An Improved Charged System Search for Feature Selection of IDS

Namdeo Kumar Ashish[1], Dileep Singh[2]

[1]School of Engineering and Technology, Jagran Lakecity University, Bhopal, India,
Email: ashish1namdeo@gmail.com
[2]School of Engineering and Technology, Jagran Lakecity University, Bhopal, India,
Email: dileep.singh@jlu.edu.in

Received: 10.04.2024 Revised : 12.05.2024 Accepted: 22.05.2024

ABSTRACT
Charged system search (CSS) algorithm which is inspired by the coulomb's law and laws of motion has been proved to be competitive with existing evolutionary algorithms, such as Particle Swarm Optimization (PSO) algorithm. The shortcomings of existing CSS, algorithm is small convergence precision and readily captured in a local optimum value at the next evolution stage. The paper introduces an enhanced Charged System Search algorithm with Levy Flight (CSSLF), by looking at the information of the, best solutions into the exploration strategy for the feature subset selection. The Support Vector Machine has been applied as a classifier for assessment of the features selected. The experimental results on KDD-NSL dataset reveals that CSSLF discover better solutions than CSS in terms of larger detection rate, nominal false alarm rate and enhanced accuracy than the existing approaches.

Keywords: Particle Swarm Optimization,Charged System Search, Levy Flight, Feature Selection, Intrusion Detection.

1. INTRODUCTION

The problem of intrusion detection is widely recognized as a pattern learning problem, an Intrusion Detection System (IDS) permits normal from abnormal network traffic on a host [42]. In addition, it is of significance to further analyze abnormal behaviour in order to initiate acceptable counter-efforts. IDS can be designed in numerous ways [1], [2]. A design of this kind generally comprises a depiction algorithm (for selecting features) and a classification algorithm (for mapping the feature vector to elements of a specific set of values, for instance normal or abnormal etc.). Some IDS, like the ones suggested in [1], likewise consist of the feature selection algorithm, which regulates the features to be handled by the depiction algorithm. Even if the feature-selection algorithm is not built in the model directly, it is invariably expected that such an algorithm is driven before the intrusion detection process [39].
The class, of the feature selection algorithm is one of the most prominent elements that affect the effectiveness of an IDS. The objective of the algorithm is to decide the most relevant features of the traffic, whose control would provide reliable detection of unusual behaviour. Since the accuracy of the classifier depends on the number of features, it is crucial to reduce the element of the set of selected features, without falling probable indicators of abnormal behaviour. Obviously, determining a suitable set of features is not a simple task. The substantial part of the work is still performed manually and the feature selection algorithm depends extremely on professional judgment. Automatic feature selection for intrusion detection is thus essential [42]. For automatic feature selection use of optimization algorithms as wrapper methods were suggested. In wrapper method, set of features is assigned a score based on accuracy and its various combinations are employed to explore for optimal feature subset. A comparative survey of Cuttle Fish Optimization (CFA), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) had been made to prove their strength in feature selection [3]. Several metaheuristic optimizations such as Gravitational Search Algorithm (GSA) [5] and Ant Colony Optimization (ACO) [4] were proposed which turned out to be suitable in case of optimization problems. The ACO algorithm was employed in featureselection task for IDS to decide optimal feature subset [6].
CSS is another multiagent meta-heuristic optimization algorithm which was developed by Kaveh and Talatahari [7] The CSS was successfully employed to many structural optimization problems by Kaveh and Ahmadi [18],Kaveh and Nasrollahi [19] and Kaveh and Talatahari [13], [14], [15], [16], [17]. Results from the above analysis indicate that CSS has a stable convergence rate and there is a reasonable balance between exploration and exploitation. However, the search approaches in these multi-agent algorithms

have limited randomization, dynamic local search and selection of the best solutions, typically uses Gaussian distributionor uniform distribution[46].
A recent study [10] indicates that fruit flies, examine their landscape adopting a sequence of short flight paths interspersed by an abrupt 90°shift, contributing to a Levy Flight style intermittent scale free exploration pattern. Investigations on human behaviour like Ju/'hoansi hunter gatherer foraging sequences further indicate the common character of Levy Flights. Even light can be described to Levy Flights [8]. Later, such behaviour has been employed to optimization, and preliminary results indicate its potential effectiveness [43], [9], [10], [11], [12]. In this paper the authors have proposed improved Charged System Search algorithm with Levy Flight (CSSLF) algorithm for feature selection in order to obtain relevant features and subset assessment is done by the classifier. The classifier being employed is Support Vector Machine (SVM) [36]. We will first frame the Charged System Search algorithm; then lay out systematically the CSS Levy Flight and finally perform the comparison about the performance of these algorithms. The CSSLF optimization seems more promising than Particle Swarm Optimization in the sense that CSSLF converges more quickly and deals with global optimization more easily.

2. Charged System Search

The Charged System Search (CSS) is heuristic optimization evolutionary algorithm which is based on the Coulomb's law, Gauss law of electrostatics and the Newtonian mechanical law. A concise description of the CSS reflected as a multi-agent approach, where each agent is a Charged Particle (CP), which is considered as a charged sphere having radius a, carrying a uniform volume charge density of magnitude [49].

$$q_i = \frac{\mathrm{fit}(i) - \mathrm{fit\_worst}}{\mathrm{fit\_best} - \mathrm{fit\_worst}}, i = 1, \dots, N \qquad (1)$$

Wherefit_best and fit_worstare the best and the worst fitness of among the charge particles;fit(i)represents the fitness of the agenti,and N is the total number of CPs [38]. The initial positions of CPs are determined randomly in the search space employing

$$x_{i,j}^{(0)} = x_{i,min} + rand_{i,j} \cdot (x_{i,max} - x_{i,min}), i = 1,2, \dots, N \qquad (2)$$

