Hybrid GSA-CSSA based Emotional ELMAN Neural Network Classifier for Software Defect Prediction

Kumudha. P  
Assistant Professor (SG), CSE  
Coimbatore Institute of Technology, Anna University, Coimbatore-641014, Tamil Nadu, India.

Venkatesan. R  
Professor, CSE  
PSG College of Technology, Anna University, Coimbatore-641004, Tamil Nadu, India.

Abstract
Fundamentally, during the development of software modules it is of prime concern they should be defect free. The ultimate aim of software developers is to provide a software package so that there exists effective and efficient allocation of resources. It is required in the growing scenario of software developments-prediction mechanisms which identifies the software with defects and enables the developers for effective and efficient resource allocation along with the required quality being met. The identification of software defects at the earliest makes the product to be delivered with highest accuracy to the end users. For the past several years, lot numbers of works have been carried out in the area of software defect prediction and numerous computational approaches are developed for effective software defect prediction. In this paper, a novel approach based on proposed emotional ELMAN neural network (EENN) classifier and hybrid gravitational search algorithm (GSA)-charged system search algorithm (CSSA) is developed to perform effective software defect prediction. The weights of the proposed EENN classifier is tuned employing the proposed hybrid GSA-CSSA approach and the accuracy is computed. Five public datasets are employed in this paper to validate the proposed hybrid GSA-CSSA based emotional ELMAN NN classifier model. The simulated results prove the effectiveness of the proposed software predictor model over the other methods available in the previous literature for the considered same datasets. The public datasets are obtained from the NASA database repository.

Keywords: Software development-Software defects-Classifier models-Neural Network-Emotional ELMAN Neural Network-Gravitational Search Algorithm-Charged System Search Algorithm

Introduction
In the growing and developed software engineering field, software quality is of high importance. Also, it should be noted that constructing software modules are highly expensive basically. As a result, to enhance the efficient resource allocation, efficiency and the general throughput of quality assurance and testing, it is highly important to predict software defects and assure that the developed software modules are defect free. The prediction of software defects at an early stage will make the company professionals to deliver a quality product to the end customers, as the cost incurred for the development play a major role.

In the growing scenario, software miners have developed numerous modules for defect prediction replacing the earlier existing conventional statistical approaches. Basically, the classifier model or the predictor model is designed in a manner to output two cases: one is the defect prone software and the other is the defect free software. To start the process of classifier models, the already known instances will be given as input to the classifier models. On the effective training of the proposed model, this will be tested with that of the available unknown instances and then the prediction will be carried out. Several studies have been carried out in the early literature based on the performance metrics [1] and design requirements [2].

Considering the literature studies, numerous works have been carried out in the area of software defect prediction employing different techniques which includes-decision trees [3, 4], boosting models [5], fuzzy predictor models [6-10], support vector machine [11, 12], neural predictors [13-19], bayes models [20], rule based approaches [21, 22] and so on. A review on the various open issues in software prediction has been presented in [23]. With respect to the discussed literature study on the existing software defect predictor models, it is noted that several predictors has not considered misclassification cost of the defective and non-defective modules and these play a major role for real world problems. Thus, this paper focuses on developing emotional ELMAN neural classifiers along with hybrid GSA-CSSA to minimize the total misclassification costs. The contribution made in this paper includes the development of emotional ELMAN neural network models tuned by the proposed hybrid gravitational search algorithm and charged system search algorithm and effectively performing the software defect prediction.
The remaining section of the paper is organized as follows: The background of the applicability of ELMAN neural network, GSA and CSSA are presented in Section 2. The datasets collected from the NASA repository and used in this research paper is given in Section 3. Section 4 details the proposed hybrid GSA-CSSA based emotional ELMAN neural network predictor model. The results of the proposed model with the metrics used and analysis are detailed in Section 5 and the conclusions on the contributions made are presented in Section 6.

Background on the applicability of ELMAN neural network, GSA and CSSA techniques

This section presents the applicability of ELMAN neural network, gravitational search algorithm and charged system search algorithm as modeled for various engineering and science applications. The literature background on applicability of these techniques is as given below:

A surface electromyography (sEMG) based eyebrow emotional expression recognition method has been proposed with six features of the sEMG time domain being extracted and used as input vectors to an emotion recognition model based on an Elman neural network (ENN). This method is used to recognize facial emotional expression effectively in human-computer interaction [24]. A hybrid method for obtaining the optimal weight set and architecture of a recurrent neural emotion classifier based on gravitational search algorithm (GSA) and its binary version (BGSA) has been done [25].

The region for emotional expression with electrical stimulation and injected neural tracers to label incoming and outgoing projections is identified in the work [26]. The result shows that the emotional responses classically attributed to the insular cortex are endowed with an en-active component intrinsic to each social and emotional behavior. The feasibility of using back-propagation (BP) neural networks and electroencephalograms (EEGs) to recognize the emotional reactions induced by sound stimuli in the dimensions of pleasure and arousal has been carried out [27] and is compared for performance recognition. An optimum training and forecasting approach for natural gas consumption forecasting and estimation in cognitive and noisy environments by an integrated approach has been done [28]. The approach is capable of modeling sharp drops or jumps in consumption with appropriate cognitive and emotional signals.

The resonant frequency response of electrically thin and thick rectangular patch antenna using emotional back propagation network by providing inputs, which are length of the patch, width of the patch, height of the patch and relative permittivity is analyzed and used in reducing a significant bottleneck like low computational speed in the electromagnetic method of micro-strip analysis, with good accuracy [29].

A new ELMAN neural network trained to predict the future value of the residual time series has been implemented and applied to Mackey-Glass and Lorenz equations which produce chaotic time series, and to a real life chaotic time series, Sunspot time series has been analyzed [30]. The developed methodology predicts the chaotic time series more effectively and accurately in comparison with that of earlier prediction methods. A neural network model named Global control ELAMN to combine the utterance-based features and segment-based features together and apply with a best segment length for emotion recognition for each emotional state has been done earlier [31].

A hybridized population-based Cuckoo search-Gravitational search algorithm (CS-GSA) for optimization has been developed by [32]. Twenty three different kinds of standard test functions are considered here to compare the performance of our hybridized algorithm with both the CS and the GSA methods.

A powerful heuristic optimization algorithm called Bat algorithm (BA) for adaptive filtering in dual channel enhancement systems has been carried out [33] and evaluated for four objective measures. Assareh & Biglari [34] presented a proportional and integral (PI) controller tuned by a radial basis function (RBF) neural network and the optimal dataset to train this neural network is provided by the Gravitational Search Algorithm (GSA).

Fuzzy gain scheduling system with optimized rules by subtractive clustering algorithm for tuning the proportional-integral-derivative (PID) controller parameters based on error and error-difference in an online mode has been carried out [35]. An Elman-type recurrent neural network (RNN) was used for inverse identification of the PV system and for estimating the solar radiation intensity to determine the MPP voltage. A suitable tuning method for optimizing the damping controller parameters using a novel hybrid Genetic Algorithm-Gravitational Search Algorithm (hGA-GSA) has been developed [36].

