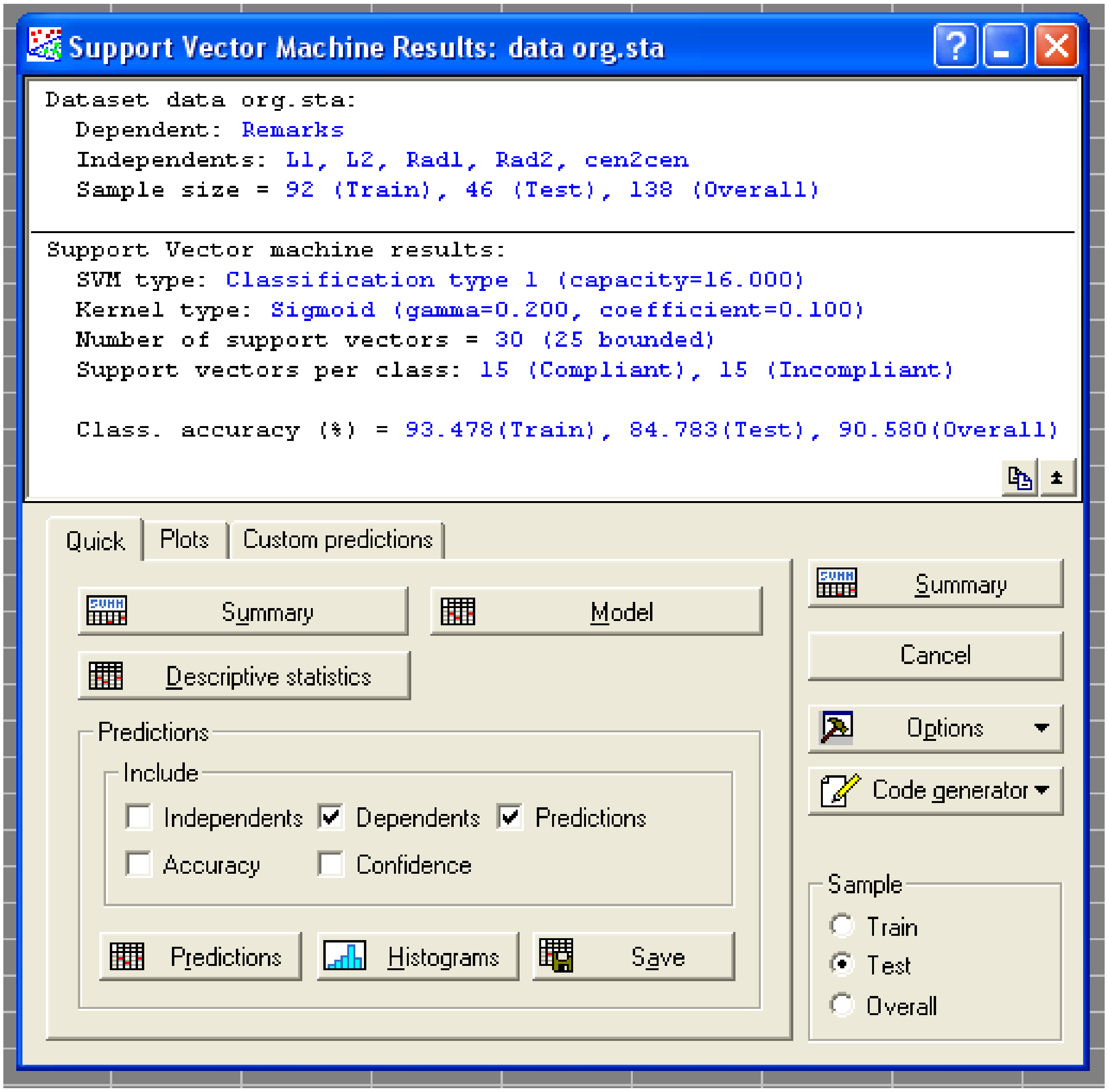


Figure 4.21: A screenshot of Statistica results window

4.5 Comparisons

This section compares all the results obtained from the different cases. After executing the model with different kernel functions, the results where the specific kernel gave the best results are identified and used for comparison along with other kernels. The following table 4.8 summarizes all the findings.