Where$x_{i,j}^{(0)}$ determines the initial value of the ithvariable for thejth CP; $x_{i,min}$ and $x_{i,max}$ are the minimum and the maximum admissible values for theith variable;$rand_{i,j}$is a random number in the interval (0,1) [48]. The initial velocities of charged particles are taken as

$$v_{i,j}^{(0)} = 0, i = 1,2, \dots, N \qquad (3)$$

CPs can impose electrical forces on the each others, and its intensity is proportional to the separation distance between the CPs, when the CP is located inside the sphere and is inversely proportional to the square of the distance between them, when it's outside the sphere. The nature of the forces can be attractive or repelling as determined by a force parameter [37].

$$ar_{i,j} = \begin{cases} +1, & k_t < rand_{i,j} \\ -1, & k_t \geq rand_{i,j} \end{cases} \qquad (4)$$

Where$ar_{i,j}$measures the nature of the force, in which +1 implies the attractive force and −1 stands for the repulsing force, and $k_t$is a parameter to manage the effect of the kind of the force. In general the attractive force gathers the agents in a part of search space and the repulsing force strives to dissipate the agents.The resultant force [37] is designated as

$$F_j = \sum_{i,i\neq j} \left(\frac{q_i}{a^3} r_{i,j} \cdot i_1 + \frac{q_i}{{r_{i,j}}^2} \cdot i_2 \right) ar_{i,j} \cdot p_{i,j} (X_i - X_j) \begin{cases} j = 1,2, \dots, N \\ i_1 = 1, i_2 = 0; \ r_{i,j} < a \\ i_1 = 0, i_2 = 1; \ r_{i,j} \geq a \end{cases} \qquad (5)$$

Where$F_j$is the resultant force acting on thejth CP; $r_{i,j}$is the distance between two charged particles which is defined [38] as

$$r_{i,j} = \frac{\|X_i - X_j\|}{\|(X_i + X_j)/2 - X_{best}\| + \epsilon} \qquad (6)$$

HereX_i and $X_j$are the positions of theith and jth CPs, respectively;X_{best} is the position achieved, and∈is a small positive number to avoid singularity. The$p_{i,j}$determines the probability [38] of moving a CP toward the other CP as

$$p_{i,j} = \begin{cases} 1, & if \ \frac{\mathrm{fit}(i) - \mathrm{fit\_best}}{\mathrm{fit}(j) - \mathrm{fit}(i)} > rand \ \vee \mathrm{fit}(j) > fit(i) \\ 0, & else \end{cases} \qquad (7)$$

The resultant forces and the laws of the motion decide the new location of the CPs. At this stage, each CP proceeds towards its new position under the response of the resultant forces and its preceding velocity [37] as

$$X_{j,new} = rand_{j1} \cdot k_a \cdot \frac{F_j}{m_j} \cdot \Delta t^2 + rand_{j2} \cdot k_v \cdot V_{j,old} \cdot \Delta t + X_{j,old} \tag{8}$$

$$V_{j,new} = \frac{X_{j,new} - X_{j,old}}{\Delta t} \tag{9}$$

Where$k_a$is the acceleration coefficient;$k_v$is the velocity coefficient o regulate the change of the precedingvelocity; $\Delta t$is the change in time instance; $rand_{j1}$ and $rand_{j2}$are two random numbers uniformly distributed in the range (0,1).If each CP jumps out of the search area, its position is rectified usingthe harmony search-based handling method [7].In addition, to preserve the best results, a memory, known as the Charged Memory [37], is used.

3.Levy Flight

Levy Flight is defined [20], [21], [22], [23], [24], [25], [26], as non-Gaussian random generation of numbers with Levy stable distribution which consists of two steps: the choice of a random direction and the step-lengths which has a probability distribution that obeys the chosen Levy distribution. Random walks are drawn from Levy stable distribution with a power-law frequency, stated as$L(s) \sim |s|^{-1-\beta}$ where $0<\beta<2$ is an index [40]. For random walk, the progression length S can be computedby Mantegna's algorithm as

$$S = \frac{u}{|v|^{1/\beta}} \tag{10}$$

where u and v are calculated from normal distributions. That is

$$u \sim N(0, \sigma_u^2), \qquad v \sim N(0, \sigma_v^2), \tag{11}$$

Where

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left[\frac{1+\beta}{2}\right]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta} \tag{12}$$

Then the step size is calculated by

$$stepsize = 0.01 \times S \tag{13}$$

Here the factor 0.01 originates from the phenomenon that L/100 should the normal step size of walks where L is the ordinary length scale; otherwise, Levy Flights may become overly aggressive, which creates new solutions to drop outside the domain (and therefore wasting computation time) [40].

4. The Proposed CSSLF Algorithm for Feature Selection

The authors have proposed a CSSLF algorithm to select important feature subset in order to increase the classification accuracy. The proposed feature selection algorithm is first applied on the train dataset to extract the best possible feature subset and then SVM classifier is used to evaluate the performance. Figure 1 gives the flow of the proposed work.

4.1 CSSLF Based Feature Selection Method

In this section, we investigate a new approach to feature selection using CSSLF. Where CSS's performance is improved by designing different technique for updating position, using Levy Flight method. Similar to original CSS Charged Particles (CP) in CSSLF algorithm are initialized as randomly selected feature subset. The value in the CP vector is either 1 or 0 depending on whether the given feature is selected or not in the subset. The CP vector represents position of the particle in the search space. The set of CP's are selected as the initial agents in the search space are represented by their positions.

The CP's are ranked according to their fitness value and best CP's are stored in the Charge Memory (CM). The particles in CM now attract every other particle in the search space towards itself. The force on each CP due to particles in CM is calculated using Equation (5). The radius a and mass m for each particle is assumed to be unity.

Since the position vectors of each CP are binary values, hamming distance $h(.)$is used to calculate the distance of separation between the particles and is given as in Equation (14). Here X_{best} is the position of best CP and $\in$ is a small value which prevents singularity.

$$r_{i,j} = \frac{h(X_i . X_j)}{h((X_i \hat{} X_j) . X_{best}) + \in} \tag{14}$$

When the random probability is greater than or equal to 0.5, the particle's position is updated by Equation (8) and (9) respectively. If the random value is less than 0.5 [35], then particle's position is updated as given in the Equation. (15). By employing Levy Flight method in updating the particle's position, particle takes long jump towards its global best and thereby enhancing the diversity of the swarm and facilitating the algorithm to perform global exploration throughout the search space.

