An Optimized Support Vector Machine for Classifying Opinions in M Learning Systems Applied To Biotechnology Domain

Mahalakshmi K. [1], Mathi Vanan P.[2], Mohana Priya D. [3], Keerthika R. [4] and Nagajothi A.[5]
1. Department of Information Technology, Karpagam College of Engineering, Coimbatore, INDIA
2. Department of Information Technology, Manakula Vinayagar Institute of Technology, Villupuram, INDIA
3. Department of Information Technology, Mailam Engineering College, Villupuram, INDIA
4. Dept of Information Technology, Karpagam College of Engineering, Coimbatore, INDIA
5. Dept. of CSE, Karpagam College of Engineering, Coimbatore, INDIA

Abstract
The new trends in teaching or learning process are mobile learning (M-learning) as well electronic learning (E-learning). Nowadays distance education is gaining more importance and leads to E-learning and M-learning. These technological educational models are widely preferred by learners. This work proposes an innovative approach for Charged System Search (CSS), Particle Swarm Optimization algorithm (PSOA) as well as Support Vector Machines (SVM). In PSO technique, there is a set of Pareto-optimum solutions. It is used to choose the global best for all particles in the population. This strongly influences the convergence as well as diversity of solution. In order to overcome the issue, this paper incorporates CSS technique into the search procedure of the PSO protocol. In CSS technique, every particle is conducted by the personal best as well as the resultant force acts on this particle. The classification problems in various domains are solved using the SVM classification. SVM classification is an active research area. Distinct test functions are utilized for testing the CSS technique and the outcomes were weighed against the results of three modern multi-objective protocols.

Keywords: Mobile Learning (M-Learning), Electronic Learning (E-Learning), Charged System Search (CSS), Particle Swarm Optimization (PSO), Support Vector Machine (SVM).

Introduction

M-Learning, a kind of e-learning is capable of delivering learning as well as supporting materials, using wireless communication devices. Similarly, mobile learning is easy to use as a personalized, connected as well as for interactive utilization through devices such as laptops in classrooms. It is employed in combined training during fieldtrips, for counselling as well as in supervision. M-learning is empowered using the progress in mobile technology interfaces, especially the omnipresent android OS. Hence, all the modern learning methods especially in biotechnology domain are currently enabled via M-Learning. [1, 2]

Multiple platforms, languages and technologies are used in developing M-Learning. This method of learning is of great use to the students as they can directly use learning contents through mobile instead of searching for proprietary materials which are not easily accessible. The smart phones are becoming widely accepted by the users as it could be customised according to specific needs.

This feature enables M-learning to be widely accepted in teaching and learning procedure. The learners are able to manage and diversify their learning activities more easily. For instance, missed lectures can be instantly downloaded by the students. It is obvious that M-Learning systems derived from android technology are dominating the realm of M-learning as, engaging multimedia contents like audio, videos as well as animations may be easily transferred to the mobile devices.

There are several merits and demerits of E-Learning over M-Learning. A few of the characteristics are discussed. Cost: In e-learning inexpensive desktop computers are used when compared to m-learning where modernised Wi-Fi supported mobile devices are used. The connectivity and technology implicated in mobile computing wireless structures are costlier than cable/wire line Local Area Network (LAN) which is generally employed in e-learning. Input Capabilities: In e-learning, user friendly input devices like keyboard, mouse are linked to desktop computers when compared to input facilities of mobile gadgets utilized in m-learning. Output Capabilities: Similarly, the output devices namely the desktop monitor or screens that are used in e-learning are easy to use than the screens of mobile device. Processing Power: A desktop PC which is utilized in e-learning has high processing speed as well as Central Processing Unit (CPU) when compared to mobile devices utilized in m-learning. Memory: E-learning uses desktop PC which is capable of storing large amount of data as it has vast memory when compared to mobile devices used in m-learning. [3]

A mobile device and desktop computer varies from each other in many aspects. A few of the characteristics were discussed above namely the output that is, the size of the screen, resolution features, and so on; input, that is, keypads, touchscreen and voice inputs; processing power, memory, application as well as their media types. The electronic and mobile learning platforms varies in compatibility as when the services of e-learning platforms are transferred into applications in m-learning platforms it could be observed that a few of the services should undergo certain changes to meet the restrictions of small mobile devices while a few of the services are not viable to be conveyed reliably. On other hand, new applications are possible in m-learning as it is provoked by the mobility. [4] The features of E- as well as M-learning are differentiated as given in Table 1.