Table 4.8: Case Studies summary

Kernel	Training Parameter 'C'	Train rate (%)	Test rate (%)	SVs
Linear	476	97	93	18(7*)
Polynomial (degree =2)	6	99	93	14(8*)
Polynomial (degree =3)	1	99	91	13(6*)
RBF (gamma = 0.5)	165	99	93	16(6*)
RBF (gamma = 2)	376	99	93	13(1*)
Sigmoid (gamma = 0.2, coeff = 0.1)	16	93	85	30(25*)

* Bounded Support Vectors

Chapter 5: CONCLUSIONS

This chapter presents insights into all the results obtained from the previous chapter and highlights the important conclusions that can be drawn from the analyses. Also, the topics where further research can be focused will be presented. The purpose of this work was to develop a support vector classifier model based on the experimental data in order to facilitate the process of e-quality control by avoiding the problem faced by time lag in the process.

The study was conducted under following assumptions:

1. The data used for analysis contained 138 different cases, which were obtained by running the experiment with different test samples.

2. The model selection for training parameter C was based on v-fold cross validation approach.

3. The range of training parameter C values included 0.01 to 500, where much higher values in the order of four digits and five digits can also be used based on characteristics of the data.

4. The parameters of the kernel functions were assumed based on the trial and error basis, to obtain the best accuracy levels.

After analyzing the data obtained from the remote inspection using Support Vector Machine Classifiers and testing the accuracy levels using different kernels the following conclusions can be drawn.

1. Since the SVMs produce good classification results for data with binary outcome, the results achieved for this data were significant.

2. The highest testing rate (%) of 93 was achieved when using Linear, Polynomial and RBF kernels in different cases.

3. Although, Polynomial kernel of second degree and RBF kernel with gamma values had slightly higher training rate (%) values.

4. Amongst all the cases, RBF kernel with gamma value 2 is identified as the best performer, as it has the lowest number of support vectors used in the classification method.

5. Heuristically, less number of support vectors signifies the robustness of the classifier. However, this might not be true in all cases as it also depends on the number of bounded SVs which are located between margins.

6. Less number of bounded support vectors indicates the robustness of the classifier.

7. The value of the training parameter ‘C’ identified as 376 for RBF kernel also satisfies the basic necessity for selecting the ideal training parameter.

8. If C is too small, insufficient stress will be placed on fitting the training data and if it is too large, the algorithm leads to over fitting the data.

Future Research

Due to the above assumptions, the Support Vector model developed in this study exhibited the following aspects that need improvement to enable this work to be powerful and useful:

1. The test samples used for this work are very simple in nature, more complex pieces lead to more variations in the data.

2. The data used contained only two output classes, the same approach can also be tried to data containing multiple output classes.

3. The number of cases were also limited to 138, where analyzing more number of cases can lead to noise in the data, which then becomes a challenging aspect to deal with.

4. The maximum value of training parameter C used for analysis can be raised in the order of thousands for more detailed sensitivity analysis of the data.

5. More emphasis can be given to radial basis function kernel, where the model can be tested with a whole range of gamma values instead of limiting to just one or two.

6. The results obtained by using support vector machines can be compared to other techniques like fuzzy logic, neural networks to analyze the similarities and differences.

References

1. A. Cohen, Simulation Based Design, DARPA Plus-Up Workshop on SBD Alpha Release (DARPA/TTO Program), 1997.
2. F. Bennis, P. Castagliola, and L. Pino, “Statistical analysis of geometrical tolerances: A case study,” Journal of Quality Engineering, vol. 17, no. 3, pp. 419–427, 2005.
3. D.Goldin, S.Venneri, and A. Noor, “Newfrontiers in engineering,” Mechanical Engineering, vol. 120, no. 2, pp. 63–69, 1998.
4. D. Goldin, S. Venneri, and A. Noor, “Ready for the future?” Mechanical Engineering, vol. 121, no. 11, pp. 61–70, 1999.
5. Y. Kwon, T. Wu, and J. Ochoa, “SMWA: A CAD-based decision support system for the efficient design of welding,” Journal of Concurrent Engineering Res. Appl., vol. 12, no. 4, pp. 295–304, 2004.
6. Y. Kwon and G. Fischer, “Three-year vision plan for undergraduate instructional laboratories: Simulation-based, reconfigurable integrated lean manufacturing systems to improve learning effectiveness,” A funded proposal with $100000, Coll. Eng. Equipment Fund. Univ. of Iowa, Iowa City, May 2003.
7. P. Waurzyniak, “Moving toward the e-factory: Manufacturing industry takes first steps toward implementing collaborative e-manufacturing systems,” SME Manufacturing Engineering, vol. 127, no. 5, pp. 43–60, Nov. 2001.
8. R. Aronson, “More automation, less manpower: Smarter cells and centers,” SME Manufacturing Engineering, vol. 134, no. 6, pp. 85–108, Jun. 2005.