An image segmentation method based on BP neural network optimized by an enhanced Gravitational Search Algorithm (GSA) has been done [37] and a cat chaotic mapping into the steps of population initialization and iterative stage of the original GSA to forms a new algorithm called CCMGSA. Kumar [38] developed an optimal location and capacity of Unified Power Flow Controller to improve the power system stability using hybrid technique composed of Ant Bee Colony-GSA algorithms.

Chandrasekar & Ponnusamy [39] discussed and applied population-based optimization techniques namely gravitational search algorithm (GSA) and hybrid particle swarm optimized gravitational search algorithm (PSO-GSA) to tune the proportional integral (PI) controller parameters of a conical tank system. Do [40] proposed a hybrid version of Gravitational Search Algorithm (GSA) and Back Propagation (BP) to make use of the advantage of both the GSA and BP algorithms. Modified hybrid Particle Swarm Optimization
(PSO) and GSA based on fuzzy logic (FL) has been developed [41] to control and improve the ability of search mechanism. A hybrid meta-heuristic optimization algorithm called the Particle Swarm Optimization and Gravitational Search Algorithm-Explore (PSOGSA-E) has been introduced to suppress the peak side-lobe level (PSL) by finding the best weight for each node [42].

Jiang [43] modeled a novel hybrid particle swarm optimization and gravitational search algorithm (HPSO-GSA) to minimize the error cost objective. Simulations are performed and the results show that the proposed algorithm has advantages over PSO, GSA and other PSO-based variants in terms of the convergence speed and the MSE levels. Hybrid gravitational search algorithm (HGSA) [44] has been developed to design the digital IIR filter. The modelled hybrid search techniques is applied effectively to solve the multi-parameter and multi-objective optimization problem of low-pass (LP), high-pass (HP), band-pass (BP), and band-stop (BS) digital IIR filter design. A nature inspired optimization algorithm to solve complex economic load dispatch problem using hybrid PSO-GSA algorithm has been developed by Dubey [45]. A hybrid meta-heuristic algorithm by combining both MCSS and particle swarm optimization (PSO) algorithms, which is called as MCSS-PSO has been developed for partitioning clustering problem [46].

Hybrid charged system search algorithm and particle swarm optimization algorithm for optimal design of single-layer barrel vault frames has been developed by Talatahari & Jahani [47]. Kaveh & Nasrollahi [48] presented a new Hybrid Charged System Search and Particle Swarm Optimization (HCSSPSO) for the optimal design of engineering structures. A hybrid meta-heuristic algorithm based on charged system search and particle swarm optimization has been developed by Kaveh et al [50] proposed a hybrid charged system search algorithm (CSS) and particle swarm optimization for identifying the parameters of the linear Muskingum model [49]. (PSO) to explore the minimum thickness design of laminated composite plates under in-plane loading where ply numbers and fiber orientations are considered as design variables.

A hybrid heuristic method using the harmony search (HS) and charged system search (CSS), called HS-CSS has also been modelled earlier to compute the minimum weight design of truss structures [51]. Further, a new hybrid meta-heuristic optimization algorithm based on the concepts of the charged system search (CSS) and the particle swarm optimization (PSO) algorithms has been developed for the design of structures [52]. Kaveh & Laknejadi proposed a new charged system search and particle swarm optimization method for multi-objective optimization problem [53]. A new hybrid algorithm by adding positive properties of the particle swarm optimization (PSO) algorithms to the charged system search (CSS) to solve constrained engineering optimization problems has been proposed by Kaveh & Talatahari [54].

From the above literature reviews, it can be noted that the considered ELMAN neural network, GSA and CSSA techniques has been employed for various applications. Thus in this paper, focus is given for developing ELMAN neural network with emotional quotients introduced and tuning the developed emotional ELMAN neural network employing the proposed hybrid GSA-CSSA approach. The cost-sensitive factors are also included in the developed model to enhance the rate of prediction of defects in software modules. The modelled hybrid GSA-CSSA based emotional ELMAN neural network technique is tested with the five public datasets from NASA promise repository and the simulated results prove the effectiveness of the proposed approach with that of the earlier available techniques from the literature.

**Description of NASA Promise repository datasets**

Datasets from the NASA Promise repository which are available in public is employed in this paper for validating the proposed hybrid GSA-CSSA based EENN. Primarily these datasets corresponds to the data of space craft modules, satellite flight control mechanisms and ground data storage management systems. There are several datasets available in the repository and the five datasets considered in this paper include-CM1, JM1, KC1, KC2 and PC1). The inputs will be the quality metrics and the output will be a defective or non-defective case. Fundamentally, these datasets rely on McCabe and Halstead feature extractors of the source code developed. The datasets were developed in C/ C++ language. The description of the datasets employed in this research paper is as given in Table 1.

**Table 1: Description of NASA promise repository datasets [55]**

<table>
<thead>
<tr>
<th>Name of the Dataset</th>
<th>Language employed</th>
<th>No. of instances</th>
<th>No. of attributes</th>
<th>Non-Defective Module</th>
<th>Defective Module</th>
<th>Defect rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM1 (NASA spacecraft instrument)</td>
<td>C</td>
<td>498</td>
<td>22</td>
<td>449</td>
<td>49</td>
<td>9.83%</td>
</tr>
<tr>
<td>JM1 (Real time predictive ground system)</td>
<td>C</td>
<td>10885</td>
<td>22</td>
<td>8779</td>
<td>2106</td>
<td>19.35%</td>
</tr>
<tr>
<td>KC1 (Storage management for receiving and processing ground data)</td>
<td>C++</td>
<td>2109</td>
<td>22</td>
<td>1783</td>
<td>326</td>
<td>15.45%</td>
</tr>
<tr>
<td>KC2 (Storage management for receiving and processing ground data)</td>
<td>C++</td>
<td>522</td>
<td>22</td>
<td>415</td>
<td>107</td>
<td>20.49%</td>
</tr>
<tr>
<td>PC1 (Flight software for earth orbiting satellite)</td>
<td>C</td>
<td>1109</td>
<td>22</td>
<td>1032</td>
<td>77</td>
<td>6.94%</td>
</tr>
</tbody>
</table>
In Table 1, it is noted that each of these five datasets possess 22 attributes (21 attributes belonging to the quality metrics and 1 attribute is the output attribute specifying whether defect or defect free). These 22 attribute information are as presented in Table 2. Feature sub-selection is employed and the attributes as given in Table 3 are employed as the input to the proposed predictor model instead of using all the 21 attributes. These attributes specified in Table 3 shows the prominence nature and hence are employed as inputs. This process reduces the computational complexity of the proposed hybrid GSA-CSA based EENN model and for effective comparison of the proposed approach, the same metrics [56] are used as inputs for the proposed model as well.