Table 1
Comparison of E-Learning to M-Learning

Comparison	E-Learning	M-Learning
Portability (easy to carry)	Desktop PCs are not portable	Mobile Devices are Portable
Flexibility	Not Flexible	Flexible
Freedom of Learning	Not Anywhere and Anytime	Anywhere and Anytime
Cost of Devices	Less Expensive	More Expensive
Cost of Technology	Less Expensive	More Expensive
Location Education	Cannot Provide Through GPS	Can Provide Through GPS

M-learning is derived from E-learning (a mode of learning made possible through the progress of varied communication methods) while e-learning is derived from d-learning i.e. distance learning. The learners using e-learning will have the intention to learn new things and particularly they wish to get specific knowledge or expertise. In E-learning, terms such as tethered which means connected to something, are basically attached and also it offers learning in a prescribed and planned manner. Conversely, M-learning is un-tethered and it is the major difference when compared to e-learning. This form of learning is more comfortable and prospective. It is also more personal, convenient as well not properly framed. The E-learning procedure is facilitated through software and internet tools. But the learning procedure in M-learning is aided by using mobile devices like smart phones, PDAs as well as iPods.[5]

The fast growth in mobile technologies is a huge support for e-learning within the model of d-learning. The idea of m-learning offers greater technical advancements in learning procedure. Despite e-learning being highly advantageous when compared to conventional learning, the drawbacks of it have guided the scientists in search of new concepts. The concept of m-learning is a result of advanced mobile technologies along with immediate requirements to reform education. The significant benefit of m- over e-learning is that the user can access any information irrespective of time and surroundings.[6]

M-Learning is a form of education that evolved as a result of coevaluation of both mobile informatics as well as e-learning domains. It offers access to e-learning materials without depending on location; the services can be utilized dynamically and easy communication with others. M-Learning is capable of supporting conventional and distance learning. By analysing the advantages of mobile education, the following can be deduced; learning for lifetime, learning without intention, learning when necessary, learning irrespective of time and environment, and learning with respect to surroundings and situations[2].

In this work, the M-Learning systems using CSS, PSO and SVM are presented. Section 2 reviews literature related to the proposed work. Section 3 illustrates methodology and Section 4 examines experimental results conducted in proposed work. Section 5 presents the conclusion of the work.

Related Works

Cheon et al[7] put forth a theoretical representation inspired by the Theory of Planned Behavior (TPB). This illustrates the way the students trust mobile devices and how it influences to implement mobile devices in their assignments. Self report data was obtained from 177 college students and was analysed using Structural equation modelling. From the results it can be understood that TPB explains the acceptance of mobile learning by college students spread far and wide. Especially the view-point, subjective norm, and control of behaviour have had strong influence on their perspective to adopt mobile learning. From the output, we can infer several means to improve recognition of m-leaning by college students.

Chen and Tseng[8] proposed the Technology Acceptance Model (TAM) as an abstract basis as well as utilized the Structure Equation Model (SEM) to evaluate the aspects, which influences the intention to utilize in-service training carried out via web-based e-learning. From the outputs, it is clear that the inspiration to use internet as well self-efficiency were notably positive and are related to behavioral intention concerning the usage of web-based e-learning for in-service training via the factors of perceived utility as well as perceived easy usage.

Kaneda et al[9] introduced the SVM as one among the finest machine learning models, which can probably offer high precision for recognition as well regression. The major setback in employing SVM is that the expense to implement the system is directly related to the quantity of training data as well as the dimensions of features space. In this work, the author has used Dimensionality Reduction (DR) to find solution for the problems.

Couellan et al[10] presented the fundamentals of SVM learning from a multi-agent optimization perspective. The complicated optimization problems are broken into basic “oracle” tasks by the multi-agent’s systems. The composite problems are solved through a combined process which results in a self-organized solution. The author also illustrates the methods to ‘tackle’ the problems in SVM as well offer different views for binary classification, for selecting hyper parameters, multiclass learning and unsupervised learning.

SVM kernel optimization for improving non-functional requirement classification was addressed in[11,12]. Optimization methods like Artificial Bee Colony (ABC), Differential Evolution (DE) and Hybrid ABC-DE were incorporated to improve the efficacy of SVM.

Methodology

This section uses the dataset obtained from different m learning apps from Google play (solo learn) and the SVM class labels namely positive, negative and opinion. Description for pre-processing techniques such as stop words removal, stemming and TF-IDF, the SVM classifier, the SVM optimized PSO algorithm and the SVM optimized CSS algorithm are given.

Stop Words Removal: Common stop words frequently used are 'and', 'are', 'this' etc. Such words are helpless in classifying the documents and so must be eliminated. Conversely, it is hard to develop these stop words list as well it is incompatible among textual sources. The procedure minimises text data and improve the system execution. Each text document trades with such words that are needless for text mining applications. Several stop words are powerfully related to discrete mathematical word problem types. For example, as & times in phrases such as five times as much as are fine indicators of Multiplicative Compare (MC) problem, similarly' each' in phrases such as each lesson and 'in all' in phrases like 12 songs in all are referring to an Equal Group (EG) problem. Hence, the widely acknowledged text pre-processing technique to eliminate stop words must be prevented in classifying mathematical word problems. [13]

Stemming: Stemming[14] is a procedure where different forms of a word are conflated into a single form, the stem. For instance, the expression: "presentation", "presented" and "presenting" can be represented as "present". This procedure is mainly used to process text for Information Retrieval (IR). It assumes that posing a query with the term 'presenting' means an interest in documents that consists of words like presentation as well presented. Stemming or suffix removal emphasizes the importance of word suffixes in categorizing texts and the words having the same theoretical meaning like computation, computing and compute must be categorized under the same word stem compute. But when categorizing the mathematical word problems, stemming affects the classification performance as few of the words related to dissimilar mathematical word problems are changed into word stems, which exist in any kind of mathematical word problem. For example, 'its', times that are largely related to MC problems are transformed into word stem it, time which doesn't indicate MC problems.

