Support Vector Machines and other Pattern Recognition Approaches to the
Diagnosis of Cerebral Palsy Gait

J Kamruzzaman¹ and R. Begg²

¹Monash University, Churchill Campus, Vic-3842, Australia
²Victoria University, Melbourne, PO Box 14426, Vic-8001, Australia

Corresponding author:

Dr. Rezaul Begg
Biomechanics Unit
Centre for Ageing, Rehabilitation, Exercise & Sport Science
City Flinders Campus
Victoria University, PO Box 14428, Melbourne City MC
Victoria 8001, Australia.
Tel: +61 3 9919 1116, Fax: +61 3 9919 1110
Abstract

Accurate identification of Cerebral Palsy (CP) gait is important for diagnosis as well as for proper evaluation of the treatment outcomes. This paper explores the use of support vector machines (SVM) for automated detection and classification of children with CP using two basic temporal-spatial gait parameters (stride length and cadence) as input features. Application of the SVM method to a children’s dataset (68 normal healthy and 88 with spastic diplegia form of CP) and testing on ten-fold cross-validation scheme demonstrated that an SVM classifier was able to classify the children groups with an overall accuracy of 83.33% (sensitivity 82.95%, specificity 83.82%, area under the receiver operating curve (AUC-ROC=0.88)). Classification accuracy improved significantly when the gait parameters were normalized by the individual leg length and age, leading to an overall accuracy of 96.80% (sensitivity 94.32%, specificity 100%, AUC-ROC area=0.9924). This accuracy result was respectively 3.21% and 1.93% higher when compared to an LDA (Linear Discriminant Analysis) and an MLP (Multi-Layer Perceptron) based classifier. SVM classifier also attains considerably higher ROC area than the other two classifiers. Among the four SVM kernel functions (linear, polynomial, radial basis and ANOVA spline) studied, the polynomial and radial basis kernel performed comparably and outperformed the others. Classifier’s performance as functions of regularization and kernel parameters was also investigated. The enhanced classification accuracy of the SVM using only two easily obtainable basic gait parameters makes it attractive for identifying CP children as well as for evaluating the effectiveness of various treatment methods and rehabilitation techniques.

Key words: Gait, cerebral palsy, support vector machines, neural networks, classification
1. Introduction

Gait analysis typically includes measurements of temporal-spatial parameters of the gait cycle, motion of joints and segments, force/moments, and electromyography patterns of muscle activation [1]. Gait analysis is routinely used in clinical settings for the assessment of walking performance. One such important application of gait analysis is in the assessment of cerebral palsy (CP) patients. For example, in a recent study involving the treatment of 102 patients with CP, gait analysis was found to play a significant role in the decision making and planning for operation outcomes [2].

Qualitative ways of classifying the abilities of CP children in their gait performance has limited significance. Therefore, there is a pressing need for devising ways to quantify the gait performance so that their overall performance can be quantified and monitored as a result of treatments. Automated recognition and classification of the CP gait characteristics could offer many potential benefits, especially for the diagnosis, assessment and evaluation of the treatment outcomes. Supervised classifiers (e.g., neural networks, support vector machines, neurofuzzy methods, etc) usually provide output values that are either in the range ‘0 to 1’ (0 being normal and 1 pathological) or a quantity that represents a membership value. Thus, the classifier output could be used to gauge the movement of the ‘overall performance’ during an intervention within the spectrum of normal and pathological conditions. For example, during the course of a certain treatment /rehabilitation intervention, if the classifier’s output shift towards zero, then it would indicate that the intervention has improved the overall gait function of the individual by being closer towards the normal healthy group. Using neural networks, Hahn et al. [3] have shown that such outputs could be used to quantify the extent of abnormality in the balance impairments in older adults. O’Malley et al. [4], using the fuzzy clustering technique, have demonstrated that a membership function, derived from basic gait features, can be used to differentiate between normal and CP patients. Using gait data from 88 spastic diplegia CP children and from 68 normal children, their technique assigned each child’s gait data a membership function related to one of the five different clusters. Improvement as a result of treatment was shown by a change in the membership function of the
subject. Like these supervised classifiers, support vector machines can also used to detect the extent of abnormality and assess the degree of improvement in patient’s gait function.

Machine learning approaches have been used on numerous occasions as automated classifiers for gait and biomedicine applications, including use in: classification of normal and pathological gait functions with the aid of neural networks and fuzzy logic classifiers [4]-[8]; identification of the category and intensity of free-living human locomotion (e.g., level walking, running, ascending and descending stairs) using neural networks combined with data from an insole measurement system [9]; detection of gait events in CP cases from the electrical activity of muscles using adaptive neuro-fuzzy inference systems (ANFIS) [10]; classification and discriminant analysis of biomechanical functional foot data [11]; classification of glaucoma disease using support vector machines and other classifiers [12]; channel selection processes and classification of electroencephalogram (EEG) signals for designing a brain computer interface system using support vector machines [13]. Furthermore, neural network approaches have been viewed as superior classification techniques when compared to statistical analysis (e.g., linear discriminant analysis) in their applications within ankle arthrodesis and normal subjects [7].