**Table 2: Attribute information of the NASA Promise repository datasets**

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>Attribute type</th>
<th>Quality metrics</th>
<th>Attribute Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>McCabe’s measure</td>
<td>loc</td>
<td>Line count of code</td>
</tr>
<tr>
<td>2</td>
<td>Basic Halstead measures</td>
<td>v(g)</td>
<td>Cyclomatic complexity</td>
</tr>
<tr>
<td>3</td>
<td>Basic Halstead measures</td>
<td>ev(g)</td>
<td>Essential complexity</td>
</tr>
<tr>
<td>4</td>
<td>Basic Halstead measures</td>
<td>iv(g)</td>
<td>Design complexity</td>
</tr>
<tr>
<td>5</td>
<td>Basic Halstead measures</td>
<td>loCode</td>
<td>Line count of code</td>
</tr>
<tr>
<td>6</td>
<td>Basic Halstead measures</td>
<td>loComment</td>
<td>Count of lines of comments</td>
</tr>
<tr>
<td>7</td>
<td>Basic Halstead measures</td>
<td>loBlank</td>
<td>Count of blank lines</td>
</tr>
<tr>
<td>8</td>
<td>Basic Halstead measures</td>
<td>loCodeAndComment</td>
<td>Count of code and comment lines</td>
</tr>
<tr>
<td>9</td>
<td>Derived Halstead measures</td>
<td>uniq_Op</td>
<td>Unique operators</td>
</tr>
<tr>
<td>10</td>
<td>Derived Halstead measures</td>
<td>uniq_Opnd</td>
<td>Unique operands</td>
</tr>
<tr>
<td>11</td>
<td>Derived Halstead measures</td>
<td>total_Op</td>
<td>Total operators</td>
</tr>
<tr>
<td>12</td>
<td>Derived Halstead measures</td>
<td>total_Opnd</td>
<td>Total operands</td>
</tr>
<tr>
<td>13</td>
<td>Derived Halstead measures</td>
<td>branchCount</td>
<td>Branch count of the flow graphs</td>
</tr>
<tr>
<td>14</td>
<td>Output Defect measure</td>
<td>n</td>
<td>Total operators + operands</td>
</tr>
<tr>
<td>15</td>
<td>Output Defect measure</td>
<td>v</td>
<td>Volume</td>
</tr>
<tr>
<td>16</td>
<td>Output Defect measure</td>
<td>l</td>
<td>Program length</td>
</tr>
<tr>
<td>17</td>
<td>Output Defect measure</td>
<td>d</td>
<td>Difficulty</td>
</tr>
<tr>
<td>18</td>
<td>Output Defect measure</td>
<td>i</td>
<td>Intelligence</td>
</tr>
<tr>
<td>19</td>
<td>Output Defect measure</td>
<td>e</td>
<td>Effort</td>
</tr>
<tr>
<td>20</td>
<td>Output Defect measure</td>
<td>b</td>
<td>Estimate of the effort</td>
</tr>
<tr>
<td>21</td>
<td>Output Defect measure</td>
<td>t</td>
<td>Time estimator</td>
</tr>
<tr>
<td>22</td>
<td>Output Defect measure</td>
<td>defects</td>
<td>(false, true)-module has/has not one or more reported defects</td>
</tr>
</tbody>
</table>

**Table 3: Input attributes to the proposed hybrid GSA-CSA-EENN model**

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>Name of the dataset</th>
<th>No. of Attributes selected</th>
<th>Name of the selected Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CM 1</td>
<td>7</td>
<td>loc, iv(g), i, IOComment, IOBlank, uniqOp, uniqOpnd</td>
</tr>
<tr>
<td>2</td>
<td>JM 1</td>
<td>8</td>
<td>loc, v(g), ev(g), iv(g), i, IOComment, IOBlank, IOCodeAndComment</td>
</tr>
<tr>
<td>3</td>
<td>KC 1</td>
<td>8</td>
<td>v, d, i, IOCode, IOComment, IOBlank, uniqOpnd, branchCount</td>
</tr>
<tr>
<td>4</td>
<td>KC 2</td>
<td>3</td>
<td>ev(g), b, uniqOpnd</td>
</tr>
<tr>
<td>5</td>
<td>PC 1</td>
<td>6</td>
<td>v(g), i, IOComment, IOCodeAndComment, IOBlank, uniqOpnd</td>
</tr>
</tbody>
</table>

**Proposed Hybrid GSA-CSA based Emotional ELMAN Neural Network**

ELMAN neural network is a recurrent neural network model wherein the information gets transferred to the hidden layer from the recurrent layer. This enables the increased accuracy and faster convergence of the network and that is the reason of choosing ELMAN neural network in this work. Also, in this paper attempt is taken to introduce emotional quotients into the ELMAN neural network, based on which the error gets reduced during the convergence of the network. For the proposed emotional ELMAN neural network, the existing randomly initialized weight components are tuned employing the proposed hybrid gravitational search algorithm and charged system search algorithm. This section presents the proposed hybrid GSA-CSA based emotional ELMAN neural network technique.

**Proposed Emotional ELMAN Neural Network (EENN) approach**

In the proposed EENN model, emotional factors are introduced at the time of training process of the neural network. Generally, emotions pertain to the feelings of an individual at a particular instant and their spontaneous reaction for that situation. In humans, there exist multiple emotional reactions-surprise, happiness, sorrow, fear, anxious, ambiguous, disgust, shock, anger and so on. The psychological insight of a particular human is depicted by these emotions. Similar to human biological neural systems, in this paper, emotions are introduced into the existing ELMAN neuronal model [57, 58] for handling complex data and as well to achieve better solutions. Thus the proposed ELMAN neural network architecture in this paper introduces two emotional factors for the convergence process-one is the anxiety coefficient and the other is the confidence coefficient.

**EENN approach with emotional factors**

In the proposed EENN approach, the emotional coefficients anxiety and confidence is introduced into the regular ELMAN neural network model along with the momentum factor and learning rate parameter for adjustment of weights in ELMAN neuronal model. The description of the emotional factors introduced into the proposed EENN approach is as follows:

i) The anxiety component and its proximity depend on the input patterns and the new patterns in turn will result in higher anxiety.

ii) This anxiety component is also based on the computed error factor of the basic ELMAN neural network. This error factor is employed by the ELMAN neural network for updating the measure of the learning process. When the error gets minimized, the anxiety coefficient also gets reduced.

iii) When the anxiety coefficient gets minimized then the confidence coefficient gets maximized.
iv) The developed EENN model consists of both the anxiety and the confidence coefficients during the learning and generalization process.

v) The introduction of these coefficients into the EENN model makes the training process faster and avoids local and global minima problems.

Proposed Algorithm for Emotional ELMAN NN model

The developed Emotional ELMAN Neural Network model is a multi layer feed forward neural network with a single hidden layer and as well a single recurrent layer connected to the hidden layer. The architecture of the proposed Emotional ELMAN model is as given in Figure 1. Gradient descent learning rule is employed and hyperbolic sigmoidal activation function is used for computing the outputs of the developed EENN model. The proposed training algorithm of EENN model consists of the following five phases:

- Weight initialization process
- Computing the output of the network employing feed forward phase
- Calculating the emotional anxiety and confidence coefficients
- Transformation of information from the recurrent layer phase
- Updating the weights and bias of the network.