Term Frequency-Inverse Document Frequency (TF-IDF): The significance of a phrase within a document is indicated through Term weighting method. In sentiments classification, TF as well TF-IDF is generally used to quantify (the weight is counted) a word. Term Frequency indicates how many times a word is repeated in a text whereas TF-IDF signifies the combination of TF as well as Inverse Document Frequency (IDF) weights. IDF signifies the basic significance given for a phrase in all the documents. IDF as well as TF-IDF [15] can be calculated as equations (1) and (2):

$$idf = \frac{The\ number\ of\ total\ documents}{The\ number\ of\ documents\ include\ a\ term} \tag{1}$$

$$tf = tf * idf \tag{2}$$

If TF-IDF count for a term is huge then it implies that the term occurs repeatedly in various parts of the documents.

Support Vector Machine (SVM): SVMs are a group of associated supervised learning technique that can be applied for classification as well as regression. SVM is a part of comprehensive linear classification family. It possesses a unique feature using which it reduces the empirical classification error to minimum as well increases the geometric margin to maximum concurrently. Hence, SVM are known as Maximum Margin Classifiers. Structural Risk Minimization (SRM) forms the basis for SVM and its map input vector to a larger dimensional space is by constructing a maximum separating hyperplane. 2 parallel hyperplanes are built on either side of the hyperplane, which splits data. Separating hyperplane is the one that makes maximum the distance among 2 parallel hyperplanes. It is assumed that when the margin or distance among the parallel hyperplanes is huge then the generalization error of the classifier is good. [16]

It considers data points given in the form (3):

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4).........., (x_n, y_n)\} \tag{3}$$

From the above equation $y_n = 1/-1$, is an invariable that denotes the class to which the point xn is a part of. n = quantity of samples. Every xn represents p-dimension real vector. The scaling is essential to protect against parameters with bigger variances. The training dataset can be viewed, by dividing hyper plane, as (4):

$$w\,.\,x + b = o \tag{4}$$

Wherein b represents scalar while w represents a p-dimension Vector. Vector w points perpendicular to separating hyper plane. When offset, variable b is added, the margin gets increased. In the absence of b, hyperplane is made to pass via the origin, limiting the solution. Parallel hyperplanes may be illustrated as in (5):

$$w.x + b = 1$$
$$w.x + b = -1 \tag{5}$$

The linearly separable training dataset may choose these hyper planes such that there aren't any points existing among them and later it tries to increase their distance to maximum. By geometry, the distance between hyperplanes is 2 / │w│.

Hence it requires minimizing |w|. In order to stimulate data points, it needs to verify that for every I either in (6):

$$w.x_i \;.\; b \;.\; 1 \; or \; w.x_i \;.\; b \;.\; -1 \tag{6}$$

It may be stated in (7):

$$y_i \;(w.x_i \;.\; b) \geq 1\,, \quad 1 \leq i \leq n \tag{7}$$

Examples along the hyper planes are known Support Vectors (SVs). A separating hyper plane with the biggest margin is given by M = 2 / |w| .The specified support vectors signify training data points closets to it equation (8). Which satisfy:

$$y_j[w^T.x_j + b] = 1 \qquad , i = 1 \tag{8}$$

Optimal Canonical Hyperplane (OCH) [17] refers to a canonical hyperplane that has maximal margin. For every given data, OCH ought to fulfil the below limitations given in equation (9):

$$y_i[w^T.x_i + b] \geq 1 \qquad ; i = 1, 2......1 \tag{9}$$

Here l represents Number of Training data point. Consequently, to determine the optimum separating hyperplane that has maximal margin, learning machines must reduce ‖w‖2 to minimum that are subjected to the inequality constraints in equation (10):

$$y_i[w^T.x_i + b] \geq 1 \qquad ; i = 1, 2.........1 \tag{10}$$

This optimization issue is resolved using saddle points of Lagrange's Function in equation (11):

$$L_p = L_{(w,b,\alpha)} = \tfrac{1}{2}\| w \| 2 - \sum_{i=1}^{1} \alpha_i (y_i (w^T x_i + b) - 1)$$
$$= \tfrac{1}{2} w^T w - \sum_{i=1}^{1} \alpha_i (y_i (w^T x_i + b) - 1) \tag{11}$$

Here αi represents Lagrange's multiplier. Finding optimum saddle points (w_0, b_0, α_0) is essential. Since Lagrange's must be reduced to minimum corresponding to w and b and must be increased to maximum corresponding to non-negative αi (αi ≥ 0). The issue may be resolved in primal format i.e. format of w and b or in dual format i.e. the format of αi. Partially differentiate equations corresponding to saddle points (w_0, b_0, α_0) in equation (12-15).

$$\partial L / \partial w_0 = 0 \tag{12}$$

$$w_0 = \sum_{i=1}^{1} \alpha_i \; \mathrm{y}_i \; \mathrm{x}_i \tag{13}$$

$$\partial L / \partial b_0 = 0 \tag{14}$$

$$\sum_{i=1}^{1} \alpha_i \; \mathrm{y}_i = 0 \tag{15}$$

Substituting above equation, the primal form is changed into dual form (16).