Support vector machines (SVMs) are a relatively new machine learning tool and has emerged as a powerful technique for learning from data and solving classification and regression problems [14]. This has been particularly effective for binary classification applications. Applications of the SVM in gait pattern detection and classification has been so far limited to the classification of young and elderly gait patterns [15] despite their demonstrated high success rates in other biomedical areas (e.g., [16]). In this paper, we explore the classification ability of SVM in normal and CP children using gait data provided by O’Malley et al. in [4] and the classification outcomes are compared with Multi-Layer Perceptron (MLP) and Linear Discriminant Analysis (LDA) classifiers.

The remainder of the paper is organized as follows. Section II describes the CP and normal gait data acquisition and normalization techniques and Section III presents different classifiers studied in this work.
and the metrics used for comparison. In Section IV, we present experimental results followed by the discussion in Section V.

II. Gait Data Acquisition and Normalization Techniques

The gait features used in this study for training and testing the SVM classifier were taken from the published CP and normal children’s gait data by O’Malley et al. [4]. The detailed description of data acquisition can be found in that paper. In the following, we present a brief overview of the gait features used in this study.

A. Gait Measurement and Feature Selection

The data set includes 88 children with spastic diplegia form of cerebral palsy (age between 2 to 20 years, mean=9.9 years) and 68 children of neurologically intact control group (age between 2 to 13 years, mean=7.1 years) with no history of motor pathology. Temporal-distance and kinematics data were collected on all children by performing at least three walking trials. Data collection and processing was done using a six-camera Vicon System and Vicon Clinical Manager. The data for each child is the average of these trials.

In their work [4], O’Malley et al. provided four feature values: stride length, cadence, leg length and age. In our work for classification purposes, we considered all four features but were particularly interested in stride length and cadence. One of the main aims was to investigate how the normalization proposed by O’Malley et al. [17] on stride length and cadence, performs in distinguishing CP from normal children when applied to a classifier, especially support vector machine. These two features are particularly significant as they are considered to be the fundamental gait parameters [18]. These are also found to be more sensitive indicators of the degree of motor involvement among children with CP than other parameters, such as kinematics focused on single joints [19]. These features are statistically independent
when normalized by the respective leg length and age, and are easy to interpret as well as clinically useful [4].

B. Normalization of Gait Data

Temporal-distance gait parameters are affected by leg-length and age, therefore it is necessary to apply a proper normalization technique before applying to the classifiers. O’Malley et al. suggested polynomial normalization for each children group and feature [17]. Thus, the stride length ($s$) was normalized with respect to leg length as follows

$$
\hat{s}_n = s_n - (a_0 + a_1 l_n + a_2 (l_n)^2 + \cdots + a_k (l_n)^k) + \bar{s}_n
$$

$$
\hat{s}_p = s_p - (a_0 + a_1 l_p + a_2 (l_p)^2 + \cdots + a_k (l_p)^k) + \bar{s}_n
$$

where the subscripts ‘n’ and ‘p’ stand for control group children and children with CP respectively, $\hat{s}_n$ is the normalized stride length, $l$ is the leg length, $\bar{s}_n$ is the average stride length of control group children, and $a_0, a_1, \ldots, a_k$ are the coefficients of $k$-th order polynomial. Similarly, the cadence ($c$) was normalized with respect to age as follows

$$
\hat{c}_n = c_n - (b_0 + b_1 g_n + b_2 (g_n)^2 + \cdots + b_k (g_n)^k) + \bar{c}_n
$$

$$
\hat{c}_p = c_p - (b_0 + b_1 g_p + b_2 (g_p)^2 + \cdots + b_k (g_p)^k) + \bar{c}_n
$$

where $\hat{c}_n$ is the normalized cadence, $g$ is the age of individual children and $\bar{c}_n$ is the average cadence of the control group children.

III. Model Building and Testing

In the following subsections, we describe the underlying theory of support vector machine for classification and the metrics used to test its performance. We also briefly describe multi-layer perceptron and linear discriminant analysis classifiers whose performances are compared with the SVM classifier.
Support Vector Machines introduced by Vapnik [14] have recently been used for applications in many diverse classification and regression tasks. In a binary classification task, SVM aims to find an optimal separating hyperplane (OSH). Fig. 1(a) shows a two-class problem with infinitely possible hyperplanes separating the two data sets that are not necessarily optimal. In Fig. 1(b) an optimal separating hyperplane (OSH) is shown which generates the maximum margin (dashed line) between the two data sets. A further important concept in SVM is the transformation of data into a higher dimensional space for the construction of OSH. SVMs perform this nonlinear mapping into a higher dimensional feature space by means of a kernel function and then construct a linear OSH between the two classes in the feature space. Those data vectors nearest to the constructed line in the transformed space are called the support vectors (SV) and contain valuable information required for classification. SVM is an approximate implementation of the method of “structural risk minimization” [20]. The principle is based on the fact that minimizing an upper bound on the generalization error rather than minimizing the training error is expected to perform better. In brief, the theory of SVM is as follows [14].