In Levy Flight method β parameter plays a significant role in distribution. By employing distinct values for, β the random distribution is adjusted accordingly. In our study, we adopt constant value for β (i.e., 1.5)[35].Loss of diversity is thus avoided by using random phenomenon of Levy Flight while updating the position. As the performance of CSS algorithm is enhanced by incorporating the advantages of random walk into the CSS, it improves particle's positions after going through high exploration and exploitation of the search space during each iteration.

$$X_{j,new} = rand_{j1} \cdot k_a \cdot \left(\frac{F_j}{m_j}\right) \cdot \Delta t^2 + rand_{j2} \cdot k_v \cdot V_{j,old} \cdot \Delta t + \omega * Levy_{walk\,(X_{j,old})} \quad (15)$$

Where

$$k_a = 1 + \left(\frac{current_{iteration}}{total_{iteration}}\right), \quad (16)$$

$$k_v = 1 - \left(\frac{current_{ite\,ration}}{total_{iteration}}\right), \quad (17)$$

$$\omega = 0.1 + 0.8 \times \left(1 - \frac{current_{iteration}}{total_{iteration}}\right), \quad (18)$$

and

$$Levy\_walk(X_{j,old}) = X_{j,old} + step \otimes random\left(size(X_i)\right) \quad (19)$$

Where

$$step = stepsize * X_{j,old} \quad (20)$$

and step size is the value acquired from Equation (13).

⊗Represents element wise multiplication.

Fitness value is calculated, after updating particle's position and velocity, if the fitness value for the new particle is better than its previous fitness value, then update value. Otherwise the value is not updated. Repeat the same process until the given number of function evaluations is reached or the optimal feature subset is attained. The pseudocode of the CSSLF algorithm is given in Table1 and flowchart is given in Figure1.

Figure1 depicts the flow of the general feature selection process. The model employs two stages: In the first stage, CSSLF feature selection finds an optimal subset of all attributes and removes low fitness attributes [45]. The fitness plays a major role in identifying the high priority attributes that are crucial for classification. In the second stage the quality of the derived feature subset is evaluated by classification algorithm, such as Support Vector Machine (SVM). However, in this approach, it is necessary to add a memory to the CSS for each particle in order to save the best position of the CP up to the current iteration. With these modifications, the performance of the CSS improves in such a way that by defining the dynamical variance, it is expected to raise the convergence rate, and by the use of the Levy Flight in the CSS, a better search will be performed.

The stages of implementation of the CSSLF are similar to that of the CSS and only these modifications are applied, with the above definitions, the steps of optimization by the CSSLF are shown in the flowchart.

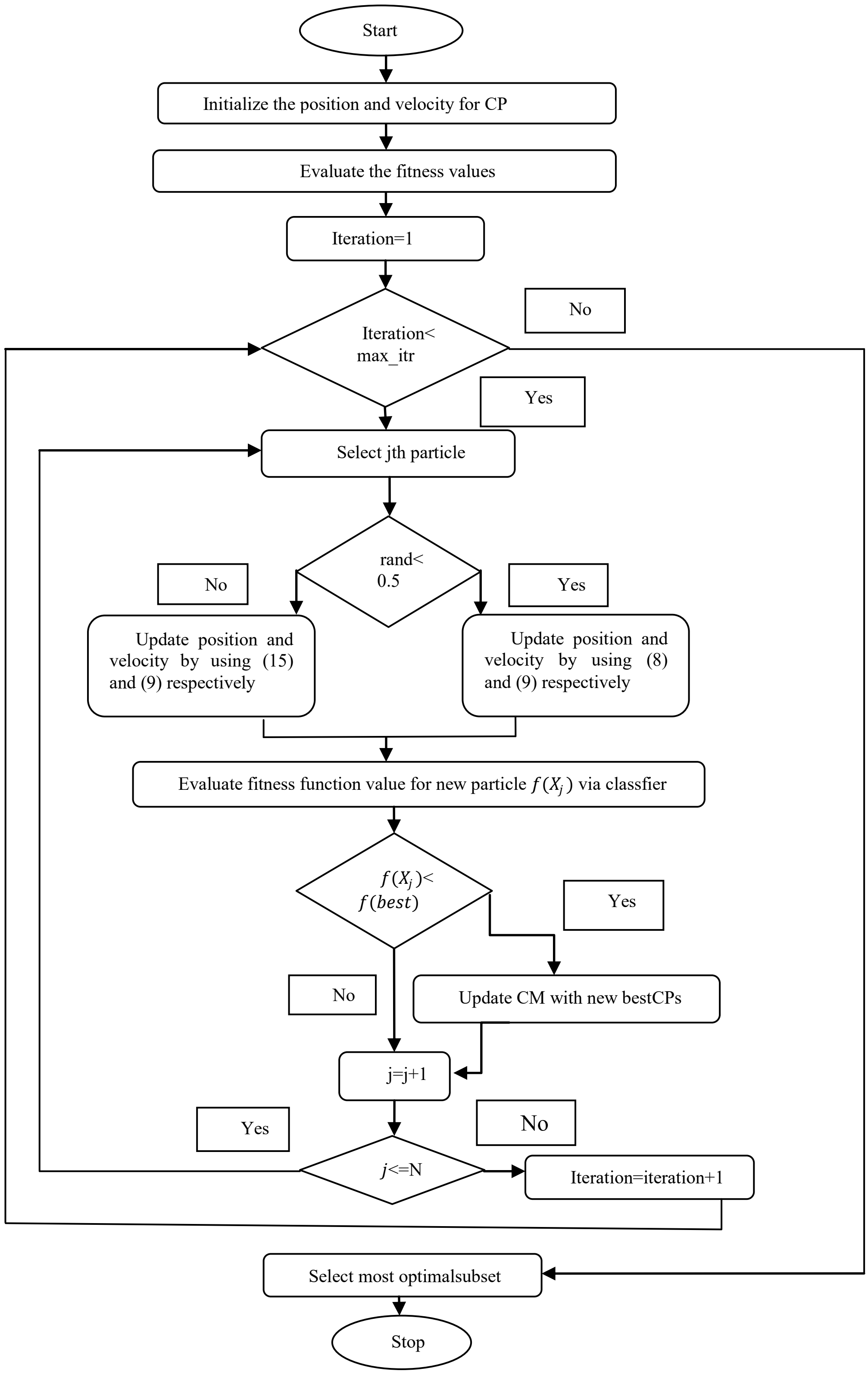

Fig.1.CSSLF based method for feature selection

Table 1. The pseudo code of CSSLF

Algorithm: CSSLF for feature selection.