$$L_d(\alpha) = \sum \alpha_i - 1/2 \sum_{i=1}^{1} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{16}$$

Subsequently, for discovering the optimum hyper plane, a dual Lagrangian (Ld) must be increased to maximum corresponding to non-negative αi (where αi should be in the non-negative quadrant) along with equality constraints as (17):

$$\alpha_i \geq 0 \qquad ,\mathrm{i=1,2......1}$$
$$\sum_{i=1}^{1} \alpha_i y_i = 0 \tag{17}$$

It is to be noted that the dual Lagrangian Ld(α) is represented with respect to training data as it depends solely on the scalar product of input patterns $(\mathrm{x}_i^T \mathrm{x}_j)$.

The kernel technique is a protocol, which relies on data solely via dot product and SVM belonging to the basic group of kernel techniques [18]. Under such circumstances the dot product is substituted by kernel functions that processes the dot products in a higher-dimension features space. The 2 merits are: Firstly, it has the capability to produce non-linear decision boundaries utilizing techniques devised for linear classifier. Secondly, by using kernel function user is permitted to classify data using classifier that doesn't have a clear fixed-dimension vector space representations.
The RBF is given as equation (18):

$$K(x_i, x_j) = \exp(-\gamma \, \| \mathrm{x}_i - x_j \|^2), \; \gamma > 0 \tag{18}$$

Here g represents kernel variable. The benefit of using kernel functions is that it can operate in the initial input parameters wherein solutions of classification problems are weighted sum of kernels tested at support vectors. [19]

The precision of SVM classification can be improved by accurately setting the parameter in the kernels. The two variables to be examined in the SVM model with the RBF kernel are C as well as gamma (γ). The single training example may attain low values that denotes 'far' and high values that denotes 'close', but on other hand the gamma parameter can instantly define the distance. The C parameter provides the exchange of training examples misclassification against decision surface simplicity. A smooth decision surface is possible with lower C values while with higher C values accurate classification of training examples is attempted. Using the variation of gamma and C parameters, several tests were conducted to analyze the performance of SVM.

SVM optimized PSOCSS (SVM-PSO): The PSO-For classification, the PSO-SVM system initially tries to optimize the precision of SVM classifier. This is done, by identifying the subset of most useful features as well evaluating the most suitable values to regularize kernel variables for SVM model. This is achieved by using PSO based optimized model. PSO-SVM protocol uses the combination of 2 machine learning techniques through optimization of the variables of SVM utilizing PSO. [20]

The PSO is initiated using n-arbitrarily chosen particles as well as through iteration identifies optimum particle. Every particle is m-dimension vector as well as denotes a potential solution. For every candidate solution, a SVM classifier is developed to assess its performance via cross validation technique. Using PSO protocol, the potential subsets can be selected leading to suitable prediction accuracy. The method generally utilizes fit particles for contributing to subsequent generations of n-potential particles. Hence it could be observed on an average that every consecutive population of potential particles fit better compared to its predecessors. The procedure is continued till the performance of SVM converges. PSO can be employed to identify the optimum features subset through ascertaining the suitable attribute grouping as they fly in the problem space from the computed data sets.

The course of action for the suggested PSO-SVM technique is given below:

Step 1: PSO is initialized with population size, inertia weights as well as generation lacking improvement.

Step 2: The fitness of all particles are computed.

Step 3: The fitness values are compared and the local and global best particles are determined.

Step 4: The velocity as well as location of each particle is updated until the values of fitness functions converge.

Step 5: Once the values converge, the global best particle in the swarm is given to SVM classifier for training.

Step 6: SVM classifier is trained.
PSO-SVM utilizes the merit of minimal structural risk of SVM as well as rapid global optimization capability of PSOA.

Similar to any other evolutionary algorithm, the protocol of optimization by particulate swarm when applied is subjective to features like stopping criterion, the particle structure along with objective function.

Stopping criterion: The stopping criterion may be the iterative counting appended to the pre-condition, value of function objectives reached or motion of particle near 0.

Structure of particle: Particle " I " would have a vector that represents 2 values; firstly the coefficient of regularization " C " as well as secondly the variable of core RBF "sigma “likes the position xij= (xi1, xi2).

The objective function: The main use of the function objectifies is to minimise generalized errors.