Consider a training set \( D = \{(x_i, y_i)\}_{i=1}^{L} \), with each input \( x_i \in \mathbb{R}^n \) and the associated output \( y_i \in \{-1, +1\} \). Searching an OSH in the original input space is too restrictive in most practical cases. In SVM, each input \( x \) is first mapped into a higher dimension feature space \( \mathcal{F} \) by \( z = \phi(x) \) via a nonlinear mapping \( \phi: \mathbb{R}^n \to \mathcal{F} \). Considering the case when the data are linearly non-separable in \( \mathcal{F} \), there exists a vector \( w \in \mathcal{F} \) and a scalar \( b \) that define the separating hyperplane as: \( w \cdot z + b = 0 \) such that

\[
y_i (w \cdot z_i + b) \geq 1 - \xi_i, \quad \forall i
\]

where \( \xi_i \geq 0 \) are often called slack variables and only the misclassified data points \( x_i \) yields nonzero \( \xi_i \). The term \( \sum_{i=1}^{L} \xi_i \) can be regarded as a measure of misclassification. The OSH is determined so that the
maximization of the margin and minimization of training error is achieved, and the hyperplane that
optimally separates the data in $\mathcal{F}$ is the one that

$$\text{minimize } \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{L} \xi_i$$

subject to  $y_i (\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1 - \xi_i$, and  $\xi_i \geq 0$, $\forall i$ (6)

where $C$ is a constant parameter, called *regularization parameter*, that determines the trade off between
the maximum margin and minimum classification error. Minimizing the first term corresponds to
minimizing the VC-dimension (defined as the maximum number of training points that can be shattered
by a set of functions and a measure of complexity of the dimension space [20]) of the classifier and
minimizing the second term controls the empirical risk.

Searching the optimal hyperplane in (6) is a Quadratic Programming (QP) problem that can be solved by
constructing a Lagrangian and transformed into a dual. The optimal hyperplane can then be shown as the
solution of

$$\text{maximize } W(\alpha) = \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

subject to $\sum_{i=1}^{L} y_i \alpha_i = 0$ and $0 \leq \alpha_i \leq C$, $\forall i$. (7)

where $\alpha_1, \alpha_2, \ldots, \alpha_L$ are the non-negative Lagrangian multipliers. The data points $\mathbf{x}_i$ corresponding to $\alpha_i > 0$
lie along the margins of decision boundary and are the support vectors. The kernel function $K(\ldots)$
describes an inner product ($K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = \mathbf{z}_i \cdot \mathbf{z}_j$) in the space $\mathcal{F}$ that satisfies the Mercer's condition
[21] and can be computed without having to obtain $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ explicitly first. Having determined the
optimum Lagrangian multipliers, optimum solution for weight vector $\mathbf{w}$ is given by

$$\mathbf{w} = \sum_{i \in \text{SVs}} \alpha_i y_i \mathbf{z}_i$$ (8)

where SVs are the support vectors. For any test vector $\mathbf{x} \in \mathbb{R}^n$, the output is then given by
\[ y = f(x) = \text{sign}(w \cdot z + b) = \text{sign} \left( \sum_{i \in SVS} \alpha_i y_i K(x_i, x) + b \right) \]  

(9)

To build an SVM classifier, the user needs to tune \( C \) and choose a kernel function and its parameters. So far, no analytical or empirical study has conclusively established the superiority of one kernel over another; thus the performance of SVMs in a particular task may vary with this choice. In this study, we experimented with the following four kernels. The notation \( \langle \cdot, \cdot \rangle \) indicates an inner product.

<table>
<thead>
<tr>
<th>Kernel Function</th>
<th>Mathematical Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>( K(x_i, x_j) = \langle x_i, x_j \rangle )</td>
</tr>
<tr>
<td>Polynomial</td>
<td>( K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^d ), ( d ) is the degree of polynomial</td>
</tr>
<tr>
<td>Radial Basis Function (RBF)</td>
<td>( K(x_i, x_j) = \exp \left( -\frac{|x_i - x_j|^2}{2\sigma^2} \right) ), ( \sigma ) is the width of RBF function</td>
</tr>
<tr>
<td>Spline (ANOVA)</td>
<td>( K(x_i, x_j) = 1 + \frac{1}{2} \langle x_i, x_j \rangle + \frac{1}{6} \min(\langle x_i, x_j \rangle, 0) - \frac{1}{6} \min(\langle x_i, x_j \rangle)^3 )</td>
</tr>
</tbody>
</table>

Fig. 1 about here

B. Multi-Layer Perceptron

A three layer MLP has an input layer, a hidden layer and an output layer. Successive layers are fully connected by weights. An MLP updates the weights iteratively to map a set of input vectors \( (x_1, x_2, \ldots, x_p) \) to a set of corresponding output vectors \( (y_1, y_2, \ldots, y_p) \). An input \( x_i \) is presented to the input layer and multiplied by the weights. All the weighted inputs to each unit in the upper layer are then summed up and produce an output governed by the following equations.

\[ h_i = f(W_h x_i + \theta_h), \]

(10)

\[ y_i = f(W_o h_i + \theta_o), \]

(11)

where \( W_h \) and \( W_o \) and are the hidden and output layer weight matrices, \( h_i \) is the vector denoting the response of the hidden layer to input \( x_i \), \( \theta_o \) and \( \theta_h \) are the output and hidden layer bias vectors,
respectively and \( f(.) \) is the sigmoid activation function. The cost function to be minimized is the sum of squared error \( E \) defined as

\[
E = \frac{1}{2} \sum_i (t_i - y_i)^T (t_i - y_i)
\]

