6.1 INTRODUCTION

Seismic hazard analysis of slope needs to be evaluated in terms of both factor of safety and earthquake induced displacement. The evaluation of factor of safety of slopes with pseudo static method using wedge method has been discussed in Chapter 4. Similarly the factor of safety using circular slip surface for different seismic coefficient has been elaborated in Chapter 5. In Chapter 2, seismic and liquefaction induced displacement have been discussed. This chapter pertains to displacement of slopes and sloping ground due to earthquakes and earthquake induced liquefaction. The artificial intelligence techniques artificial neural network and support vector machine have been used to predict the same.

Artificial neural network (ANN) is now being used successfully as an alternate statistical method with high predictability as discussed in Chapter 3. However, it is also known as black box system, has some inherent drawbacks such as slow convergence speed, less generalizing performance, arriving at local minimum and over-fitting problems. Support vector machine (SVM) is an emerging machine learning technology where prediction error and model complexity are simultaneously minimized. In different complex engineering problems the SVM models are being used and found to perform better that ANN models in terms of generalization. However, use of SVM in geotechnical engineering is limited (Pal 2006, Goh and Goh 2007, Samui 2008, Samui et al. 2008).
With the above in view, in the present work SVM models (SVM-G, SVM-P, SVM-S) have been developed to predict the liquefaction induced horizontal ground displacement. Different ANN models (BRNN, LMNN and DENN) as described in Chapter 3 are also developed to compare the results. The statistical performance criteria like overfitting ratio, maximum absolute error, average absolute error and root mean square error are used to evaluate different ANN and SVM models.

6.2 METHODOLOGY

The details of different ANN models adopted here have been already discussed in Chapter 3. The details of SVM and RVM can be found in many literatures (Vapnik; Smola and Scholkopf), but a brief methodology on SVM and RVM are presented here for completeness.

6.2.1 Support Vector Machine

The SVM originally developed for classification/pattern recognition problems is now being used as regression technique. In the present study, the SVM has been used as support vector regression (SVR).

Linear regress using SVR

There are two basic aims in SVR; first is to find a function \( f(x) \) that has at most a deviation from each of the targets of the training inputs and secondly the function to be flat is possible. For the linear case, \( f \) is given by

\[
f(x) = \langle w, x \rangle + b
\]

(6.1)

where \( w \) is an adjustable weight vector and \( b \) is the scalar threshold (bias).
The flat function refers to a small $w$ in Eq. (6.1). This second aim is important in the formulation of the optimization problem used to construct the SVR approximation.

$$\text{Minimize} \quad \frac{1}{2} |w|^2$$

Subject to

$$\begin{cases} 
  y_i - (w^T x_i) - b \leq \varepsilon \\
  (w^T x_i) + b - y_i \leq \varepsilon 
\end{cases} \quad (6.2)$$

The above Equation is valid as long as the error is less than $\varepsilon$ i.e. within the $\varepsilon$ – tube (Fig. 1), which is also known as error-insensitive zone. The prediction errors more than $\pm \varepsilon$ are taken into the above optimization problem by use of slack variables ($\xi_i$ and $\xi_i^*$) and generalization parameter (penalty function) $C$. The slack variables, $\xi_i$ and $\xi_i^*$, can be incorporated into the optimization problem to yield the following formulation.

$$\text{Minimize} \quad \frac{1}{2} |w|^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*)$$

Subject to

$$\begin{cases} 
  y_i - (w^T x_i) - b \leq \varepsilon + \xi_i \\
  (w^T x_i) + b - y_i \leq \varepsilon + \xi_i^* \\
  \xi_i, \xi_i^* \geq 0
\end{cases} \quad (6.3)$$

Where the constant $C > 0$ determines the tradeoff between flatness (small $w$) and the degree to which deviation larger than $\xi$ are tolerated (see Fig. 6.1), and $l$ is the number of samples. Using this penalty function, the goal is to simultaneously minimize both the coefficient size thereby improving generalization and prediction error. This is an approach which is often used in multivariate calibration and that can deal with ill-posed problems.
Figure 6.1. Prescribed accuracy $\varepsilon$ and slack variables $\zeta$ and $\zeta^*$ in SVR