Initialize the particles with random positions for N CP's
Initialize the velocityfor CP
Evaluate fitness function value
iteration=1
while (convergence) **do**
    $k_a = 1 + \left(\frac{\text{curren t}_{\text{iteration}}}{\text{tota l}_{\text{iteration}}}\right)$,
    $k_v = 1 - \left(\frac{\text{curren t}_{\text{iteration}}}{\text{to ta l}_{\text{iteration}}}\right)$,

$$\omega = 0.1 + 0.8 \times \left(1 - \frac{\text{current}_{\text{iteration}}}{\text{total}_{\text{iteration}}}\right)$$

        for j=1 to N**do**
            if rand ( ) < 0.5 **then**
                Update the velocity and position of the particle by Equation (9) and (15) respectively
            else
                Update the velocity and position of the particle by Equation(9) and (8) respectively
            end if
            Position values are brought to the boundary value when its values are moved out of the boundary
            Evaluate the fitness function for new particle X_j
            if fit(i) $< fitbest$**then**
                select k best CPs via Classifier's Evaluation and assign them to CM
                calculate magnitude and direction of force on each CP due to particles in CM
            end if
        end for
    Record the best solution
    iteration=iteration+1
end while
Output the best solution

The CSSLF algorithm is effective towards solving the tradeoff between exploration and exploitation. It also solves the problem of premature convergence and trapping in local optima. These advantages add to efficient feature subset selection for the Intrusion Detection System.

4.2 Preprocessing

The pre-processing deals with the conversions applied to your data [47] before the feature reduction and classification algorithms [36] are applied on it. This phase has two steps:

- Transforms each nominal feature with n possible values into n numerical features [47].
- Normalizationis a method used to standardize the features so that they'll have the properties of a standard normal distribution with μ=0 and σ=1, where μ is the mean (average) and σ is the standard deviation from the mean [47].

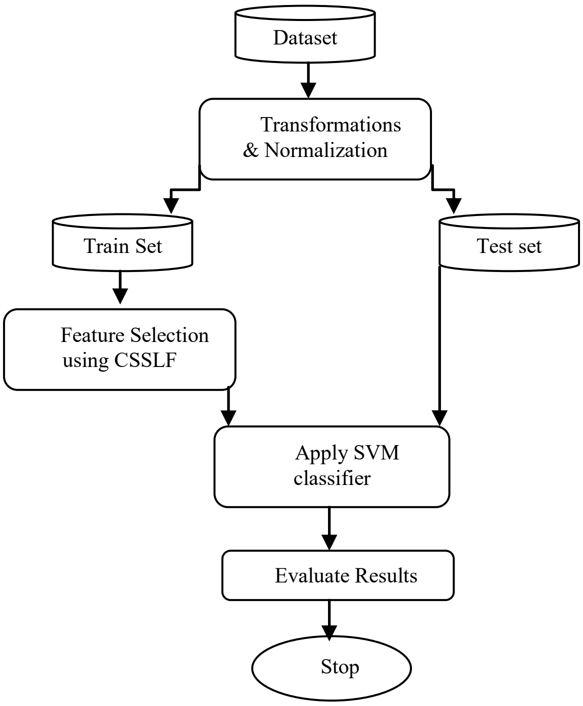

Fig 2. Intrusion detection model using CSSLF for feature selection

Figure 2 shows the stages of SVM classifier for predicting the class label of the network traffic. The dataset is divided into two sets: Training set and Testing set [44]. In training phase, the feature matrix is fed in to the classifier model to identify the class label. The testing phase obtains the learning rules from training phase to identify the pattern of the unknown traffic.

4.3 Classification with Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised binary classification algorithm. The algorithm constructs the optimal hyper plane separating the two classes [27]. In order to extend from linear to nonlinear classification the kernel trick is used [28], where kernel functions nonlinearly map input data into high-dimensional feature spaces in a computationally-efficient manner [41].

For classification problems with multiple classes, two approaches are commonly used for binary SVMs, one-against-one and one-against-all [29]. Both techniques lead to similar results in terms of classification accuracy, but the former, which was the one usually, requires shorter training time, although incurring a higher number of binary decompositions [41].

For the current experiments, we used the LibSVM library [30], [41].Since SVM performed well among the classical intrusion detection algorithms [31], we also use SVM to detect intrusions on the same dataset for comparison.

5. Experiments and Results

In this section, the detailed evaluation of the proposedCSSLF algorithm is presented. For comparison, two other algorithms are used [35].They are Standard PSO [32], which is one of the state of the art of evolutionary algorithms and anotherbeing trapped in local optima is CSS[7]. The goal of this paper is to improve the performance of CSS algorithm for choosing the optimal subset of features.

5.1 Dataset Used

In this experiment, NSL-KDD dataset is used for evaluating the proposed feature selection approach. This dataset is a modified version of KDDCUP'99 which is the mostly widely used standard dataset for the evaluation of Intrusion Detection Systems [33].This dataset has a large number of network connections with 41 features for each of them which means it is a good example for large scale dataset to test on. Each connection sample belongs to one of five main labeled classes (Normal, DOS, Probe, R2L, and U2R). NSL-KDD dataset includes training dataset with 23 attack types,and 1,25,973distinctconnection recordswhere as test dataset with additional 14 attack types and 22,554 distinct connection records.