(12)

where \( t_i \) is the target output vector for pattern ‘\( i \)’. The standard backpropagation training algorithm [22] uses gradient descent techniques to minimize \( E \), but suffers from slow convergence and frequently stuck in local minima. There are a number of variations to backpropagation algorithm to achieve faster convergence and to avoid local minima. However, generalization ability of neural network is the most important factor. A desired neural network model should produce small error not only on sample data but also on out of sample data. A potential problem is the unsmoothing of the trained weights which may contribute to a network’s poor performance in generalization. To produce a network with better generalization ability, MacKay [23] proposed a method to constrain the size of network parameters by regularization. Regularization technique forces the network to settle to a set of weights and biases having smaller values. This causes the network response to be smoother and less likely to overfit [24] and capture noise. In regularization technique, the cost function \( F \) is defined as

\[
F = \gamma E + (1 - \gamma) E_w
\]

(13)

where \( E \) is the same as defined in (12), \( E_w = \| \mathbf{w} \|^2 / 2 \) is the sum of squares of the network parameters, and \( \gamma \) \((<1.0)\) is the performance ratio parameter, the magnitude of which dictates the emphasis of the training on regularization. A large \( \gamma \) will drive the error \( E \) to small value whereas a small \( \gamma \) will emphasize parameter size reduction at the expense of error and yield smoother network response. One approach of determining optimum regularization parameter automatically is the Bayesian framework. It considers a probability distribution over the weight space, representing the relative degrees of belief in different values for the weights. Using Gaussian probability distribution and Bayesian rule, the optimum value of \( \gamma \) at the minimum point of \( F \) can be determined. A detailed description of the method is available in [23].
C. Linear Discriminant Analysis

For a two class problem, \( n \)-dimensional data is projected onto a line according to a given direction \( \mathbf{w} \). The choice of the projection direction is determined by different criteria. The Fisher’s linear discriminant aims at maximizing the ratio of between-class scatter to within-class scatter \([25]\). Let \( I_y = \{i: y_i=y\}, y \in \{-1, +1\} \) be the sets of indices of training vectors belonging to the first and second class, respectively. The class separability in a direction \( \mathbf{w} \in \mathbb{R}^n \) is defined as

\[
F(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}
\]

where \( \mathbf{S}_b \) is the between-class scatter matrix and given by

\[
\mathbf{S}_b = (\mu_{-1} - \mu_{+1})(\mu_{+1} - \mu_{-1})^T
\]

\[
\mu_y = \frac{1}{|I_y|} \sum_{i \in I_y} \mathbf{x}_i
\]

\( \mathbf{S}_w \) is within-class scatter matrix and given by

\[
\mathbf{S}_w = \mathbf{S}_{+1} + \mathbf{S}_{-1}
\]

\[
\mathbf{S}_y = \sum_{i \in I_y} (\mathbf{x}_i - \mu_y)(\mathbf{x}_i - \mu_y)^T
\]

The maximum ratio of between-class to within-class scatter is reached at

\[
\mathbf{w} = (\mathbf{S}_{+1} + \mathbf{S}_{-1})^{-1}(\mu_{+1} - \mu_{-1})
\]

D. Testing a Classifier

Once a binary classifier is trained with the known gait data and the corresponding target associations (CP children=+1, normal children=-1), response to an unseen gait data from a new subject can be interpreted to belong to one of the categories. The following three measures (accuracy, sensitivity and specificity) \([12], [26]\) were used to assess the performance of the classifier.

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\%
\]
\[ Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (21) \]

\[ Specificity = \frac{TN}{TN + FP} \times 100\% \quad (22) \]

where TP is the number of true positives, i.e., the classifier identifies a CP gait that was labeled as CP; TN is the number of true negatives, i.e., classifier identifies a normal gait that was labeled as normal; FP is the number of false CP identification; and FN is the number of false normal identification. Accuracy indicates overall detection accuracy for both normal and CP gait patterns, sensitivity is defined as the ability of the classifier to accurately recognize a CP gait pattern whereas specificity would indicate the classifier’s ability not to generate a false detection.

In addition to the above measures, classifier’s performance was also evaluated in terms of receiver operating characteristic (ROC). ROC analysis is commonly used in medicine and healthcare to qualify the accuracy of diagnostic tests and evaluate the performance of intelligent medical systems [27]-[29]. ROC curve plots sensitivity against (1–specificity) as the threshold level of the classifier is varied and depicts the performance of a classifier without regard to class distribution. The area under the ROC curve (AUC) summarizes the quality of classification and is used as a single measure of accuracy. In this study, ROC curves were plotted and AUC values were calculated for all three classifiers to make a quantitative comparison in their ability to distinguish between normal and pathological gait data.

IV. Experimental Results

We employed \( k \)-fold cross-validation (a widely used technique) for model building, selection and testing. The total dataset of 156 children (88 with CP and 68 normal) was divided into ten nearly equal-sized segments: 6 segments containing 16 children and 4 segments containing 15 children. The ratio of normal and CP children in each segment was roughly the same in the total dataset. For each fold, ten segments were then grouped into training, validation and test sets. Each time one segment was kept as the test set,
of the remaining 9 segments, 8 were used as the training set to build models and 1 was used as the validation set to estimate the misclassification probability of the models. As suggested by Schwarzer et al. [30], the model with the smallest misclassification error rate on the validation set was then selected and the test set was used to obtain an unbiased estimate of the classification ability of the model. For the 10-fold cross-validation scheme, the process was repeated 10 times (test sets are disjoint), each time with a different test set. The average of the ten classification accuracy rates was then taken as an unbiased estimate of the model on the complete dataset and is reported here along with the standard deviation over the folds. The results of the 10 folds were combined into a single ROC plot and AUC calculation for each classifier.