SVM optimized CSS (SVM-CSS): The CSS is a meta-heuristic algorithm built in recent times. Using this various kind of structures can be optimally designed. The basis of CSS protocol is derived from the governing laws of physics. CSS consists of Charged Particles (CPs) or some agents that can influence one and other with respect to Coulomb as well as Gauss’s laws from electrostatics. The optimizing procedure in CSS protocol would proceed by assessing the influence of the resultant force on every CP. Consequently, the agents are shifted to new locations with respect to the Newtonian motion law. The following movement of CP guides the method towards optimal solution. [21]

The CSS [22] has its basis on electrostatic as well as Newtonian laws. The Coulomb as well as Gauss laws give the value of electric field at a point within as well as away from a charged insulating solid sphere, correspondingly, as (19):

$$E_{ij} = \begin{cases} \dfrac{k_e q_i}{a^3} r_{ij} & if \;\; r_{ij} < a \\ \dfrac{k_e q_i}{r_{ij}^2} & if \;\; r_{ij} \geq a \end{cases} \tag{19}$$

Here k_e represents a invariant called as Coulomb constant; r_{ij} signifies the separation of the center of sphere as well as the chosen point; q_i represents the magnitude of charge. The radius of charged sphere is given by ‘a’. From the theory of super-position, the resultant electric force because of N charged spheres is equivalent to (20):

$$F_j = k_e q_j \sum_{i=1}^{N} \left(\frac{q_i}{a^3} r_{ij}.i_1 + \frac{q_i}{r_{ij}^2}.i_2 \right) \frac{r_i - r_j}{\| r_i - r_j \|} \quad \left\langle \begin{array}{l} i_1 = 1, i_2 = 0 \; if \;\; r_{ij} < a \\ i_1 = 0, i_2 = 1 \; if \;\; r_{ij} \geq a \end{array} \right. \tag{20}$$

By the rule, in the initial cycle, particles are away from one another and so the magnitude of the resulting force functioning on particles is inversely proportional to square of the separation between particles. So, there is high power of exploration under this condition, since more searches are performed during initial cycles. It is essential to improve the exploitation of the protocol as well as to reduce exploration steadily. Once several searches are done, the particles are accumulated in a tiny space and so the resulting force is proportional to separation distances of particles. Hence, the variable ‘a’ differentiates the global as well as the local search stage.

The Newtonian mechanics has the following (21):

$$\Delta r = r_{new} - r_{old}$$
$$v = \frac{r_{new} - r_{old}}{\Delta t}$$
$$a = \frac{v_{new} - v_{old}}{\Delta t} \tag{21}$$

Where r_{old} and r_{new} represents the initial as well as final position of a particle correspondingly; v is particle velocity while a represents particle acceleration. When the aforementioned equations are combined and Newton’s second law is used, it is possible to obtain the displacement of objects as function of time (22):

$$r_{new} = \frac{1}{2}\frac{F}{m}\Delta t^2 + v_{old}.\Delta t + r_{old} \quad (22)$$

Based on the above electrostatic as well as Newtonian mechanics laws, the idea of CSS optimization technique can be given as below [23]:

Step 1 Initialization: A set of particles with arbitrary positions is initialized. The initial velocity of particles are assumed to be 0. Every particle has charge of magnitude (q) specified by taking into account the quality of solution (fitness value in one objective optimization problems). The separation distance r_{ij} among 2 charged particles i and j is specified as Euclidean distance among them in search space.

Step 2 Search: The attractive force vector for every particle is evaluated with respect to equation (20). In the equation F_j is the resulting force that affects jth particle. Once the resulting force functioning on every particle is computed, every particle are shifted to its novel position while the velocities are revised as (23 & 24):

$$x_{j,new} = r_{j1}.k_a.\frac{F_j}{m_j}.\Delta t^2 + r_{j2}.k_v.v_{j,old}.\Delta t + x_{j,old} \quad (23)$$

$$v_{j,new} = \frac{x_{j,new} - x_{j,old}}{\Delta t} \quad (24)$$

Here r_{j1} and r_{j2} refer to 2 arbitrary numbers uniformly distributed within (0, 1). The mass of jth particle is represented by m_j and is measured equivalent to qj as in the main protocol. Here Δt denotes timestep with value set to 1. Likewise k_a denotes acceleration coefficient; k_v signifies the velocity coefficient for controlling the influence of earlier velocities. The coefficients may be regarded as fixed or changing through the search procedure.

Results and Discussion

In this section, the 3000 positive, 2000 negative and 1500 neutral values are used, SVM RBF Kernel (C, Gamma) function is used. The SVM RBF Kernel (1, 0.1), SVM RBF Kernel (10, 0.1), SVM RBK Kernel (100, 0.1), SVM RBF Kernel (1, 0.01), SVM RBF Kernel (10, 0.01), SVM RBF Kernel (100, 0.01), SVM-PSO and SVM-CSS techniques are evaluated. The figure 1 and 2 shows that the classification accuracy and number of iterations. Table 2 to 4 shows that the precision, recall and F1 score.

From the figure 1, it can be observed that the SVM-CSS has higher classification accuracy by 10.95% for SVM RBF Kernel (1, 0.1), by 9.21% for SVM RBF Kernel (10, 0.1), by 7.5% for SVM RBK Kernel (100, 0.1), by 12.71% for SVM RBF Kernel (1, 0.01), by 6.41% for SVM RBF Kernel (10, 0.01), by 4.34% for SVM RBF Kernel (100, 0.01) and by 1.82% for SVM-PSO.