5.2 Performance Metrics

The performance of our method is measured by employing the following metrics. These values are true negatives (TN), true positives (TP), false positives (FP) and false negatives (FN) where TN specifies the abnormal behavior that is correctly predicted, TP indicates the normal performance that is identified as correct FP denotes the normal behavior falsely considered as abnormal, and FN specifies the abnormal performance that is misclassified as normal.

We consider the false alarm rate (FAR), accuracy (ACC),detection rate (DR),and precision (PC)which are mostly used in literature to estimate the performance of intrusion detection. They can be determined from the confusion matrix, as given in Table 2.

Table 2. Confusion Matrix

Class	Predicted (normal)	Predicted (attack)
Actual (normal)	TP	FN
Actual (attack)	FP	TN

Precision is the ratio of the number of TP records classified to the total number of predicted records.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{21}$$

A False Alarm Rate in IDSs is an attack alarm that is raised incorrectly divided by total network sessions. Detection Rate is the ratio of total numbers of attack intrusions detected to the total number of attacks currently available in the data set [50].

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \tag{22}$$

$$\text{Detection Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{23}$$

$$\text{False alarm rate} = \frac{\text{FN}}{\text{TN} + \text{FP}} \tag{24}$$

An IDS system must have a higher attack detection rate along with a low false alarm rate for the better performance.

5.3 Experimental Setup

The experiments for this work were performed on a personal computer with Intel Core i5 processor of 2.5GHz speed and a RAM of 8.00GB. The programming language used to implement the algorithms was Python. The implementation was executed on Jupyter Notebook for Python.The dataset NSL-KDD [34] was obtained from the Canadian Institute for Cyber security datasets repository.

5.4 Result Analysis

The proposed anomaly based IDS proved to be reliable and successful in distinguishing anomalous user behaviors from the normal ones. The model’s efficiency has been evaluated on the NSL-KDD dataset. The results for the proposed IDS using Charged System Search algorithm with Levy Flight (CSSLF) algorithm for feature selection is done in two parts.

In the first part of the IDS model, the CSSLF algorithm uses cross validation accuracy for fitness value calculation for each Charged Particle. A 10 fold cross validation is performed on the train set for each feature subset and the feature subset with highest accuracy at the end of the search is selected as best feature subset. As the search proceeds towards optimal solution, CPs in search space with different number of selected features attain different accuracy. From the fitness optimization curve that is illustrated in Figure 3, it can be certainly observed that CSS and PSO may either fall into a local optimum quickly, or have a comparatively slow evolution momentum, while CSSLF is made with good global search ability and fast convergence [44]. The CSSLF algorithm finds the optimal feature subset by initializing random CPs and then guiding bad CPs to the good CPs, searching for optima through the solution space.

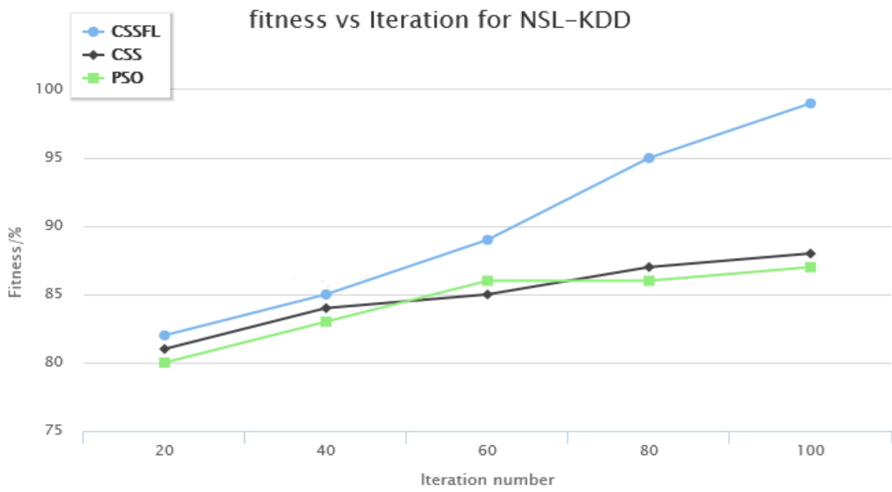

Fig 3. Comparison between Fitness optimization of CSSFL, CSS, and PSO.

The results shows that the classifier used in the experiment perform better for the selected features. The Charged Particles in CSSLF algorithm strikes balance between the exploration and exploitation during the search as well as it is successful in finding the optimal feature subset which is responsible for increase in accuracy of the classifier. The IDS model achieves high values for precision and detection rate.