Figure 1: Architecture of the proposed Emotional ELMAN neural network model

In the architecture given in Figure 1, it is noted that each of the layers perform independent computations and they receive and pass the results to another layer and then determine the output the network. The network in the proposed model learns based on the current input along with the record of the previous state output. Also, the value of X(k) gets transmitted through the second connection multiplied with that of the considered activation function. Henceforth, the past information reflects on the considered ELMAN neural network model.

The various steps involved in the algorithmic flow of proposed EENN model are as follows:

Step 1: Initialize the weight between the input layer to hidden layer, hidden layers to output layer and recurrent layers to small random values.

Step 2: Perform initialization of the momentum factor and learning rate parameter.

Step 3: When the stopping condition is false do steps 4-11.

Step 4: For each the training dataset pair do steps 5-10.

Step 5: Each input unit belonging to the input layer receives the input signals $x_i$ and transmits this signals to all units in the hidden layer above i.e. to the hidden units

Step 6: Each hidden layer units (h_j, j=1, ..., p) sums the received weighted input signals

$$h_{inj} = w_{oij} + \sum_{i=1}^{n} x_i w_{ij}$$

(1)

Applying the continuous hyperbolic sigmoidal activation function at this point,

$$H_j = f(h_{inj}) \text{ i.e. } f(H_{inj}) = 1/(1+e^{-H_{inj}})$$

(2)

and sends this signal to all units in the layer above i.e. output units.

Step 7: For each of the output unit (f_k, k=1, ..., m), compute its net input,

$$f_{-ink} = v_{ok} + \sum_{j=1}^{p} h_j v_{jk}$$

(3)

and apply the hyperbolic sigmoidal activation function to the net input to calculate the output signals.

$$F_k = f(f_{-ink}) \text{ i.e. } f(F_{ink}) = 1/(1+e^{-F_{ink}})$$

(4)

Step 8: Considering the emotional factors, the anxiety coefficient is given by,

$$\mu_{ac} = F_{average} + \beta$$

(5)

Where, $F_{average}$ is the average value of all presented samples to the neural network in each iterations and the ‘$\beta$’ represents the feedback error given by,

$$\beta = \frac{\sum_{k=1}^{m} (t_k - f_k)^2}{m}$$

(6)

Where, ‘$t_k$’ is the desired target and ‘$f_k$’ is the computed output.

The confidence coefficient is given by,

$$\zeta_{ac} = \mu_{ac(0)} - \mu_{ac(i)}$$

(7)

Where, ‘$\mu_{ac(0)}$’ is the initial anxiety coefficient and ‘$\mu_{ac(0)}$’ is the anxiety coefficient during the iteration process.
Step 9: Each output unit (f_i, k=1, ..., m) receives a target pattern corresponding to an input pattern, error information term is calculated as,
\[ \delta_k = (t_k - f_k) J(f_{i=1...k}) \]  
(8)
Step 10: Each hidden unit (h_j, j=1, ..., n) sums its delta inputs from units in the layer above,
\[ \delta_{-inj} = \sum_{k=1}^{m} \delta j v_{jk} \]  
(9)
Error information term is calculated as,
\[ \delta j = \delta_{-inj}f^j(h_{-inj}) \]  
(10)
Step 11: Compute the weight correction term between the output unit and hidden unit and is given by,
\[ \Delta \nu_{jk} = \alpha \delta_j h_j + \phi \Delta w_{jk} \text{(old)} \]  
(11)
And the bias correction term is given by,
\[ \Delta \nu_{ok} = \alpha \delta_k + \phi \Delta w_{ok} \text{(old)} \]  
(12)
Step 12: Compute the weight correction term between the hidden unit and input unit, recurrent unit is given by,
\[ \Delta \nu_{ij} = \alpha \delta_j x_j + \phi \Delta w_{ij} \text{(old)} \]  
(13)
And the bias correction term is given by,
\[ \Delta \nu_{ej} = \alpha \delta_j + \phi \Delta w_{ej} \text{(old)} \]  
(14)
Step 13: Each output unit (f_i, k=1, ..., m) updates its bias and weights (j=0, ..., p) are given by,
\[ v_{jk}(\text{new}) = v_{jk}(\text{old}) + \Delta \nu_{jk} + \Delta \nu_{ek} \]  
(15)
Step 14: Each recurrent hidden unit (h_j, j=1, ..., p) updates its bias and weights (i=0, ..., n) and are given by,
\[ w_{ij}(\text{new}) = w_{ij}(\text{old}) + \Delta \nu_{ij} + \Delta \nu_{ej} \]  
(16)
\[ w_{oj}(\text{new}) = w_{oj}(\text{old}) + \Delta \nu_{oj} \]  
(17)
Step 14: Test for the stopping condition of the EENN model. The stopping condition can be the number of iterations reached; minimization of the MSE value or the learning rate gets decreased to a particular value.

**Proposed Hybrid GSA-CSSA Algorithm**

This section presents the proposed hybrid gravitational search algorithm-charged system search algorithm which is employed in this paper for tuning the weights of the proposed emotional ELMAN neural network model. Fundamentally, GSA operates on the law of gravity and based on the literature studies made; it is observed that improvement can be made in the exploration and exploitation process of the search mechanism. In a similar manner, classical CSSA when operated alone gets more time to perform exploitation process and henceforth the merits of GSA and CSSA are combined together resulting in hybrid GSA-CSSA to perform effective exploration and exploitation search mechanism and achieve better weight values to train proposed EENN model.

**Gravitational Search Algorithm-An Overview**

In GSA algorithm [59], the objects (particles) are evaluated with their masses with four features: particle position, inertial mass, active gravitational mass, and passive gravitational mass. Each position of the object provides a solution. The gravitational and inertia masses are navigated with the fitness function of the problem defined. The system in GSA is well defined with N mass (agent) where the position of the i th agent is denoted as:
\[ x_i = (x_1^i, x_2^i, ..., x_n^i) \text{for } i = 1, 2, 3, ..., N \]  
(19)
Where \( x_i^d \) denotes present position in d th dimension of agent i and n the search space dimension. The force applied between \( i \) th and \( j \) th mass at time t is given as:
\[ F_{ij}^d = \frac{G(t)M_{ai}(t)M_{pj}(t)}{R_{ij}(t)^2} \left( x_j^d(t) - x_i^d(t) \right) \]  
(20)
Where \( M_{pj}(t) \) is the active gravitational mass, \( M_{ai}(t) \) is the passive gravitational mass, \( \zeta \) is the small positive constant, \( R_{ij}(t) \) denotes the Euclidean distance between particle i and j at time (t),
\[ R_{ij}(t) = \left\| x_i(t) - x_j(t) \right\|_2 \]  
(21)
\( G(t) \) represents the gravitational coefficient defined as in eq (22),
\[ G(t) = G_0 e^{-\alpha^2 t / T} \]  
(22)
Where, \( G_0 \) and \( \alpha \) are constant with T being the maximum iteration. The total resultant force exerted on mass i in d dimension,
\[ F_{i}^d(t) = \sum_{j=1}^{N} F_{ij}^d(t) \]  
(23)
where \( r_j \) is a uniformly distributed random number in the interval [0, 1]. k is the set of first K agents with the best fitness value to avoid local optimal solutions, where only the k best masses, i.e., the ones with highest fitness values will attract the others. Hence by law of motion, the acceleration of the mass i at time i in d th dimension is defined as,
\[ a_i^d = \frac{F_{i}^d(t)}{M_i(t)} \]  
(24)
where the \( M_i(t) \) is the inertial mass of particle i. The gravitational mass and the inertial mass are updated by the following equations:
\[ M_{ai} = M_{pi} = M_i(i = 1, 2, ..., N) \]  
(25)
\[ m_i(t) = \frac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)} \]  