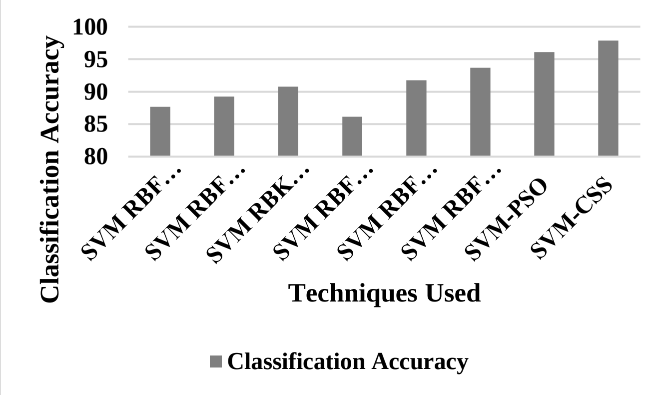

Figure 1: Classification Accuracy

From the table 2, it can be observed that the SVM-CSS has higher precision-positive by 7.26% for SVM RBF Kernel (1, 0.1), by 5.79% for SVM RBF Kernel (10, 0.1), by 3.95% for SVM RBK Kernel (100, 0.1), by 5.28% for SVM RBF Kernel (1, 0.01), by 1.23% for SVM RBF Kernel (10, 0.01), by 1.14% for SVM RBF Kernel (100, 0.01) and by 0.71% for SVM-PSO.

From the table 2, it can be observed that the SVM-CSS has higher precision-negative by 11.26% for SVM RBF Kernel (1, 0.1), by 5.44% for SVM RBF Kernel (10, 0.1), by 6.5% for SVM RBK Kernel (100, 0.1), by 15.43% for SVM RBF Kernel (1, 0.01), by 4.66% for SVM RBF Kernel (10, 0.01), by 3.84% for SVM RBF Kernel (100, 0.01) and by 2.79% for SVM-PSO.

From the table 2, it can be observed that the SVM-CSS has higher precision-neutral by 16.67% for SVM RBF Kernel (1, 0.1), by 19.96% for SVM RBF Kernel (10, 0.1), by 15.48% for SVM RBK Kernel (100, 0.1), by 22.22% for SVM RBF Kernel (1, 0.01), by 17.44% for SVM RBF Kernel (10, 0.01), by 10.75% for SVM RBF Kernel (100, 0.01) and by 2.7% for SVM-PSO.

From the table 2, it can be observed that the SVM-CSS has higher average precision by 11.63% for SVM RBF Kernel (1, 0.1), by 10.16% for SVM RBF Kernel (10, 0.1), by 8.5% for SVM RBK Kernel (100, 0.1), by 14.03% for SVM RBF Kernel (1, 0.01), by 7.51% for SVM RBF Kernel (10, 0.01), by 5.14% for SVM RBF Kernel (100, 0.01) and by 2.05% for SVM-PSO.

From the table 3, it can be observed that the SVM-CSS has higher recall-positive by 16.00% for SVM RBF Kernel (1, 0.1), by 8.33% for SVM RBF Kernel (10, 0.1), by 6.5% for SVM RBK Kernel (100, 0.1), by 14.03% for SVM RBF Kernel (1, 0.01), by 9.08% for SVM RBF Kernel (10, 0.01), by 5.78% for SVM RBF Kernel (100, 0.01) and by 1.88% for SVM-PSO.

From the table 3, it can be observed that the SVM-CSS has higher recall-negative by 7.13% for SVM RBF Kernel (1, 0.1), by 11.32% for SVM RBF Kernel (10, 0.1), by 8.51% for SVM RBK Kernel (100, 0.1), by 11.32% for SVM RBF Kernel (1, 0.01), by 4.16% for SVM RBF Kernel (10, 0.01), by 3.1% for SVM RBF Kernel (100, 0.01) and by 1.8% for SVM-PSO.

From the table 3, it can be observed that the SVM-CSS has higher recall-neutral by 6.33% for SVM RBF Kernel (1, 0.1), by 8.17% for SVM RBF Kernel (10, 0.1), by 8.17% for SVM RBK Kernel (100, 0.1), by 11.93% for SVM RBF Kernel (1, 0.01), by 4.18% for SVM RBF Kernel (10, 0.01), by 3.11% for SVM RBF Kernel (100, 0.01) and by 1.72% for SVM-PSO.

From the table 3, it can be observed that the SVM-CSS has higher average recall by 9.73% for SVM RBF Kernel (1, 0.1), by 9.26% for SVM RBF Kernel (10, 0.1), by 7.72% for SVM RBK Kernel (100, 0.1), by 12.42% for SVM RBF Kernel (1, 0.01), by 5.78% for SVM RBF Kernel (10, 0.01), by 3.99% for SVM RBF Kernel (100, 0.01) and by 1.8% for SVM-PSO

Table 2
Precision

	SVM RBF Kernel (1,0.1)	SVM RBF Kernel (10,0.1)	SVM RBK Kernel (100,0.1)	SVM RBF Kernel (1,0.01)	SVM RBF Kernel (10,0.01)	SVM RBF Kernel (100,0.01)	SVM-PSO	SVM-CSS
Precision Positive	0.9174	0.931	0.9483	0.9358	0.9745	0.9754	0.9796	0.9866
Precision Negative	0.869	0.9211	0.9114	0.8333	0.9284	0.936	0.9459	0.9727
Precision Neutral	0.8209	0.7941	0.8308	0.7761	0.8145	0.8712	0.9443	0.9702
Average Precision	0.8691	0.882067	0.896833	0.8484	0.9058	0.927533	0.9566	0.9765