This is referred to as the $\varepsilon$-insensitive loss function proposed by Vapnik (1998) and objects with prediction error larger than $\pm \varepsilon$ are known as support vectors which are to be obtained for regression. The optimization function and linear constraints in Eq. (6.3) can be written as the Lagrangian function

$$L = \frac{1}{2} |w|^2 + C \sum_{i=1}^n (\xi_i^+ + \xi_i^-) - \sum_{i=1}^n \alpha_i (e_i + \xi_i^- y_i + \langle w, x_i \rangle + b - \sum_{i=1}^n \alpha_i^+ (a_i + \xi_i^- y_i - \langle w, x_i \rangle - b)) - \sum_{i=1}^n (\eta_i \xi_i^+ + \eta_i^\prime \xi_i^-)$$

(6.4)

Hence as per Lagrangian theory, necessary conditions for $\alpha$ to be solutions to the original optimization problem are

$$d_\alpha L = \sum_{i=1}^n (\alpha_i^+ - \alpha_i^-) = 0$$

(6.5)

$$d_w L = w - \sum_{i=1}^n (\alpha_i^+ - \alpha_i^-) x_i = 0$$

(6.6)

$$d_{\xi_i} L = C - \alpha_i - \eta_i = 0$$

(6.7)
Substituting Eqs. (6.5 - 6.8) into Eq. (6.8) yields the optimization problem in dual form

\[
\begin{align*}
\text{Minimize} & \quad \left\{ -\frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i - \alpha_j) (\alpha_i - \alpha_j) (x_i, x_j) \right\} \\
\text{Subject to} & \quad \left\{ \sum_{i=1}^{n} (\alpha_i - \alpha_i) x_i = 0 \right\} \\
& \quad \left\{ (\alpha_i - \alpha_j) \in [0, C] \right\}
\end{align*}
\]

Subject to

\[ w = \sum_{i=1}^{n} (\alpha_i - \alpha_i) x_i \]  

(6.10)

So the linear regression in Eq. (6.1) becomes

\[ f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i) (x_i, x) + b \]  

(6.11)

Thus, the training algorithm and the regression function \( f(x) \) can be expressed in terms of the dot product \((x_i, x)\). There are two advantages by transforming the optimization problem into dual form. First, the optimization problem is now a quadratic programming problem with linear constraints and a positive definite Hessian matrix, ensuring a unique global optimum. Second, as can be seen in Eq. (6.9), the input vectors only appear inside the dot product. The dot product of each pair of input vectors is a scalar and can be preprocessed and stored in the quadratic matrix \( M_{ij} = ((x_i, x_j))_{ij} \). In this way, the dimensionality of the input space is hidden from the remaining computations.

**Nonlinear regression using SVR**

For the cases when the regression relation is nonlinear, the input space is mapped into a high dimensional dot-product space called feature space (Figure 6.2). This is achieved by replacing the dot product of input vectors with a
nonlinear transformation on the input vectors in the dual form of the optimization problem. This transformation is referred to as the kernel function, and is represented by $k(x, x')$, where $x$ and $x'$ is input vectors. A number of kernel functions are discussed in literature, but polynomial, radial basis function and spline kernels are mostly used for different civil engineering problems (Smola and Scholkopf 2004; Goh and Goh 2007, Samui et al. 2008).
Applying the kernel function to the dot product of input vectors, we obtain

\[
\begin{align*}
\text{Minimize} & \left\{ -\frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i^* - \alpha_i^0) (\alpha_j^* - \alpha_j^0) k(x_i, x_j) \right. \\
& \left. - \sum_{i=1}^{n} \alpha_i^* y_i + \sum_{i=1}^{n} y_i (\alpha_i^* - \alpha_i^0) \right\}
\end{align*}
\]

Subject to
\[
\begin{align*}
\sum_{i=1}^{n} (\alpha_i^* - \alpha_i^0) &= 0 \\
(\alpha_i^* - \alpha_i^0) &\in [0, C]
\end{align*}
\] (6.12)

Replacing the dot product in Eq. (6.10), the SVR approximation becomes

\[
f(x) = \sum_{i=1}^{n} (\alpha_i^* - \alpha_i^0) k(x_i, x) + b
\] (6.13)

So by using a kernel function and corresponding kernel matrix, nonlinear function approximation can be achieved with SVR while maintaining the simplicity and computational efficiency of linear SVR approximation.