9. Center for Intelligent Maintenance Systems. (2005). [Online]. Available: http://wumrc.engin.umich.edu/ims/?page=home.

10. Ming-Der Yang , Tung-Ching Su, Automated diagnosis of sewer pipe defects based on machine learning approaches, Expert Systems with Applications (2007), doi:10.1016/j.eswa.2007.08.013

11. Yi-Hung Liu, Szu-Hsien Lin, Yi-Ling Hsueh, Ming-Jiu Lee, Automatic target defect identification for TFT-LCD array process inspection using kernel FCM-based fuzzy SVDD ensemble, Expert Systems with Applications (2008), doi:10.1016/j.eswa.2007.12.015

12. Achmad Widodo, Bo-Suk Yang, Tian Han, Combination of independent component analysis and support vector machines for intelligent faults diagnosis of induction motors, Expert Systems with Applications 32 (2007) 299–312

13. Sheng-Fa Yuan, Fu-Lei Chu, Fault diagnostics based on particle swarm optimization and support vector machines, Mechanical Systems and Signal Processing 21 (2007) 1787–1798

14. V. Sugumaran, G.R. Sabareesh, K.I. Ramachandran, Fault diagnostics of roller bearing using kernel based neighborhood score multi-class support vector machine, Expert Systems with Applications 34 (2008) 3090–3098

15. L. Barelli, G. Bidini, F. Mariani, M. Svanziroli, A non-conventional quality control system to detect surface faults in mechanical front seals, Engineering Applications of Artificial Intelligence (2008), doi:10.1016/j.engappai.2007.11.007

16. H.C.W. Lau, G.T.S. Ho, K.F. Chu, William Ho, C.K.M. Lee, Development of an intelligent quality management system using fuzzy association rules, Expert Systems with Applications (2008), doi:10.1016/j.eswa.2007.12.066

17. G. Rosati, G. Boschetti, A. Biondi, A. Rossi, On-line dimensional measurement of small components on the eyeglasses assembly line, Optics and Lasers in Engineering doi:10.1016/j.optlaseng.2007.11.011

18. Che-Wei Chang , Cheng-Ru Wu, Huang-Chu Chen, Using expert technology to select unstable slicing machine to control wafer slicing quality via fuzzy AHP, Expert Systems with Applications 34 (2008) 2210–2220

19. Feng Zhang,Timwah Luk, A Data Mining Algorithm for Monitoring PCB Assembly Quality, IEEE Transactions On Electronics Packaging Manufacturing, Vol. 30, No. 4, October 2007

20. Shendy M. El-Shal and Alan S. Morris, A Fuzzy Expert System for Fault Detection in Statistical Process Control of Industrial Processes, IEEE Transactions On Systems, Man, And Cybernetics-Part C: Applications And Reviews, Vol. 30, No. 2, May 2000

21. Hongbin Jia, Yi Lu Murphey, Jianjun Shi, Tzyy-Shuh Chang, An Intelligent Real-time Vision System for Surface Defect Detection, Proceedings of the 17th International Conference on Pattern Recognition (ICPR’04)

22. Shih-Feng Chen , An Optical Inspection System for the Solder balls of BGA using Support Vector Machine Classification, Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007

23. Ye Feng, Song Yong Lun, Li Di, Lai Yi Zong, Application of SVMs to quality monitoring in robotized arc welding, 0-7803-7278-6/02 IEEE

24. Keesug Choi, Kyungmo Koo, Jin S. Lee, Development of Defect Classification Algorithm for POSCO Rolling Strip Surface Inspection System, SICE-ICASE International Joint Conference 2006 Oct. 18-21, 2006 in Bexco, Busan, Korea

25. D.A. Karras , Improved Defect Detection Using Support Vector Machines and Wavelet Feature Extraction Based on Vector Quantization and SVD Techniques, 0-7803-7898-9/03/$17.00 Q2003 IEEE

26. Bernardete Ribeiro , Support Vector Machines for Quality Monitoring in a Plastic Injection Molding Process, IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications and Reviews, Vol. 35, No. 3, August 2005

27. Vapnik V. and Lerner A. 1963. Pattern recognition using generalized portrait method. Automation and Remote Control, 24: 774–780.