Table 3
Recall

	SVM RBF Kernel (1,0.1)	SVM RBF Kernel (10,0.1)	SVM RBK Kernel (100,0.1)	SVM RBF Kernel (1,0.01)	SVM RBF Kernel (10,0.01)	SVM RBF Kernel (100,0.01)	SVM-PSO	SVM-CSS
Recall Positive	0.8333	0.9	0.9167	0.85	0.8933	0.9233	0.96	0.9783
Recall Negative	0.9125	0.875	0.9	0.875	0.94	0.95	0.9625	0.98
Recall Neutral	0.9167	0.9	0.9	0.8667	0.9367	0.9467	0.96	0.9767
Average Recall	0.8875	0.891667	0.905567	0.8639	0.923333	0.94	0.960833	0.978333

Table 4
F1 Score

	SVM RBF Kernel (1,0.1)	SVM RBF Kernel (10,0.1)	SVM RBK Kernel (100,0.1)	SVM RBF Kernel (1,0.01)	SVM RBF Kernel (10,0.01)	SVM RBF Kernel (100,0.01)	SVM-PSO	SVM-CSS
F1 Score - Positive	0.8733	0.9152	0.9322	0.8908	0.9321	0.9486	0.9697	0.9824
F1 Score - Negative	0.8902	0.8975	0.9057	0.8536	0.9342	0.9429	0.9541	0.9763
F1 Score - Neutral	0.8662	0.8437	0.864	0.8189	0.8713	0.9074	0.9521	0.9734
Average F1 Score	0.876567	0.885467	0.900633	0.854433	0.912533	0.932967	0.958633	0.977367

From the table 4, it can be observed that the SVM-CSS has higher F1 score-positive by 11.75% for SVM RBF Kernel (1, 0.1), by 7.08% for SVM RBF Kernel (10, 0.1), by 5.24% for SVM RBK Kernel (100, 0.1), by 9.78% for SVM RBF Kernel (1, 0.01), by 5.25% for SVM RBF Kernel (10, 0.01), by 3.5% for SVM RBF Kernel (100, 0.01) and by 1.3% for SVM-PSO.

From the table 4, it can be observed that the SVM-CSS has higher F1 score-negative by 9.22% for SVM RBF Kernel (1, 0.1), by 8.41% for SVM RBF Kernel (10, 0.1), by 7.5% for SVM RBK Kernel (100, 0.1), by 13.41% for SVM RBF Kernel (1, 0.01), by 4.4% for SVM RBF Kernel (10, 0.01), by 3.48% for SVM RBF Kernel (100, 0.01) and by 2.3% for SVM-PSO.

From the table 4, it can be observed that the SVM-CSS has higher F1 score-neutral by 11.65% for SVM RBF Kernel (1, 0.1), by 14.27% for SVM RBF Kernel (10, 0.1), by 11.9% for SVM RBK Kernel (100, 0.1), by 17.24% for SVM RBF Kernel (1, 0.01), by 11.06% for SVM RBF Kernel (10, 0.01), by 7.01% for SVM RBF Kernel (100, 0.01) and by 2.21% for SVM-PSO.

From the table 4, it can be observed that the SVM-CSS has higher average F1 score by 10.87% for SVM RBF Kernel (1, 0.1), by 9.86% for SVM RBF Kernel (10, 0.1), by 8.17% for SVM RBK Kernel (100, 0.1), by 13.42% for SVM RBF Kernel (1, 0.01), by 6.86% for SVM RBF Kernel (10, 0.01), by 4.64% for SVM RBF Kernel (100, 0.01) and by 1.93% for SVM-PSO.

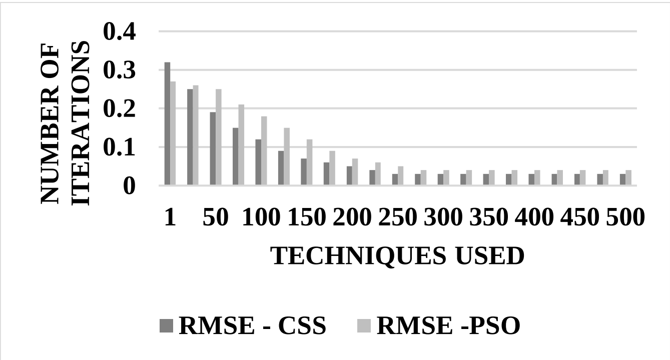

Figure 2: Number of Iterations

From the figure 2, it can be observed that the SVM-CSS has lower average number of iteration by 23.28% for SVM-PSO.