(26)

\[ M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \]  

(27)

where \( \text{fit}_i(t) \) denotes the fitness value of mass \( i \). For the minimization problem, the worst \( (i) \) and best\((i) \) are given as,

\[ \text{best}(t) = \min_{j=1,2,\ldots,N} \text{fit}_j(t) \]  

(28)

\[ \text{worst}(t) = \max_{j=1,2,\ldots,N} \text{fit}_j(t) \]  

(29)

In GSA, the updation of velocity and position in each iteration is in accordance with Newton’s laws of motion is formulized as:

\[ v_i^d(t + 1) = \text{rand} \cdot v_i^d(t) + a_i^d(t) \]  

(30)

\[ x_i^d(t + 1) = x_i^d(t) + v_i^d(t + 1) \]  

(31)

Where \( \text{rand} \) is a random number between the interval between [0, 1].

**Charged System Search Algorithm-Revisited**

Charged System Search Algorithm [60] is a population based stochastic evolutionary algorithm modeled on the fundamental concepts of the Coulomb and Gauss laws from electrical physics and the governing laws of motion from the Newtonian mechanics. This technique is a multi-agent based approach, where each agent acting will be called as the Charged Particles (CP). With respect to the theory of physics, the charged particles are modeled to be charged sphere with radius and possessing uniform volume charge density resulting in electric force on the other charged particles. The charged particles movement and position change is done based on the calculated force magnitude. The force magnitude for charged particles will be located inside the sphere and this is proportional to that of the separation distance between the charged particles, on the other hand the charged particles located outside the sphere is observed to be inversely proportional to the square of the separating distance between the particles. Thus the new position of the charged particles is computed based on the resultant forces or acceleration and the motion laws. The basic algorithm of CSSA is as follows:

**Step 1:** Initialize an array of charged particles with random positions and their associated velocities are assumed to be zero.

**Step 2:** Compute the fitness function for the CPs. Arrange the CPs in an increasing order based on the fitness evaluated.

**Step 3:** A number of the first CPs and their respective fitness function values will get stored in a memory, called as Charged Memory (CM).

**Step 4:** Calculate the forces on CPs. The force vector is calculated for each CP as

\[ F_j = \sum_{i \neq j} \left( \frac{q_i q_j}{r_{ij}^3} \right) \]  

(32)

where \( F_j \) is the resultant force acting on the \( j \)-th CP; \( N \) is the number of CPs. The magnitude of charge for each CP \((q_i)\) is defined considering the quality of its solution.

\[ q_i = \frac{\text{fit}_i - \text{fit}\text{worst}}{\text{fitbest} - \text{fit}\text{worst}}, \quad i = 1,2,\ldots,N \]  

(33)

where ‘fitbest’ and ‘fitworst’ are the best and the worst fitness of all CPs, respectively; \( \text{fit}(i) \) represents the fitness of the agent \( i \); and \( N \) is the total number of CPs. The separation distance \( r_{ij} \) between two charged particles is defined as follows:

\[ r_{ij} = \left\| \frac{X_i - X_j}{2 - X_{best}} \right\| + \varepsilon \]  

(34)

where \( X_i \) and \( X_j \) are respectively the positions of the \( i \)-th and \( j \)-th CPs, \( X_{best} \) is the position of the best current CP, and \( \varepsilon \) is a small positive number. Here, \( p_{ij} \) is the probability of moving each CP towards the others and the function for evaluating this is given by.

\[ p_{ij} = \begin{cases} 1 & \frac{\text{fit}(i) - \text{fitbest}}{\text{fit}(j) - \text{fit}(i)} > \text{rand} \wedge \text{fit}(j) > \text{fit}(i) \\ 0 & \text{else} \end{cases} \]  

(35)

In Eq. (32), \( \text{rand} \) indicates the kind of force and is given by,

\[ \text{rand} = \begin{cases} +1 & \text{rand} < 0.8 \\ -1 & \text{otherwise} \end{cases} \]  

(36)

where \( \text{rand} \) represents a random number.

**Step 5:** In this step, solution has to be constructed. Each CP moves to the new position and the new velocity is computed as,

\[ X_{j,\text{new}} = c_1 \cdot k_a \cdot F_j + c_2 \cdot k_v \cdot V_{j,\text{old}} + \alpha X_{j,\text{old}} \]  

(37)

\[ V_{j,\text{new}} = X_{j,\text{new}} - X_{j,\text{old}} \]  

(38)

In equation (37), \( k_a \) is the acceleration coefficient; \( k_v \) is the velocity coefficient to control the influence of the previous velocity; and \( c_1 \) and \( c_2 \) are two random numbers uniformly distributed in the range \((0, 1)\) and \( \alpha \) is the scaling factor.

**Step 6:** Process of updation is now carried out. If a new CP exits from the allowable search space, then sort it increasingly.
to correct its position. In addition, if some new CP vectors are better than the worst ones in the CM; these are replaced by the worst ones in the CM.

**Step 7:** Apply termination control criterion. Perform Steps 2-6 repeatedly until termination criterion is met.

**Proposed Hybrid GSA-CSSA technique**

To avoid the premature convergence and dimensionality problem of the fundamental gravitational search optimization algorithm, the basic GSA version is hybridized with CSSA in order to overcome the said drawbacks. The CSSA algorithm is invoked when the best particles do not get changed over time. The introduction of CSSA increases the exploitation rate of the search process by the charged sphere mechanism employed. With respect to the resultant force, the charged particle position and the location of the new charged point, CSSA algorithm moves in the search space. The algorithmic steps of the proposed hybrid GSA-CSSA technique is as given below:

**Step 1:** Start the algorithmic process.

**Step 2:** Initialize the number of particles (charged particles), gravitational coefficient and other related constant values.

**Step 3:** For each generation

**Step 4:** For each particle (charged particle) randomly generated

**Step 5:** Calculate the fitness function.

**Step 6:** Compute the resultant force using equation (20) and (23).

**Step 7:** Update the gravitational mass and inertial mass employing the equations (25) to (27).

**Step 7:** If global best is the best value attained so far in the generations; return the value.

**Step 8:** Based on the particle best and global best value, update the particle position and henceforth the velocity using equations (30) to (31). Return these and update the memory of each of the particle.