28. Vapnik V. and Chervonenkis A. 1974. Theory of Pattern Recognition [in Russian]. Nauka, Moscow. (German Translation: Wapnik W. & Tscherwonenkis A., Theorie der Zeichenerkennung, Akademie-Verlag, Berlin, 1979).

29. Sch¨olkopf B., Burges C., and Vapnik V. 1995. Extracting support data for a given task. In: Fayyad U.M. and Uthurusamy R. (Eds.), Proceedings, First International Conference on Knowledge Discovery & Data Mining, Menlo Park, AAAI Press.

30. Sch¨olkopf B., Burges C., andVapnikV. 1996. Incorporating invariances in support vector learning machines. In: von der Malsburg C., von Seelen W., Vorbr¨uggen J. C., and

Sendhoff B. (Eds.), Artificial Neural Networks ICANN'96, pp. 47–52, Berlin, Springer Lecture Notes in Computer Science, Vol. 1112.

31. Vapnik V., Golowich S., and Smola A. 1997. Support vector method for function approximation, regression estimation, and signal processing. In: Mozer M.C., Jordan M.I., and Petsche T. (Eds.) Advances in Neural Information Processing Systems 9, MA, MIT Press, Cambridge. pp. 281–287.
32. Sch¨olkopf B. 1997. Support Vector Learning. R. Oldenbourg Verlag, M¨unchen. Doktorarbeit, TU Berlin. Download: http://www.kernel-machines.org.
33. Mattera D. and Haykin S. 1999. Support vector machines for dynamic reconstruction of a chaotic system. In: Sch¨olkopfB., Burges C.J.C., and Smola A.J. (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, pp. 211– 242.
34. Sch¨olkopf B., Burges C.J.C., and Smola A.J. 1999a. (Eds.) Advances in Kernel Methods—Support Vector Learning. MIT Press, Cambridge, MA.
35. Haykin S. 1998. Neural Networks: A Comprehensive Foundation. 2nd edition. Macmillan, New York.
36. Cherkassky V. and Mulier F. 1998. Learning from Data. JohnWiley and Sons, New York.
37. Hearst M.A., Sch¨olkopf B., Dumais S., Osuna E., and Platt J. 1998. Trends and controversies—support vector machines. IEEE Intelligent Systems 13: 18–28.
38. V. Vapnik and A. Chervonenkis, A note on one class of perceptrons, Automation and Remote Control, 25, 1964.
39. Chang-Xue Jack Feng, Zhi-Guang (Samuel) Yu, Unnati Kingi, M. Pervaiz Baig, Threefold vs. Fivefold Cross Validation in One-Hidden-Layer and Two-Hidden-Layer

Predictive Neural Network Modeling of Machining Surface Roughness Data, Journal of Manufacturing Systems Vol. 24/No. 2, 2005.

40. Ming-Syan Chen, Data Mining: An Overview from a Database Perspective, IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, December 1996.

41. Tzu-Liang (Bill) Tseng, M.C. Jothishankar, Tong (Teresa) Wu, Quality Control Problem in Printed Circuit Board Manufacturing-An Extended Rough Set Theory Approach, Journal of Manufacturing Systems, Vol. 23/No. 1, 2004.

42. Yongjin Kwon, Shreepud Rauniar, E-Quality for Manufacturing (EQM) Within the Framework of Internet-Based Systems, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, Vol. 37, No. 6, November 2007