In neural networks, the performance can vary with the number of hidden layers, initial weights and learning parameters. We experimented with a three-layer MLP with the number of input units the same as the number of features (2 as well as 4 features were investigated); one hidden layer with a varying number of units (between 2 to 10) and a single output layer unit representing the class label. For a fixed hidden size, 20 different MLP models were generated by varying the training parameters, initial weights and hidden units. The results (classification performance) presented are, therefore, based on the neural network model selected using the above mentioned model selection method. Similarly, in SVM the parameters that influence the performance are the regularization parameter $C$ and also the parameters that define the functions in some kernels. Experiments were conducted with values of $C$ ranging from $10^{-4}$ to $10^6$ and with different kernel functions as well as different kernel parameters ($\sigma$ was varied from $0.01$ to $2.0$ in RBF and $d$ was varied from 1 to 20 in polynomial kernel) to investigate kernel performance in distinguishing the CP gait characteristics. For each kernel type, the SVM classifier selected by the above model selection method is reported separately. All simulations were carried out in a Matlab environment using Matlab’s neural network toolbox (The MathWorks, Natick, MA), Gunn’s SVM toolbox [31], statistical pattern recognition toolbox for Matlab [32].
Effective preprocessing/normalization of raw data can play an important role in gait data analysis. In this study, we experimented using three sets of data containing different feature values: i) stride length and cadence without normalization, ii) all four features – stride length, cadence, leg length and age, and iii) polynomial normalization of stride length and cadence by leg length and age respectively as described in Section II.B. The polynomial coefficients used for normalization were adopted from [4]: \( a_0 = 0.28 \text{m}, a_1 = 1.31, \quad s_n = 1.02 \text{m}, \quad b_0 = 174.07 \text{steps/min}, \quad b_1 = -7.04 \text{ steps/min.yr}, \quad b_2 = 0.22 \text{ steps/min.yr}^2, \quad c_n = 136.84 \text{ steps/min}. \)

Table I presents the overall accuracy, sensitivity and specificity of LDA, MLP and SVM (RBF kernel) classifiers for recognition of cerebral palsy with different raw and normalized gait data. In all cases, each gait feature was scaled by calculating their z-scores (i.e., \((x-\mu)/\sigma\), where \(\mu\) is the mean and \(\sigma\) is the standard deviation for the gait feature) before applying them to the classifier. With raw stride length and cadence data, LDA and MLP achieved 80.13% and 81.41% accuracy, respectively while SVM achieved higher accuracy (83.33%). In this case, LDA suffered from wider differences in sensitivity and specificity results (72.73% vs 89.71%) while both MLP and SVM attained relatively close sensitivity and specificity (80.68% vs 82.35% in MLP and 82.95% vs 83.82% in SVM). A consistently larger difference in sensitivity and specificity in LDA in comparison with MLP and SVM was observed for all the three feature sets, which indicate that for the given dataset, a linear discriminant function is less effective in differentiating CP children from normal children. Classification performance improved significantly (above 10%) for the classifiers when all the four features were used in model building. Further improvement was achieved when polynomial normalization of stride length and cadence were used to build the classifiers. With such normalization SVM achieves an overall accuracy of 96.80% which is a significant improvement by applying normalization on only two features. SVM also displayed better accuracy rate than MLP and LDA by a margin of 1.93% and 3.21% respectively. Normalization also reduced the standard deviation of accuracy rates over the folds for all classifiers. Fig. 2(a) confirms better performance by the SVM classifier with normalized data over a range of threshold values as demonstrated.
by the greater ROC area and higher sensitivity at all specificity values. With SVM classifier, normalization of gait data increases AUC from 0.87 to 0.99.

Table II shows the performance of the SVM classifier with respect to four kernel types: linear, polynomial, RBF and ANOVA spline. All the three performance metrics varied slightly with the kernel choice. Overall accuracy with respect to kernel function varied by a maximum of 2.56% with raw stride length and cadence, by 1.28% with all features and by 1.27% with polynomial normalization. Irrespective of the kernel choice in most occasions, the SVM classifier outperformed the MLP and LDA classifiers for this data.

The superior performance of the SVM classifier was further substantiated by the ROC plots and the area under the curves. Fig. 2(b) shows the ROC curves of the three classifiers displaying the relationship between sensitivity and (1–specificity) for a range of thresholds. The results indicate that consistently higher sensitivity values were observed for all specificity values in an SVM classifier. The AUC values for SVM (RBF), MLP and LDA classifiers were 0.99, 0.94 and 0.93, respectively. The classifiers’ sensitivity values at 0.9 specificity were 0.97, 0.93 and 0.89, and at 0.8 specificity were 0.98, 0.94 and 0.90 respectively. The comparative AUC values attained by the classifiers are in agreement with the sensitivity, specificity and overall accuracy measures presented in Table I.

Fig. 3 illustrates distribution of the subjects over the feature space using raw stride length and cadence and their normalized values. There is a fair degree of overlapping between the data of the control group and children with CP without any normalization as can be seen in Fig. 3(a). With polynomial normalization overlapping between the two classes decreased significantly (see Fig. 3(b)). The increased separation of the subjects across the two groups in the feature space plays an important role in achieving better classification due to data normalization.
Fig. 4 (a) shows the scatter plot of one data segment used as the test data along with the separating hyperplane for linear and RBF kernels for the raw data, while Fig. 4 (b) shows the same for the normalized gait data. The figure shows the visual representation of the individual subjects where the misclassified test data are located on the wrong side of the separating hyperplane. This segment had substantial overlapping between the CP and normal children group and was particularly difficult to classify correctly. Figs. 4(b) & (d) illustrate that in both raw and normalized data, the RBF kernel was able to create highly non-linear and better separating boundaries. Furthermore, when compared with Fig. 4(b), the “ad hoc” transformation performed by the polynomial normalization placed the data points of Fig. 4(d) farther from the separating hyperplane.