**Step 9:** If the best particle does not change over time, invoke CSSA module.

**Step 10:** Perform charged particle ranking and store it in charged memory.

**Step 11:** Determine the attracting force vector

**Step 12:** Construct the solution by moving the charged particles

**Step 13:** Perform position correction of the charged particles

**Step 14:** Do the charged particle ranking based on the fitness evaluated.

**Step 15:** Compare the value of the objective functions and sort them increasingly.

**Step 16:** Exclude charged particles with minimum fitness and increase the newer ones with better fitness into the charged memory.

**Step 17:** Test for the stopping condition as devised

**Step 18:** Return the charged particles stored in the charged memory with a better fitness.

**Step 19:** Stop the algorithmic process.

The proposed hybrid GSA-CSSA approach is embodied into the ELMAN neural network to make it act as a predictor model to address the occurrence of software defects in the NASA Promise repository datasets.

**Proposed Hybrid GSA-CSSA based Emotional ELMAN Neural Network**

This research paper focuses on developing hybrid gravitational search algorithm-charged system search algorithm based emotional ELMAN neural network for performing effective software prediction of the considered datasets. The advantages of the hybrid version of GSA and CSSA along with proposed emotional neural networks are brought out to perform effective prediction action. This section presents the proposed hybrid GSA-CSSA-EENN predictor to determine the software defects of the considered datasets.

The key objective of introducing GSA-CSSA combination into emotional ELMAN neural network architecture is the ability to adapt the change in environment, escape from local and global optima problem and ensure robustness of the system. In this paper, the computation of optimal connection weights is carried out and minimization of error function (mean square error-MSE) is employed as an objective function. This paper concentrates on evolution of weights using hybrid GSA-CSSA and then performing the prediction action using the proposed EENN classifier. Table 4 presents the pseudo code for proposed hybrid GSA-CSSA-EENN predictor. The proposed hybrid GSA-CSSA-EENN predictor is applied to improve the accuracy and minimize the mean square error of the software defect prediction module.

**Table 4:** Pseudo code for proposed hybrid GSA-CSSA-EENN Predictor

<table>
<thead>
<tr>
<th>Phase I:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
</tr>
<tr>
<td>Initialize the weights, learning rate and bias values of EENN model.</td>
</tr>
<tr>
<td>Random generation of weight and bias values.</td>
</tr>
<tr>
<td>Present the input details to the EENN model. Calculate the net input of the designed layers</td>
</tr>
<tr>
<td>Apply activations to compute output for the calculated net input.</td>
</tr>
<tr>
<td>Compute the anxiety and confidence coefficients.</td>
</tr>
<tr>
<td>Carry out the above process for input layer to hidden layer and hidden layer to output layer and the recurrent layer. Compute the output from the output layer.</td>
</tr>
<tr>
<td>Update the weights till termination condition met (MSE reaches a minimal value)</td>
</tr>
</tbody>
</table>
Phase II:
Present the computed outputs from EENN model into proposed Hybrid GSA-CSSA.
Invoke proposed Hybrid GSA-CSSA
Evaluate Fitness value
Perform velocity and position updation
If number of particles does not get updated over generations
Invoke CSSA
Assign the current particles to the members in the group.
Construct the solution by moving the charged particles
Perform position correction of the charged particles
Do the charged particle ranking based on the fitness evaluated.
Compare the value of the objective functions and sort them increasingly.
Exclude charged particles with minimum fitness and increase the newer ones with better fitness into the charged memory.
Present the points of best fitness to EENN model.

Phase III:
Invoke EENN model with the inputs from the output of Phase II
Tune the weights to this EENN with GSA-CSSA approach
Invoke Hybrid GSA-CSSA
Perform Phase I process for tuned weights of GSA-CSSA
Continue Phase II.
Repeat until stopping condition met (Stopping conditions can be number of iterations/ generations or attaining an optimal fitness value)
Stop.

Experimental Results and Discussion
The developed hybrid GSA-CSSA based Emotional ELMAN neural network model is applied for the five public datasets as available in the NASA Promise repository dataset and the computed simulation results are presented in this section. The various performance metrics employed for validating the proposed algorithm and as well that are calculated is also given in this section. It should be noted that the proposed hybrid GSA-CSSA-EENN algorithm is applied and simulation results are computed for both the non-cost sensitive case and cost sensitive case.

Performance metrics computed
Performance metrics play a major role in developing the predictive model and analyzing the performance of the proposed predictors. Table 5 gives the various performance metrics employed in this research paper.

Table 5: Performance metrics for predictor model

<table>
<thead>
<tr>
<th>Performance Metrics</th>
<th>Definition of the metric</th>
<th>Metric Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity (or) Recall (or) True Positive Rate (or) Probability of Detection (pd)</td>
<td>TP TP + FN</td>
<td>Proportion of Defective Modules correctly predicted</td>
</tr>
<tr>
<td>Precision</td>
<td>TP TP + FP</td>
<td>Proportion of modules predicted as defective</td>
</tr>
<tr>
<td>False Positive Rate (or) Probability of False Alarm (pf)</td>
<td>FP FP + TN</td>
<td>Proportion of Non-defective modules predicted as defective</td>
</tr>
<tr>
<td>Specificity</td>
<td>TN TN + FP</td>
<td>Proportion of correctly predicted non-defective modules</td>
</tr>
<tr>
<td>Classification Accuracy</td>
<td>TN + TP TN + FN + FP + TP</td>
<td>Proportion of correctly predicted modules</td>
</tr>
<tr>
<td>Balance</td>
<td>1 - \sqrt{(0 - pf)^2 + (1 - pd)^2 \over 2}</td>
<td>Balance combines pf and pd into one measure and is defined as the distance from the ROC ‘sweet spot’ (where pd=1, pf=0).</td>
</tr>
<tr>
<td>Receiver Operating Characteristics Curve (ROC)</td>
<td>A graphical plot of ‘pd’ vs ‘pf’ where the discrimination threshold is varied.</td>
<td></td>
</tr>
</tbody>
</table>

The accuracy will not be appropriate for datasets possessing uneven class distribution. An optimal predictor that performs effective prediction should achieve a TPR (pd) of 1, FPR (pf) of 0 and precision of 1. When the computed ‘pd’ and ‘pf’ are plotted, they result in Receiver Operating Characteristics (ROC) curve and from ROC; the area under the curve (AUC) is to be noted. AUC is noted to be between 0 and 1, and 1 being the optimal solution point. Certain predictors result in low AUC values, but can be tuned further to produce high balance metrics. Further to this, this paper considers the implementation of the algorithm with both the cost-sensitive and non-cost sensitive case. Henceforth, the objective function of the cost-sensitive EENN model to be minimized employing the proposed hybrid GSA-CSSA is given by the following equation

\[
\min_{NECM} = A + B
\]  

(39)

where \( A = pf \times P_{\text{non-defect-prone}} \) and \( B = \frac{\text{cost false-negative}}{\text{cost false-positive}} \times pf_{\text{true}} \times P_{\text{defect-prone}} \)
Where, ‘NECM’ is the normalized expected cost of misclassification, ‘pf’ is the false positive rate, ‘pfnr’ represents the false negative rate, ’cost\false\_positive’ is the cost pertaining to false positive error, ’cost\false\_negative’ is the cost pertaining to false negative error, ‘P\non\_defect\_prone’ and ‘P\defect\_prone’ are the percentage of non-defect prone modules and defect prone modules respectively. For the non-cost sensitive case, the equation (39) gets re-modified as,

\[
\min = pf \times P\non\_defect\_prone + pfnr \times P\defect\_prone
\]

i.e., the cost of false positive and false negative are assumed to be of equal weight and thus, \( \frac{cost\false\_negative}{cost\false\_positive} = 1 \) in equation (39).