Appendix

Appendix A: Experimental Data

L1	L2	rad1	rad2	cen2cen	Remarks
52.08	50.43	5.79	5.8	36.97	Incompliant
50.64	51.56	6.45	5.71	36.92	Incompliant
49.95	50.97	5.72	5.69	36.9	Compliant
49.94	51.34	5.71	6.66	37.14	Incompliant
51.39	50.79	6.08	6.01	37.05	Compliant
49.7	51.19	5.74	5.68	36.77	Compliant
50	51.17	5.8	5.76	36.89	Compliant
49.96	51.17	5.81	5.78	36.94	Compliant
49.69	50.4	5.83	5.83	38.57	Incompliant
50.16	50.85	5.83	5.7	36.82	Compliant
50.41	50.41	5.76	5.75	38.29	Incompliant
50.14	50.79	5.81	5.8	36.81	Compliant
50.26	50.79	5.69	5.83	38.47	Incompliant
50.33	51.31	5.76	5.81	36.66	Compliant
50.38	51.57	5.74	5.66	36.86	Incompliant
49.59	51.17	5.89	5.78	36.86	Compliant
50.34	51.24	5.71	5.68	37.02	Compliant
49.82	50.99	5.72	5.75	36.82	Compliant
50.69	51.18	5.81	5.7	36.91	Compliant
46.4	51.69	6.72	5.75	37.13	Incompliant
49.89	50.3	5.78	5.76	38.32	Incompliant
49.93	50.53	5.78	5.75	36.76	Compliant
50.56	51.18	5.68	5.67	36.92	Compliant
0.67	50.84	5.82	5.74	36.87	Compliant
50.28	52.81	6.06	6.11	36.95	Incompliant
50.32	52.73	5.98	5.99	37.04	Incompliant
50.37	50.92	5.71	6.57	37.03	Incompliant
50.07	52.59	5.71	5.78	36.8	Incompliant
50.16	52.78	5.9	5.6	36.92	Incompliant
50.53	50.88	5.71	5.7	36.83	Compliant
49.46	51.17	5.8	5.73	36.92	Compliant
49.59	51.17	5.8	5.74	36.88	Compliant
49.28	50.39	5.8	5.87	38.42	Incompliant
49.57	51.17	5.73	5.71	36.88	Compliant

49.76	51.24	5.81	5.77	36.82	Compliant
49.47	51.26	5.81	5.76	36.84	Compliant
50.12	50.92	5.73	5.76	36.8	Compliant
50.11	50.88	5.77	5.73	36.69	Compliant
50.55	51.2	5.76	5.74	36.85	Compliant
49.65	50.82	5.85	5.79	36.62	Compliant
50.2	51.56	5.77	5.76	36.78	Incompliant
50.1	51.17	5.57	5.66	36.84	Compliant
50.07	51.34	5.75	5.72	36.88	Compliant
50.31	51.1	6.09	6	37.07	Compliant
50.36	51.1	5.75	5.75	36.88	Compliant
50.03	51.44	6.63	5.72	37.02	Incompliant
50.63	50.49	5.81	5.75	36.84	Compliant
50.24	50.8	5.78	5.79	36.87	Compliant
50.02	50.1	5.87	5.86	38.07	Incompliant
50.16	50.78	5.75	5.77	36.9	Compliant
49.5	51.18	5.84	5.77	38.57	Incompliant
50.7	50.8	5.72	5.73	36.92	Compliant
50.25	50.78	5.69	5.75	36.83	Compliant
50.12	50.39	5.94	5.84	36.62	Compliant
50.15	50.85	5.66	5.66	36.77	Compliant
50.22	50.88	5.74	5.79	36.65	Compliant
50.14	52.73	5.78	5.68	36.79	Incompliant
49.89	51.1	5.72	5.79	36.93	Compliant
50.61	52.68	6.02	6.04	36.97	Incompliant
50.12	50.79	5.75	6.66	37.04	Incompliant
49.71	50.4	5.85	5.83	38.57	Incompliant
50.39	50.68	5.75	5.81	38.28	Incompliant
49.67	51.17	5.78	6.63	37.13	Incompliant
49.85	50.8	5.83	5.83	38.47	Incompliant
49.88	50.78	5.76	5.69	38.08	Incompliant
49.15	51.04	6.52	5.73	36.98	Incompliant
50.16	50.46	5.77	5.8	38.36	Incompliant
49.85	50.65	5.78	5.87	38.42	Incompliant
50.25	51.23	5.85	6.66	36.99	Incompliant
51.45	50.82	5.79	5.73	36.98	Compliant
50.34	50.8	6.12	5.99	36.85	Compliant
50.22	51.17	5.82	5.82	36.8	Compliant
48.99	51.49	5.75	5.68	36.83	Compliant
50.2	51.31	5.82	5.72	36.83	Compliant