We further investigated the effect of parameter selection on SVM classifier’s ability to correctly identify CP and normal children using normalized stride length and cadence data. The regularization parameter $C$ in SVM determines a trade-off between the training error and the probability of error in unseen patterns. Figs. 5 (a) & (b) plot the effect of regularization parameter on overall accuracy for polynomial and RBF kernels. Applying the similar model selection criterion as described earlier, for each value of $C$, kernel parameter ($d$ or $\sigma$) were varied to generate models and the accuracy on the test data on the basis of the lowest error rate on the validation set is taken. Accuracy results presented in the figure is the average on ten-fold cross-validation method. Results show that there exists a narrow range of $C$ values (0.07−2.0 for polynomial and 0.0001−7.0 for RBF) which best classifies the gait patterns. For a wide range of $C$ values ($C>10$), the performance, however, does not change significantly.

The transformation of the original input space into high dimensional space and hence the kernel matrix depends on the parameters that define the kernel function, e.g., degree of polynomial ($d$) and width of the RBF ($\sigma$). The effect of these parameters on overall accuracy is shown in Fig. 6. Parameters $d$ and $\sigma$ were varied from 1 to 20 and 0.01 to 2.0, respectively. In this case, the value of $C$ was varied ($10^{-4}$ to $10^{6}$) for
model selection for each $d$ or $\sigma$. Usually the value of $d$ is assigned in the range 2~7 and that of $\sigma$ between 0.5~1.5 in most applications, e.g., in [33]. In this work, overall performance of the classifier did not vary significantly due to variation in kernel parameters. In polynomial kernel, the best accuracy was obtained with $d=5$, remained unchanged till $d=18$, and the overall accuracy varied by 1.2% over the range investigated. Interestingly, the accuracy drops from $d=1$ to $d=2$, 3 before improving (Fig. 6 (a)). This type of variation may be data dependent and is also evident in other studies [33, 34]. In RBF, accuracy was relatively low for very small value of $\sigma$ (below 92.3% for $\sigma \leq 0.05$), however, attained the best accuracy level for $\sigma \geq 0.1$ and varied by 1.2% between $\sigma=0.1$ to 2.0.

********** Figs. 5 & 6 about here  **********

V. Discussion

The present study primarily aimed at investigating: i) the applicability of SVM as a machine classifier to distinguish children with cerebral palsy from normal children, ii) the effectiveness of polynomial normalization proposed by O’Malley et al. [4] in such classification in comparison with raw data and iii) the influence of regularization and kernel function parameters on the overall accuracy of the SVM. Results of this study suggest that SVM is capable of classifying the children with cerebral palsy with significantly high accuracy rate (nearly 97%). An SVM classifier performed superiorly to both MLP and LDA approaches irrespective of the gait data normalization and the number of gait features. With raw gait data, inclusion of leg length and age improved recognition accuracy for all classifiers, but in both cases (2 or 4 features), the SVM outperformed other classifiers (Table I). With normalized gait data, SVM achieved about 2% higher accuracy over MLP and >3% over LDA classifier. It also achieved 100% specificity indicating its excellent ability to avoid false detection. To test the statistical significance of the results attained by the classifiers, we conducted ten cross-validation experiments with normalized gait data, using ten different sets through repartitioning of the data [35]. The Wilcoxon rank sum test, to test the significance of difference between the accuracy of SVM (RBF) and that of MLP and LDA, yielded $p$-
values of 0.000127 and 0.000124, respectively. The mean value of the kappa coefficient of the confusion matrix for SVM, MLP and LDA classifiers displayed 0.93, 0.88 and 0.84, respectively. These results suggest that the improved performance of SVM over other classifiers is statistically significant. Previous research applying SVM in other classification [36], [37] and regression tasks [38], [39] has reported its robust performance, and in many binary classification applications, SVM has been found to outperform other techniques, e.g., neural networks [40]. Superior performance demonstrated by the SVM in gait pattern detection and classification suggests its potential for applications in CP gait identification, diagnosis and assessment through the use of SVM output, similar to the one used by Hahn et al. [3].

The results presented in Tables I & II indicate that normalization of the gait data can significantly affect the outcome of an SVM classifier. Without normalization, the overall accuracy of the classifier was low (about 83%) and such performance might be of limited use in gait pattern recognition. In contrast, normalization resulted in significant improvement in classification, sensitivity and specificity accuracy. The use of four feature values improved overall accuracy, sensitivity and specificity considerably but normalized data of only two features produced better results. The effectiveness of this normalization along with the classification technique was further supported by the area under the ROC curve (AUC) which is a more complete representation of the classification performance and has been widely used in evaluating medical expert systems. Normalization yielded greater AUC (0.99 vs 0.87) compared to when no normalization was used.