Conclusion

In this work, a few principles from physics as well as mechanics are taken as the basis for the suggested modern optimization method and is it termed as CSS. It uses the governing Coulomb law from electrostatics as well as the Newtonian laws of mechanics. CSS is a multi-agent method such that every agent is a CP. CSS may be used to optimize fields; particularly applicable for non-smooth or non-convex domains. It does not require gradient data as well as the continuity of the search space. The outcomes reveal that the SVM-CSS has greater classification accuracy by 10.95% for SVM RBF Kernel (1,0.1), by 9.21% for SVM RBF Kernel (10, 0.1), by 7.5% for SVM RBK Kernel (100,0.1), by 12.71% for SVM RBF Kernel (1,0.01), by 6.41% for SVM RBF Kernel (10,0.01), by 4.34% for SVM RBF Kernel (100, 0.01) and by 1.82% for SVM-PSO.

References

1. Hanafi H.F. and Samsudin K., Mobile learning environment system (MLES): the case of Android-based learning application on undergraduates' learning, *International Journal of Advanced Computer Science and Applications(IJACSA)*, **3(3)**, DOI: 10.14569/IJACSA.2012.030311 **(2012)**

2. Sarrab M., Elgamel L. and Aldabbas H., Mobile learning (m-learning) and educational environments, *International journal of distributed and parallel systems*, **3(4)**, 31 **(2012)**

3. Asabere N.Y., Towards a perspective of Information and Communication Technology (ICT) in education: Migrating from electronic learning (E-learning) to mobile learning (M-learning), *International Journal of Information and Communication Technology Research*, **2(8)**, 646-649 **(2012)**

4. Behera S.K., E- and M-Learning: A comparative study, *International Journal on New Trends in Education and Their Implications*, **4(3)**, 65-78 **(2013)**

5. Ozuorcun N.C. and Tabak F., Is M-learning versus E-learning or are they supporting each other?, *Procedia-Social and Behavioral Sciences*, **46**, 299-305 **(2012)**

6. Korucu A.T. and Alkan A., Differences between m-learning (mobile learning) and e-learning, basic terminology and usage of m-learning in education, *Procedia-Social and Behavioral Sciences*, **15**, 1925-1930 **(2011)**

7. Cheon J., Lee S., Crooks S.M. and Song J., An investigation of mobile learning readiness in higher education based on the theory of planned behaviour, *Computers & Education*, **59(3)**, 1054-1064 **(2012)**

8. Chen H.R. and Tseng H.F., Factors that influence acceptance of web-based e-learning systems for the in-service education of junior high school teachers in Taiwan, *Evaluation and program planning*, **35(3)**, 398-406 **(2012)**

9. Kaneda Y., Zhao Q. and Watarai K., An study on the effect of learning parameters for inducing compact SVM, In 2012 IEEE International Conference on Systems, Man and Cybernetics (SMC) 1367-1372 **(2012)**

10. Couellan N., Jan S., Jorquera T. and Georgé J.P., Self-adaptive Support Vector Machine: A multi-agent optimization perspective, *Expert Systems with Applications*, **42(9)**, 4284-4298 **(2015)**

11. Mahalakshmi K., Prabhakar D.R. and Balakrishnan D.V., Kernel Optimization for Improved Nonfunctional Requirements Classification, *Journal of Theoretical and Applied Information Technology*, **60(1)**, 64-72 **(2016)**

12. Mahalakshmi K. and Prabhakar R., Hybrid Optimization of SVM for Improved Non-Functional Requirements Classification,

International Journal of Applied Engineering Research, **10(20) (2015)**

13. Kannan S. and Gurusamy V., Preprocessing Techniques for Text Mining **(2014)**

14. Mahalakshmi K., Prabhakar R. and Balakrishnan V., Optimizing Support Vector Machine for Classifying Non Functional Requirements, *Research Journal of Applied Sciences, Engineering and Technology*, **7(17)**, 3643-3648 **(2014)**

15. Chen L.S. and Chang C.W., A new term weighting method by introducing class information for sentiment classification of textual data, In proceedings of International Multi Conference of Engineers and Computer Scientists **(2011)**

16. Durgesh K.S. and Lekha B., Data classification using support vector machine, *Journal of Theoretical and Applied Information Technology*, **12(1)**, 1-7 **(2010)**

17. Aburomman A.A. and Reaz M.B.I., A novel SVM-kNN-PSO ensemble method for intrusion detection system, *Applied Soft Computing*, **38**, 360-372 **(2016)**

18. Ben-Hur A. and Weston J., A user’s guide to support vector machines, *Methods Mol Biol*, doi: 10.1007/978-1-60327-241-4_1 **(2010)**

19. Ivanciuc O., Applications of support vector machines in chemistry, *Reviews in computational chemistry*, **23**, 291 **(2007)**

20. Ardjani F., Sadouni K. and Benyettou M., Optimization of SVM multiclass by particle swarm (PSO-SVM), In 2010 2nd International Workshop on Database Technology and Applications, IEEE, 1-4 **(2010)**

21. Kaveh A. and Talatahari S., Hybrid charged system search and particle swarm optimization for engineering design problems, *Engineering Computations*, **28(4)**, 423-440 **(2011)**

22. Kaveh A. and Talatahari S., A novel heuristic optimization method: charged system search, *Acta Mechanica*, **213(3-4)**, 267-289 **(2010)**

23. Kaveh A. and Laknejadi K., A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization, *Expert Systems with Applications*, **38(12)**, 15475-15488 **(2011)**.