50.03	51.17	5.86	5.74	36.88	Compliant
50.23	51.17	5.86	5.79	36.84	Compliant
50.27	50.79	5.84	5.77	36.86	Compliant
50.06	50.41	5.94	5.86	36.84	Compliant
49.72	51.22	5.81	5.76	36.79	Compliant
49.62	51.17	5.83	5.84	36.64	Compliant
50.41	50.43	5.82	5.74	36.76	Compliant
50.14	50.79	5.84	5.75	36.7	Compliant
50.46	51.04	5.7	5.75	36.79	Compliant
50.26	50.9	5.82	5.68	36.85	Compliant
26.98	29.52	5.68	5.73	36.74	Compliant
51.07	50.78	6.08	6.07	37.06	Compliant
49.85	51.31	5.8	5.76	36.92	Compliant
49.92	50.92	6.59	5.75	36.97	Incompliant
50.31	51.18	5.66	5.69	36.83	Compliant
50.16	50.78	5.74	5.64	36.81	Compliant
50.5	50.48	5.9	5.84	36.88	Compliant
49.84	51.1	5.74	5.73	36.86	Compliant
50.68	51.17	5.73	5.49	36.63	Compliant
52.97	51.35	5.77	5.77	36.96	Incompliant
50.04	50.82	5.72	5.66	36.78	Compliant
49.82	51.34	5.68	5.66	36.85	Compliant
49.97	51.24	5.72	5.77	36.96	Compliant
50.19	51.07	5.8	5.76	36.77	Compliant
51.15	53.25	5.96	6	37.07	Incompliant
50.36	50.78	5.71	5.7	36.92	Compliant
50.42	50.81	5.74	5.74	36.88	Compliant
50.97	51	5.75	5.71	36.95	Compliant
50.32	50.83	5.79	5.59	36.91	Compliant
50.82	50.47	5.82	5.74	36.87	Compliant
50.35	50.94	5.73	5.78	36.91	Compliant
49.95	50.94	5.74	6.51	37.24	Incompliant
50.18	50.9	5.77	5.7	38.31	Incompliant
50	50.42	5.85	5.86	38.54	Incompliant
50.28	50.68	5.79	6.6	37.05	Incompliant
50.17	50.79	5.82	5.76	38.5	Incompliant
49.88	50.04	5.78	5.69	38.24	Incompliant
51.09	51.04	6.65	5.68	37.2	Incompliant
50.21	52.35	6.02	5.99	37.02	Incompliant
50.17	51.26	5.75	5.85	36.86	Compliant

50.01	52.28	5.73	5.76	37.04	Incompliant
50.01	50.99	5.82	5.7	36.58	Compliant
49.79	51.07	5.78	5.7	36.88	Compliant
49.71	50.79	6	5.87	36.89	Compliant
50.7	51.05	5.79	5.69	36.92	Compliant
49.82	50.79	5.76	5.73	36.83	Compliant
46.69	51.34	5.73	5.74	36.92	Compliant
50.76	50.79	5.85	5.73	36.72	Compliant
50.31	50.6	5.53	5.73	37.08	Compliant
49.4	50.84	5.7	5.68	36.77	Compliant
51.37	50.82	6.13	5.94	36.95	Compliant
50.09	50.78	5.77	5.71	36.87	Compliant
50.19	50.92	5.76	5.74	36.92	Compliant
49.76	50.79	5.75	5.77	36.82	Compliant
50.24	51.04	5.66	5.7	36.8	Compliant
50.38	50.88	5.72	5.7	36.73	Compliant
49	51.1	5.72	6.62	37.13	Incompliant
50	50.36	5.79	5.75	38.29	Incompliant
50.35	50.49	5.88	5.8	38.5	Incompliant
50.01	51.34	5.73	6.55	37.18	Incompliant
49.58	50.79	5.91	5.85	38.52	Incompliant
49.82	50.4	5.81	5.86	38.25	Incompliant
51.36	51.28	6.55	5.66	37.05	Incompliant
50.11	52.42	6.06	6.11	36.87	Incompliant

Curriculum Vita

Kalyan Reddy Aleti was born in Hyderabad, India on March 19, 1985. The only son of Satyanarayana Reddy Aleti and Sunitha Aleti, he graduated from Birla Institute of Technology and Science, Pilani, India with a Bachelor of Engineering degree in Mechanical Engineering in the spring of 2006. He entered the University of Texas at El Paso in fall 2006 to pursue his Master of Science degree in Industrial Engineering. While at the university, he worked as a research assistant at the intelligent systems engineering lab in the Industrial Engineering department. He was also a member of Alpha Pi Mu El Paso chapter during his stay at University of Texas at El Paso.

Permanent Address: 808 Heisig Ave, Apt B

El Paso, Texas, 79902

This thesis was typed by Kalyan Aleti