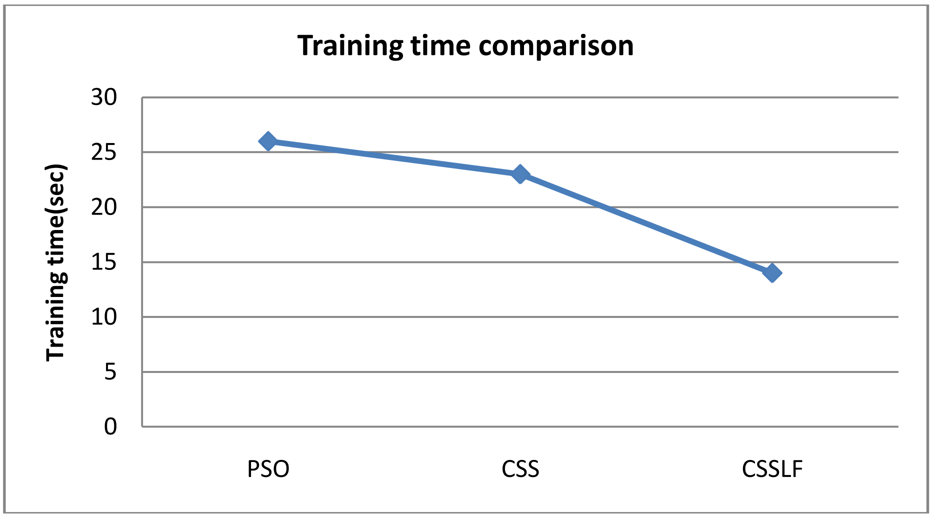

Fig 4. The training time comparison of different approaches

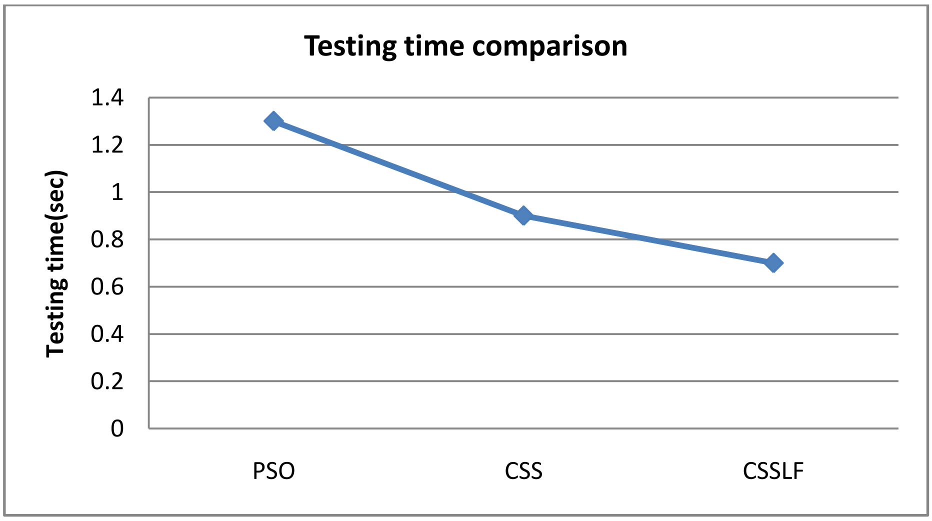

Fig 5. The testing time comparison of different approaches

Figure 4 and Figure 5 show a comparison of time taken by each method for training and testing respectively [36].It is observed from the figure that the computational time of proposed CSSLF has been reduced as compared to PSO and CSS. This reduction in time was possible due to the use of Levy Flight distribution, as PSO and CSS gets stuck in the local minima while CSSLF algorithm escapes the local minima and obtains the better result than those [35].Thus, one of the advantages of using Levy Flightis that it considerably lowers the cost of computation in terms of time taken [36].

Table 3. Comparison of feature selection techniques based on number of features

Methods	Features selected
PSO	10
CSS	9
CSSLF (proposed)	8

Table 3 demonstrates the comparison of number of attributes selected by different feature selection techniques. The obtained feature subsets outperform a raw feature set. Further, CSSLF based feature selection indicates best performance with minimum number of features.

Table 4. Confusion Matrix obtained after feature selection

Methods	TP	TN	FP	FN
PSO	8367	12003	1344	830
CSS	12856	11508	889	3
CSSLF (proposed)	13011	11803	133	2

Table 5. Performance comparison of feature selection methods

Methods	Detection rate	False alarm rate	Accuracy	Precision
PSO	0.9353	0.1384	0.9035	0.8616
CSS	0.9997	0.0647	0.9647	0.9353
CSSLF (proposed)	0.9998	0.0099	0.9946	0.9762

Table 5 shows the performance metrics of each class such as attack and normal. The false alarm rate is very less and the Detection rate is very high which is an important criteria as achieved by our proposed method. Precision and Accuracy performance metrics are close to 1.Table 4 shows the Confusion Matrix of each class obtained after feature selection. The rows in the matrix represent values; and entries along the columns specify methods.

Table 6. Features Comparison Metrics via Different Classifier and Algorithm

Feature Selection Algorithm	Classifier	Detection rate	False alarm rate	Accuracy	Precision
Principal Component Analysis	Logistic Regression	0.7744	0.2872	0.8162	0.7128
	Ada Boost	0.7814	0.0179	0.7919	0.9821
	Decision Tree	0.7961	0.1364	0.8489	0.8636
	Naive Bayes	0.8329	0.0945	0.7980	0.9055
	Random Forest	0.8379	0.1997	0.8336	0.8003
Particle Swarm Optimization	Logistic Regression	0.8080	0.1550	0.8604	0.8450
	Ada Boost	0.8027	0.0236	0.8134	0.9764
	Decision Tree	0.8103	0.1334	0.8376	0.8666
	Naive Bayes	0.8547	0.0813	0.8214	0.9187
	Random Forest	0.7971	0.0532	0.8746	0.9468
Charged System Search	Logistic Regression	0.9784	0.1492	0.9277	0.8508
	Ada Boost	0.8346	0.0191	0.8400	0.9809
	Decision Tree	0.8911	0.1258	0.8925	0.8742
	Naive Bayes	0.8811	0.0617	0.8540	0.9383
	Random Forest	0.8105	0.0080	0.8955	0.9920
Charged System Search algorithm with Levy Flight	Logistic Regression	0.9907	0.0533	0.9742	0.9467
	Ada Boost	0.9154	0.0174	0.9112	0.9826
	Decision Tree	0.8401	0.1249	0.9023	0.8751
	Naive Bayes	0.8998	0.0250	0.8851	0.9750
	Random Forest	0.9578	0.0507	0.9515	0.9493