The improved prediction outcomes obtained as a result of gait data normalization can be explained by the distribution of data over the input features as illustrated in Fig. 3. As can be seen, both the control & CP groups overlap considerably in the feature space when the gait data are not normalized (Fig. 3(a)). However, overlapping between the two groups decreased when polynomial normalization was applied to stride length and cadence data (Fig. 3(b)). This is supported by the standard deviation results of the control and CP groups as presented in Table III, which indicate that with polynomial normalization each group becomes more densely clustered. The centers (stride length, cadence) of the control and CP group
clusters, and the associated normalized Euclidean distance (measured by scaling each feature by its maximum value) between them are presented in Table IV. The distance between the control and CP group cluster centers was increased significantly (0.318 vs 0.22) when the normalization technique was applied. The polynomial normalization definitely helped the classification process and resulted in the control and CP group children to become well separated (but not completely) in the feature space (Fig. 3(b) vs Fig. 3(a)). The SVM performed the best in classification tasks using the normalized data and by creating an optimal separating hyperplane that led to the maximization of the margin between the classes.

******** Table III & IV here ********

The regularization parameter $C$ controls the trade-off between the model complexity (the first term in (4)) and the training error (the second term) in order to ensure good generalization capability. The optimum value for $C$ is usually determined experimentally by trial and error. When the problem is non-linearly separable in the high dimensional space transformed by kernel function, selecting too high value for $C$ would penalize the training error heavily and compromise the error on the test set. However, too low value would mean the model would not learn from the training data properly, thereby resulting in underfitting and leading to inferior performance. Keerthi et al. [41] suggested determining its value by studying asymptotic behaviour of the generalization error and then finding a good region in which the parameter value is most likely to produce the best generalization. However, this method is specific to RBF kernel and computation intensive. Haykin [20] suggested calculating $C$ analytically by estimating the VC dimension of the classifier, but determining the optimum value still remains an area for further research. Results presented in Fig. 3 suggests that, for both polynomial and RBF kernels, the best overall accuracy rate is obtained within a smaller range of $C$ but does not change significantly over a wide range of higher values. This kind of behavior, i.e., minimal effect of $C$ on performance when $C$ is larger than a particular threshold is also supported in a recent study by Cherkassky et al. [42]. As Cherkassky et al. pointed out there have been many conflicting strategies suggested for setting the value of SVM parameters. In [42], they proposed a methodology to analytically determine $C$. The method is claimed to determine “practical”
values of $C$, however, whether this method can select the best or optimum value is yet to be verified in a wide range of real world problems.

As regards to the choice of kernel function no theoretical proof yet exists supporting the advantage of one over the others. Factors determining the choice of kernel functions is many, e.g., the type and size of the data, the underlying function that generated the data, noise-to-signal ratio, computational cost etc. In practice, various models are tested with different kernel functions and the one that offers the best generalization performance for that particular application is selected. In the current application of children CP gait recognition, the variation in overall accuracy with $d$ and $\sigma$ was typically around 1.2%. However, for very small values of $\sigma$ ($\leq 0.03$), RBF kernel performs highly nonlinear mapping on the input data and can overfit the training data irrespective of $C$ value. With such value of $\sigma$, the model learns all the training data correctly but performs poorly on the validation as well as the test data as shown in Fig. 6(b).

So far, the application of SVM in gait analysis has been limited. It has been applied to gender [43] and ageing [15] classification tasks with considerably high accuracy rate (>92%). This study demonstrates SVM’s applicability in identifying children with cerebral palsy with significantly higher accuracy (~96.8%) using only two temporal distance gait parameters. Previously, gait pattern analysis using neural networks by Holzreiter et al. [44] reported close to 95% accuracy in detecting pathological and normal gait. But this level of accuracy was obtained using kinetic features measured by two ground reaction force platforms which involve a more complicated and expensive data acquisition procedure involving many features. Our study using easily measurable two gait data (stride length and cadence) attains nearly 97% accuracy and significantly high value of AUC with an SVM based classifier. Such superior performance demonstrates potentials for SVM’s future applications in clinical gait diagnosis and assessment of gait improvements as a result of various rehabilitation techniques. Furthermore, there are other gait variables (e.g., ankle, knee and hip joint angles, lower limb muscle activity patterns, etc) that have been reported to be affected in cerebral palsy gait [1, 10]. In future studies, inclusion of gait
features derived from these variables might further improve the classification performance of the SVM.

References


TABLE I

OVERALL ACCURACY, SENSITIVITY AND SPECIFICITY OF DIFFERENT CLASSIFIERS WITH RAW AND POLYNOMIAL NORMALIZATION OF GAIT DATA BASED ON TEN-FOLD CROSS VALIDATION. THE NUMBER WITHIN THE BRACKET INDICATES STANDARD DEVIATION OVER THE FOLDS.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Performance measure</th>
<th>Stride length and cadence (raw)</th>
<th>All four feature values</th>
<th>Stride length and cadence (normalized)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>Accuracy</td>
<td>80.13 (±10.33)</td>
<td>91.02 (±6.68)</td>
<td>93.59 (±6.59)</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>72.73 (±16.72)</td>
<td>86.36 (±10.73)</td>
<td>89.77 (±9.63)</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>89.71 (±11.08)</td>
<td>97.05 (±4.86)</td>
<td>98.53 (±3.78)</td>
</tr>
<tr>
<td>MLP</td>
<td>Accuracy</td>
<td>81.41 (±13.12)</td>
<td>93.59 (±5.10)</td>
<td>94.87 (±5.75)</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>80.68 (±12.67)</td>
<td>92.04 (±6.36)</td>
<td>94.32 (±6.28)</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>82.35 (±14.43)</td>
<td>95.59 (±4.94)</td>
<td>95.59 (±5.06)</td>
</tr>
<tr>
<td>SVM (RBF)</td>
<td>Accuracy</td>
<td>83.33 (±12.09)</td>
<td>94.87 (±7.20)</td>
<td>96.80 (±4.42)</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>82.95 (±13.56)</td>
<td>93.18 (±8.91)</td>
<td>94.32 (±8.27)</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>83.82 (±11.34)</td>
<td>97.06 (±5.14)</td>
<td>100 (0.0)</td>
</tr>
</tbody>
</table>
TABLE II