### Parameters Employed for Simulation

The optimal parameters chosen for the operation of hybrid GSA-CSSA based EENN algorithm is tabulated in Table 6.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>EENN model</th>
<th>Parameters</th>
<th>Hybrid GSA-CSSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning rate</td>
<td>0.2</td>
<td>Number of particles (agents)</td>
<td>30</td>
</tr>
<tr>
<td>Momentum factor</td>
<td>0.1</td>
<td>Gravitational constants</td>
<td>100</td>
</tr>
<tr>
<td>No. of Hidden Neurons</td>
<td>( \frac{1}{2} ) the number of input neurons</td>
<td>c1 and c2</td>
<td>0.3</td>
</tr>
<tr>
<td>Maximum iteration</td>
<td>100</td>
<td>Number of iterations</td>
<td>100</td>
</tr>
<tr>
<td>Activation function</td>
<td>Sigmoidal</td>
<td>Activation Function</td>
<td></td>
</tr>
<tr>
<td>No. of output neurons</td>
<td>1 (Defect or Defect-free)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Simulation Results for the Proposed Prediction Model (Non Cost-Sensitive Case)

The objective function employed for this case is as given in equation (40) i.e., the simulation results are obtained without considering the cost-sensitive component. The methodology is implemented for NASA Promise datasets given in Table 1. The performance results of these datasets are given in Table 7.

<table>
<thead>
<tr>
<th>NASA Datasets</th>
<th>Sensitivity (pd)</th>
<th>Specificity (s)</th>
<th>False Positive Rate (FPR or pf)</th>
<th>Balance</th>
<th>Accuracy</th>
<th>Area Under ROC (AUC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JM 1</td>
<td>77.82</td>
<td>81.32</td>
<td>36.92</td>
<td>69.04</td>
<td>66.49</td>
<td>0.84</td>
</tr>
<tr>
<td>KC 1</td>
<td>84.87</td>
<td>85.01</td>
<td>24.38</td>
<td>81.99</td>
<td>82.63</td>
<td>0.91</td>
</tr>
<tr>
<td>KC 2</td>
<td>86.95</td>
<td>85.44</td>
<td>19.46</td>
<td>82.95</td>
<td>89.39</td>
<td>0.88</td>
</tr>
<tr>
<td>PC 1</td>
<td>89.81</td>
<td>88.06</td>
<td>35.27</td>
<td>72.90</td>
<td>93.44</td>
<td>0.92</td>
</tr>
<tr>
<td>Mean</td>
<td>84.06</td>
<td>83.78</td>
<td>29.21</td>
<td>76.36</td>
<td>84.72</td>
<td>0.8840</td>
</tr>
</tbody>
</table>

From Table 7, it is observed that the area under curve value is noted to be greater than 0.8 for all the five cases, conveying that the proposed predictor model has resulted in acceptable solutions. On viewing the accuracy and area under curve metrics, KC2 and PC1 datasets are observed to result in better solutions than the other considered three datasets. The proposed hybrid GSA-CSSA without the cost factor is simulated for 30 trial runs and the specified solutions in Table 9 are obtained. The resulted solutions in Table 7 prove the effectiveness and robustness of the non-cost-sensitive hybrid GSA-CSSA-EENN predictor model. Receiver Operating Characteristics is studied for the proposed classifier and the resulted plots are presented Figure 2(a) through (e). ROC curve proves the effectiveness of the proposed model with the AUC values being nearer to unity.
Probability of Detection

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Probability of False Alarm

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Receiver Operating Characteristics - NASA KC1 Dataset

AUC=0.91

Receiver Operating Characteristics - NASA KC2 Dataset

AUC=0.88

Receiver Operating Characteristics - NASA PC1 Dataset

AUC=0.92

Table 8 in Appendix gives the comparison of the proposed classifier with the other algorithms [14, 56, 62] applied for the same NASA datasets in terms of the performance metrics—sensitivity, specificity, probability of false alarm, balance, accuracy, area under curve and error value. From Table 8, it is inferred that the proposed hybrid GSA-CSSA based emotional EENN is noted to possess solution than that of the earlier methods from the literature. With respect to ROC characteristic and AUC value, the proposed hybrid GSA-CSSA based EENN is noted to possess values nearer to 1, proving the validity of the results computed.