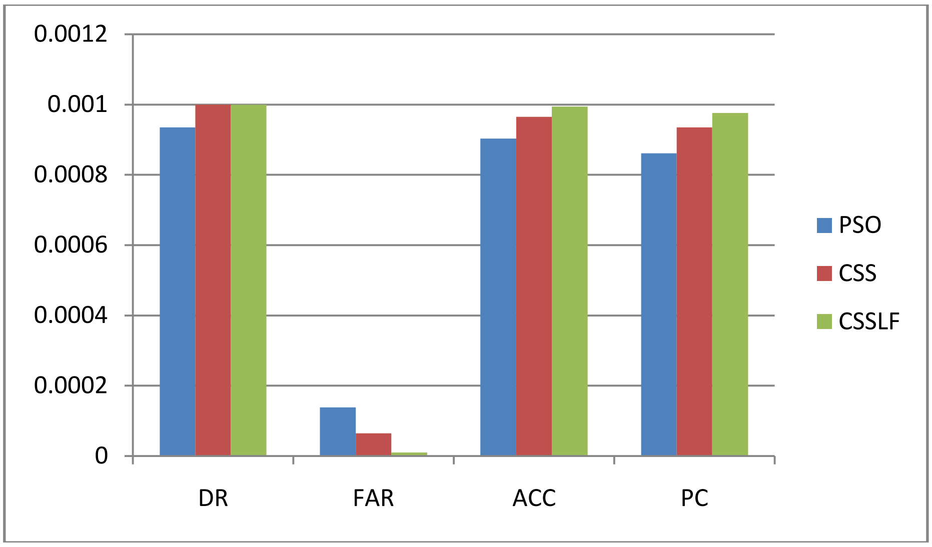

Fig 6. Statistical analysis of different methods

The IDS model achieves high values for precision and detection rate. The best result (shown in Figure 6) from a proposed approach was achieved by CSS with Levy Flight. It gives around 99.98% of detection rate with reasonable 0.99% of false alarm rate. The time taken by the model to train by CSSLF is greatly reduced to 14 seconds which is less than that of other approaches. Thus, it can be said that the proposed model is capable in achieving high detection accuracy with minimum is classification rate in a very less time. This quality makes the model computationally effective [36].
The CSSLF algorithm is an efficient metaheuristic optimization technique for feature selection. Thus, this feature selection technique can be applied with better anomaly detection models in order to achieve higher accuracy and lower false alarm rate. The proposed Intrusion Detection System proves to be reliable and effective in order to detect intrusion.

6. CONCLUSION

In this paper, a Charged System Search algorithm with Levy Flight(CSSLF) was proposed to select the best feature subset. The improved CSS is basically the standard CSS combined with Levy Flight search strategy. Then the CSSLF is used for feature selection from IDS. The experiment results of IDS indicated that the CSSLF is feasible to optimize the feature subset and improving the IDS accuracy, detection rate and false alarm rate, which has a 9.11%, 6.45% and12.85% respectively improvement than PSO whereas with reference to CSS it is 2.99%, 0.01% and 5.48%respectively enhancement in performance. The proposed method suggested effective ways to handle exploration-exploitation tradeoff. This method restricts the agents escaping the search space. The convergence of the CSSLF algorithm is fast and is attributed to better exploration and exploitation. The proposed method reduced the number of features in IDS from 41 to 8 features which help to improve the accuracy as well as makes it faster and efficient in detecting anomalous user behavior. In their future work the authors look forward to incorporate improvements such as adding variable weights and radius for the Charge Particles which will enhance effectiveness of method for feature selection.

Compliance with ethical standards

Conflict of interest

The authors declare that they have no conflict of interest.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

REFERENCES

[1] Gu, Guofei, Prahlad Fogla, David Dagon, Wenke Lee, and Boris Skoric. "Towards an information-theoretic framework for analyzing intrusion detection systems." In ESORICS, pp. 527-546. 2006.
[2] Di Crescenzo, Giovanni, Abhrajit Ghosh, and Rajesh Talpade. "Towards a theory of intrusion detection." In ESORICS, pp. 267-286. 2005.

[3] Anusha, K., and E. Sathiyamoorthy. "Comparative study for feature selection algorithms in intrusion detection system." Automatic Control and Computer Sciences 50, no. 1 (2016): 1-9.
[4] Dorigo, Marco, Vittorio Maniezzo, and Alberto Colorni. "Ant system: optimization by a colony of cooperating agents." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 26, no. 1 (1996): 29-41.
[5] Rashedi, Esmat, Hossein Nezamabadi-Pour, and Saeid Saryazdi. "GSA: a gravitational search algorithm." Information sciences 179, no. 13 (2009): 2232-2248.
[6] Aghdam, Mehdi Hosseinzadeh, and Peyman Kabiri. "Feature Selection for Intrusion Detection System Using Ant Colony Optimization." IJ Network Security 18, no. 3 (2016): 420-432.
[7] Kaveh, A., and S. Talatahari. "A novel heuristic optimization method: charged system search." Acta Mechanica 213, no. 3 (2010): 267-289.
[8] Barthelemy, Pierre, Jacopo Bertolotti, and Diederik S. Wiersma. "A Lévy flight for light." Nature 453, no. 7194 (2008): 495-498.
[9] Pavlyukevich, Ilya. "Lévy flights, non-local search and simulated annealing." Journal of Computational Physics 226, no. 2 (2007): 1830-1844.
[10] Reynolds, Andy M., and Mark A. Frye. "Free-flight odor tracking in Drosophila is consistent with an optimal intermittent scale-free search." PloS one 2, no. 4 (2007): e354.
[11] Shlesinger, Michael F., George M. Zaslavsky, and Uriel Frisch. "Lévy flights and related topics in physics." Lecture notes in physics 450 (1995): 52.
[12] Shlesinger, Michael F. "Mathematical physics: Search research." Nature 443, no. 7109 (2006): 281-282.
[13] Kaveh, A., and S. Talatahari. "A charged system search with a fly to boundary method for discrete optimum design of truss structures." Asian J Civil Eng 11, no. 3 (2010): 277-93.
[14] Kaveh, A., and S. Talatahari. "Charged system search for optimum grillage system design using the LRFD-AISC code." Journal of Constructional Steel Research 66, no. 6 (2010): 767-771.
[15] Kaveh, Ali, and Siamak Talatahari. "Optimal design of skeletal structures via the charged system search algorithm." Structural and Multidisciplinary Optimization 41, no. 6 (2010): 893-911.
[16] Kaveh, Ali, and Siamak Talatahari. "An enhanced charged system search for configuration optimization using the concept of fields of forces." Structural and Multidisciplinary Optimization 43, no. 3 (2011): 339-351.
[17] Kaveh, Ali, and Siamak Talatahari. "Geometry and topology optimization of geodesic domes using charged system search." Structural and Multidisciplinary Optimization 43, no. 2 (2011): 215-229.
[18] Kaveh, A., and B. Ahmadi. "Simultaneous analysis, design and optimization of structures using the force method and supervised charged system search algorithm." Scientia Iranica 20, no. 1 (2013): 65-76.
[19] Kaveh, A. A., and A. Nasrollahi. "Charged system search and particle swarm optimization hybridized for optimal design of engineering structures." Scientia Iranica. Transaction A, Civil Engineering 21, no. 2 (2014): 295.
[20] Haklı, Hüseyin, and Harun Uğuz. "A novel particle swarm optimization algorithm with Levy flight." Applied Soft Computing 23 (2014): 333-345.
[21] Chechkin, Alexei V., Ralf Metzler, Joseph Klafter, and Vsevolod Yu Gonchar. "Introduction to the theory of Lévy flights." Anomalous transport: Foundations and applications 1 (2008): 129-162.
[22] Yang, Xin-She, and Suash Deb. "Cuckoo search via Lévy flights." In Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, pp. 210-214. IEEE, 2009.
[23] Tuba, Milan, Milos Subotic, and Nadezda Stanarevic. "Modified cuckoo search algorithm for unconstrained optimization problems." In Proceedings of the 5th European conference on European computing conference, pp. 263-268. World Scientific and Engineering Academy and Society (WSEAS), 2011.
[24] Bacanin, Nebojsa. "An object-oriented software implementation of a novel cuckoo search algorithm." In Proc. of the 5th European Conference on European Computing Conference (ECC'11), pp. 245-250. 2011.
[25] Yang, Xin-She. "Firefly algorithm, Levy flights and global optimization." Research and development in intelligent systems XXVI (2010): 209-218.
[26] Xie, Jian, Yongquan Zhou, and Huan Chen. "A novel bat algorithm based on differential operator and Lévy flights trajectory." Computational intelligence and neuroscience 2013 (2013).
[27] Vapnik, Vladimir N. "An overview of statistical learning theory." IEEE transactions on neural networks 10, no. 5 (1999): 988-999.
[28] Scholkopf, Bernhard, and Alexander J. Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2001.