OVERALL ACCURACY, SENSITIVITY AND SPECIFICITY OF SVM CLASSIFIER WITH DIFFERENT KERNEL TYPES
BASED ON TEN-FOLD CROSS VALIDATION. THE NUMBER WITHIN THE BRACKET INDICATES STANDARD
DEVIATION OVER THE FOLDS.

<table>
<thead>
<tr>
<th>Kernel type</th>
<th>Performance measure</th>
<th>Stride length and cadence (raw)</th>
<th>All four feature values</th>
<th>Stride length and cadence (normalized)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>Accuracy</td>
<td>82.05 (±11.72)</td>
<td>93.59 (±3.34)</td>
<td>95.51 (±5.27)</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>79.54 (±12.94)</td>
<td>92.04 (±4.85)</td>
<td>94.32 (±7.66)</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>85.29 (±9.26)</td>
<td>95.59 (±4.06)</td>
<td>98.53 (±4.17)</td>
</tr>
<tr>
<td>Poly</td>
<td>Accuracy</td>
<td>82.69 (±12.38)</td>
<td>94.23 (±7.43)</td>
<td>96.79 (±4.88)</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>79.54 (±13.95)</td>
<td>94.32 (±8.66)</td>
<td>94.32 (±8.76)</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>86.76 (±11.47)</td>
<td>94.11 (±6.93)</td>
<td>100 (0.0)</td>
</tr>
<tr>
<td>RBF</td>
<td>Accuracy</td>
<td>83.33 (±12.09)</td>
<td>94.87 (±7.20)</td>
<td>96.78 (±4.42)</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>82.95 (±13.56)</td>
<td>93.18 (±8.91)</td>
<td>94.32 (±8.27)</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>83.82 (±11.34)</td>
<td>97.06 (±5.14)</td>
<td>100 (0.0)</td>
</tr>
<tr>
<td>Spline (ANOVA)</td>
<td>Accuracy</td>
<td>80.77 (±12.85)</td>
<td>93.59 (±3.34)</td>
<td>95.51 (±5.27)</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>80.68 (±13.27)</td>
<td>93.18 (±5.39)</td>
<td>94.32 (±7.06)</td>
</tr>
<tr>
<td></td>
<td>Specificity</td>
<td>80.88 (±11.56)</td>
<td>94.11 (±4.74)</td>
<td>98.53 (±4.73)</td>
</tr>
</tbody>
</table>
**TABLE III**

**Variability (Standard Deviation) of Stride Length and Cadence Data for the Control and CP Groups With and Without Normalization Techniques.**

<table>
<thead>
<tr>
<th>Data type</th>
<th>Control group</th>
<th>CP group</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>stride length</td>
<td>cadence</td>
<td>stride length</td>
<td>cadence</td>
</tr>
<tr>
<td>Raw data</td>
<td>0.180</td>
<td>15.81</td>
<td>0.210</td>
<td>31.49</td>
</tr>
<tr>
<td>Polynomial normalization</td>
<td>0.096</td>
<td>10.98</td>
<td>0.170</td>
<td>29.28</td>
</tr>
</tbody>
</table>

**TABLE IV**

**Centers of the Control and CP Group Clusters, and the Normalized Euclidian Distance Between the Centers in Each Case.**

<table>
<thead>
<tr>
<th>Data types</th>
<th>Control group center</th>
<th>CP group center</th>
<th>Norm. distance between centers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw data</td>
<td>(1.023, 136.84)</td>
<td>(0.736, 121.04)</td>
<td>0.220</td>
</tr>
<tr>
<td>Polynomial normalization</td>
<td>(1.015, 136.64)</td>
<td>(0.603, 127.82)</td>
<td>0.318</td>
</tr>
</tbody>
</table>
Fig. 1. An example of two-class (+ & -) problem with: a) many possible separating hyperplanes dividing the two groups, b) optimal separating hyperplane and the maximum margin. The circles and squares represent samples of class +1 and -1, respectively.
Fig. 2. ROC curves and the area under the curves for a) SVM with and without data normalization and b) SVM, MLP and LDA classifiers with normalized data.
Fig. 3. Distribution of the subjects in feature space (a) raw data without normalization (b) polynomial normalization of stride length and cadence by leg length and age.
Fig. 4. Scatter plots showing the test data and separating hyperplane using raw and normalized gait data by the SVM classifier with linear and RBF kernel. (a) with raw stride length and cadence (b) with normalized stride length and cadence.

(a)

(b)
Fig. 5: The overall classification accuracy vs regularization parameter $C$ in case of data normalization with two kernels (a) polynomial (b) radial basis function.
Fig. 6. Variation of overall performance as a function of kernel parameters (a) $d$, degree of polynomial (b) $\sigma$, width of radial basis function.