<table>
<thead>
<tr>
<th>NASA Datasets</th>
<th>Technique</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>FPR or F: E</th>
<th>Balance</th>
<th>Accuracy</th>
<th>AUC</th>
<th>MSE (Error)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM 1</td>
<td>Naive Bayes [56]</td>
<td>71.03</td>
<td>78.65</td>
<td>34.09</td>
<td>68.37</td>
<td>64.57</td>
<td>0.75</td>
<td>0.1456</td>
</tr>
<tr>
<td></td>
<td>Random Forest [56]</td>
<td>70.09</td>
<td>71.29</td>
<td>32.17</td>
<td>68.94</td>
<td>60.98</td>
<td>0.74</td>
<td>0.2314</td>
</tr>
<tr>
<td></td>
<td>C4.5 Miner [56]</td>
<td>74.91</td>
<td>74.66</td>
<td>27.68</td>
<td>73.58</td>
<td>66.71</td>
<td>0.53</td>
<td>0.3765</td>
</tr>
<tr>
<td></td>
<td>Immunos [56]</td>
<td>73.65</td>
<td>75.02</td>
<td>30.99</td>
<td>71.24</td>
<td>66.03</td>
<td>0.63</td>
<td>0.1732</td>
</tr>
<tr>
<td></td>
<td>ANN-ABC [56]</td>
<td>75.00</td>
<td>81.00</td>
<td>33.05</td>
<td>71.00</td>
<td>68.00</td>
<td>0.77</td>
<td>0.2435</td>
</tr>
<tr>
<td></td>
<td>Hybrid Self Organizing Map [14]</td>
<td>70.12</td>
<td>78.96</td>
<td>30.63</td>
<td>69.73</td>
<td>72.37</td>
<td>0.80</td>
<td>0.0810</td>
</tr>
<tr>
<td></td>
<td>Support vector Machine [62]</td>
<td>78.97</td>
<td>79.08</td>
<td>31.27</td>
<td>73.35</td>
<td>78.69</td>
<td>0.79</td>
<td>0.0154</td>
</tr>
<tr>
<td></td>
<td>Majority Vote [62]</td>
<td>79.80</td>
<td>80.00</td>
<td>30.47</td>
<td>74.16</td>
<td>77.01</td>
<td>0.81</td>
<td>0.1968</td>
</tr>
<tr>
<td></td>
<td>AntMiner+ [62]</td>
<td>80.65</td>
<td>78.88</td>
<td>30.96</td>
<td>74.22</td>
<td>79.43</td>
<td>0.84</td>
<td>0.0345</td>
</tr>
<tr>
<td></td>
<td>Proposed Hybrid GSA-CSSA-EENN model</td>
<td>80.86</td>
<td>79.05</td>
<td>30.01</td>
<td>74.93</td>
<td>81.65</td>
<td>0.87</td>
<td>0.0149</td>
</tr>
<tr>
<td>JM 1</td>
<td>Naive Bayes [56]</td>
<td>68.98</td>
<td>69.77</td>
<td>36.54</td>
<td>66.11</td>
<td>60.78</td>
<td>0.68</td>
<td>0.6547</td>
</tr>
<tr>
<td></td>
<td>Random Forest [56]</td>
<td>66.72</td>
<td>72.38</td>
<td>33.47</td>
<td>66.62</td>
<td>63.97</td>
<td>0.75</td>
<td>0.6721</td>
</tr>
<tr>
<td></td>
<td>C4.5 Miner [56]</td>
<td>69.08</td>
<td>68.55</td>
<td>40.67</td>
<td>63.87</td>
<td>62.35</td>
<td>0.61</td>
<td>0.5498</td>
</tr>
<tr>
<td></td>
<td>Immunos [56]</td>
<td>70.99</td>
<td>70.21</td>
<td>43.00</td>
<td>63.32</td>
<td>64.55</td>
<td>0.63</td>
<td>0.4219</td>
</tr>
<tr>
<td></td>
<td>ANN-ABC [56]</td>
<td>71.00</td>
<td>73.05</td>
<td>41.00</td>
<td>64.00</td>
<td>61.00</td>
<td>0.71</td>
<td>0.4057</td>
</tr>
<tr>
<td></td>
<td>Hybrid Self Organizing Map [14]</td>
<td>71.02</td>
<td>74.90</td>
<td>40.57</td>
<td>64.75</td>
<td>72.33</td>
<td>0.82</td>
<td>0.5692</td>
</tr>
<tr>
<td></td>
<td>Support vector Machine</td>
<td>70.89</td>
<td>79.00</td>
<td>39.87</td>
<td>65.09</td>
<td>70.32</td>
<td>0.81</td>
<td>0.3759</td>
</tr>
</tbody>
</table>

Table 8: Comparison results and error analysis on NASA Promise datasets (Non cost-sensitive case)
Simulation results for the proposed prediction model (cost-sensitive case)

This paper also implemented the proposed hybrid GSA-CSSA based emotional ELMAN neuronal model considering the cost sensitive case with the objective function as in equation (39) and locates the defect free and defect prone modules. This subsection, details the simulated results for the considered NASA Promise repository datasets with cost-sensitive factor included and tuned employing the proposed algorithmic approach. Table 9 in Appendix presents the results computed on employing the proposed classifier with four different cost ratios and their comparison of results with the existing methodologies from the literature [56]. The values of cost ratio are considered from literature [56].

Table 9: Performance and Comparison results of the proposed predictor on the five NASA Promise datasets (cost-sensitive case-four different cost ratios)

<table>
<thead>
<tr>
<th>NASA Datasets</th>
<th>CR (Cost Ratio)</th>
<th>Performance Metrics</th>
<th>CR (Cost Ratio)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\text{COS})</td>
<td>false negative</td>
<td>(\text{COS})</td>
</tr>
<tr>
<td>CR = 4.00</td>
<td>CR = 1.50</td>
<td>CR = 0.67</td>
<td>CR = 0.25</td>
</tr>
<tr>
<td>ANNSaved</td>
<td>ANNSaved</td>
<td>ANNSaved</td>
<td>ANNSaved</td>
</tr>
<tr>
<td>[56] EENN</td>
<td>[56] EENN</td>
<td>[56] EENN</td>
<td>[56] EENN</td>
</tr>
<tr>
<td>Model</td>
<td>Model</td>
<td>Model</td>
<td>Model</td>
</tr>
<tr>
<td>Proposed</td>
<td>Proposed</td>
<td>Proposed</td>
<td>Proposed</td>
</tr>
<tr>
<td>Hybrid</td>
<td>Hybrid</td>
<td>Hybrid</td>
<td>Hybrid</td>
</tr>
<tr>
<td>GSA-CSSA-EENN</td>
<td>GSA-CSSA-EENN</td>
<td>GSA-CSSA-EENN</td>
<td>GSA-CSSA-EENN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minority</td>
<td>86.95</td>
<td>85.44</td>
<td>19.4</td>
</tr>
<tr>
<td>Vote [62]</td>
<td>86.95</td>
<td>85.44</td>
<td>19.4</td>
</tr>
<tr>
<td>AntMiner+</td>
<td>86.95</td>
<td>85.44</td>
<td>19.4</td>
</tr>
<tr>
<td>[62]</td>
<td>86.95</td>
<td>85.44</td>
<td>19.4</td>
</tr>
<tr>
<td>Proposed</td>
<td>86.95</td>
<td>85.44</td>
<td>19.4</td>
</tr>
<tr>
<td>Hybrid</td>
<td>86.95</td>
<td>85.44</td>
<td>19.4</td>
</tr>
<tr>
<td>GSA-CSSA-EENN</td>
<td>86.95</td>
<td>85.44</td>
<td>19.4</td>
</tr>
<tr>
<td>model</td>
<td>model</td>
<td>model</td>
<td>model</td>
</tr>
<tr>
<td>Proposed</td>
<td>86.95</td>
<td>85.44</td>
<td>19.4</td>
</tr>
<tr>
<td>Hybrid</td>
<td>86.95</td>
<td>85.44</td>
<td>19.4</td>
</tr>
<tr>
<td>GSA-CSSA-EENN</td>
<td>86.95</td>
<td>85.44</td>
<td>19.4</td>
</tr>
<tr>
<td>model</td>
<td>model</td>
<td>model</td>
<td>model</td>
</tr>
</tbody>
</table>
In this paper, a novel hybrid gravitational search algorithm and charged system search algorithm is developed to tune the optimal weight values of the developed emotional ELMAN neural network model. The hybridization of GSA and CSSA is carried out to achieve faster convergence and perform effective exploration and exploitation process in the search mechanism. The emotional ELMAN model is developed with the anxiety and confidence coefficients included and this makes the neural network minimize the error at faster rate and avoid local and global minima occurrences. The developed hybrid GSA-CSSA based EENN model is applied for the public datasets from the NASA Promise repository. From the simulation results, it is proved that the proposed predictor model involving the merits of GSA, CSSA and EENN model has resulted in better prediction accuracy in comparison with that of the earlier existing methodologies as available in the literature. The results were simulated with respect to both the cost sensitive and non cost-sensitive case.

### References

8. Yadav, Harikesh Bahadur, and Dilip Kumar Yadav. "A fuzzy logic based approach for phase-wise software defects prediction using software..."