[29] Witten, Ian H., Eibe Frank, Mark A. Hall, and Christopher J. Pal. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.
[30] Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." ACM transactions on intelligent systems and technology (TIST) 2, no. 3 (2011): 27.
[31] Sung, Andrew H., and Srinivas Mukkamala. "Identifying important features for intrusion detection using support vector machines and neural networks." In Applications and the Internet, 2003. Proceedings. 2003 Symposium on, pp. 209-216. IEEE, 2003.
[32] Ahmad, Iftikhar. "Feature selection using particle swarm optimization in intrusion detection." International Journal of Distributed Sensor Networks 11, no. 10 (2015): 806954.
[33] Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set." In Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on, pp. 1-6. IEEE, 2009.
[34] NSL-KDD dataset (2009). URL: http://www.unb.ca/cic/research/datasets/nsl.html.
[35] R. Jensi and G.W. Jiji, “An enhanced particle swarm optimization with levy flight for global optimization,” Applied Soft Computing, vol.43, pp.248-261, 2016.
[36] N. Acharya and S. Singh. “An IWD-based feature selection method for intrusion detection system.” Soft Computing, pp.1-10, 2017.
[37] S. Talatahari, R. Sheikholeslami, M. Shadfaran and M. Pourbaba, “Optimum design of gravity retaining walls using charged system search algorithm,”Mathematical Problems in Engineering, 2012.
[38] S.Talatahari, R. Sheikholeslami, B. Farahmand Azar, and H. Daneshpajouh. “Optimal parameter estimation for Muskingum model using a CSS-PSO method.”Advances in Mechanical Engineering, vol.5, pp.480-954, 2013.
[39] H. Moradi Koupaie, S. Ibrahim and J. Hosseinkhani, “Outlier detection in stream data by machine learning and feature selection methods,” 2014.
[40] S Xian, J Zhang, Y Xiao and J Pang, “A novel fuzzy time series forecasting method based on the improved artificial fish swarm optimization algorithm,” Soft Computing, pp.1-11, 2017.
[41] H.K. Turesson, S. Ribeiro, D.R. Pereira, J.P. Papa and V.H.C. de Albuquerque, “Machine learning algorithms for automatic classification of marmoset vocalizations,”PloS one, vol. 11, no. 9, p.e0163041, 2016.
[42] H. Nguyen, S. Petrović and K. Franke, “A comparison of feature-selection methods for intrusion detection,” Computer network security, pp.242-255, 2010.
[43] X.S. Yang and S. Deb, “Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization,”Nature Inspired Cooperative Strategies for Optimization (NICSO),2010, pp.101-111.
[44] Q. Zhang, L. Fang, S. Su and Y. Lv, “Parameters Optimization of SVM Based on Improved FOA and Its Application in Fault Diagnosis,”JSW, vol. 10, no. 11, pp.1301-1309, 2015.
[45] I.S. Thaseen and C.A. Kumar, “Intrusion detection model using fusion of PCA and optimized SVM,” In Contemporary Computing and Informatics (IC3I), International Conference, 2014, pp. 879-884.
[46] X.S. Yang, “Firefly algorithm, Levy flights and global optimization,” Research and development in intelligent systems XXVI, pp.209-218, 2010.
[47] S. Danish (2016). PracticalGuide on Data Preprocessing in Python using Scikit Learn [Online]. Available:https://www.analyticsvidhya.com/blog/2016/07/practical-guide-data-preprocessing-python-scikit-learn.
[48] Y.K. Wu, G.T. Ye and M. Shaaban, “Analysis of impact of integration of large pv generation capacity and optimization of pv capacity: Case studies in taiwan,” IEEE Transactions on Industry Applications, vol. 52, no.6, pp.4535-4548, 2016.
[49] A. Kaveh and S. Talatahari,“Geometry and topology optimization of geodesic domes using charged system search,” Structural and Multidisciplinary Optimization, vol.43, no. 2, pp.215-229, 2011.
[50] N. Cleetus and K.A. Dhanya, “Genetic algorithm with different feature selection method for intrusion detection,” In Computational Systems and Communications (ICCSC), 2014, pp. 220-225.