As SVR is trained with optimization of a convex, quadratic cost function, which can be solved by using quadratic programming. In SVM, the number of support vectors has been determined by algorithm rather than by trial-and-error which has been used by artificial neural network (ANN) for determining the number of hidden nodes. However, user-defined parameters like regularization parameter C, kernel specific parameters and the \(\varepsilon\)-insensitive values are to be optimized by trial and error. A large C assigns higher penalties to errors so that the SVM is trained to minimize error with lower generalization while a small C assigns fewer penalties to errors; this allows the minimization of margin with errors, thus higher generalization ability. If C goes to infinitely large, SVM would not allow the occurrence of any error and result in a complex model, whereas when C goes to zero, the result would tolerate a large amount of errors and the model would be less complex.
The SVM formulation leads to a sparse model dependent only on a subset of training examples and their associated kernel functions (Vapnik, 1995). However, SVM suffers the necessity to manually tune some parameters and from the selection of kernel function parameters which also must satisfy Mercer’s condition (Vapnik, 1995; Tipping, 2001). Relevance vector machine (RVM) introduced by Tipping (2001) produces sparse solutions using an improper hierarchical prior and optimizing over hyperparameters. The RVM is a Bayesian kernel methods that choose sparse basis sets using an automatic relevance detection (ARD) (Bishop, 1995) style prior that pushes non-essential weights to zero. RVM has also the advantage of a probabilistic output.

### 6.2.2 Relevance Vector Machine

The RVM, introduced by Tipping (2001), is a sparse linear model. In this section, a brief introduction about RVM for classification problem is presented. Briefly, RVM is Bayesian approach for training a linear model. Consider a set of example of input vectors \( \{x_i\}_{i=1}^N \) is given along with a corresponding set of targets \( t_i \). For classification problem, \( t_i \) should be 0 for class \( C_1 \) and +1 for class \( C_2 \). The RVM constructs a logistic regression model based on a set of sequence features derived from the input patterns, i.e.

\[
p(C_1 | x) \approx \sigma(y(x; w)) \quad \text{Where} \quad y(x; w) = \sum_{i=1}^N w_i \Phi_i(x) + w_0
\]

(6.14)

Where basis function

\[
\Phi(x) = (\Phi_1(x), \Phi_2(x), ..., \Phi_N(x))^T = \left[ 1, K(x_i, x_1), K(x_i, x_2), ..., K(x_i, x_N) \right]^T,
\]

\( \mathbf{w} = (w_0, ..., w_N)^T \) are a vector of weights, \( \sigma(y) = \frac{1}{1 + \exp(-y)} \) is the logistic
sigmoid link function and \( \kappa(x_i, x_j) \) are kernel terms. Assuming a Bernoulli distribution for \( \mathbf{p}(t|x) \), the likelihood can be written as:

\[
P(t|w) = \prod_{i=1}^{N} \sigma(y_i; w)^{t_i} [1 - \sigma(y_i; w)]^{1-t_i}
\]

(6.15)

To form a Bayesian training criterion, we must also impose a prior distribution over the vector of model parameters or weights, \( p(w) \). The RVM adopts a separable Gaussian prior, with a distinct hyper-parameter, \( \alpha_i \), for each weight,

\[
p(w/\alpha) = \prod_{i=1}^{N} \mathcal{N}(w_i / 0, \alpha_i^{-1})
\]

(6.16)

The optimal parameters of the model are then given by the minimiser of the penalized negative log-likelihood,

\[
\log [P(t|w)p(w/\alpha)] = \sum_{i=1}^{N} \left[ t_i \log y_i + (1 - t_i) \log (1 - y_i) \right] - \frac{1}{2} w^T A w
\]

(6.17)

Where \( y_i = \sigma(y_i; w) \) and \( A = \text{diag}(\alpha) \) is a diagonal matrix with non-zero elements given by the vector of hyper-parameters. Next, Laplace’s method is used to obtain a Gaussian approximation to the posterior density of the weights,

\[
p(w/t, \alpha) \approx \mathcal{N}(w/\mu, \Sigma)
\]

(6.18)

Where the posterior mean and covariance are given by

\[
\mu = \sum \Phi^T B t \quad \text{and} \quad \Sigma = \left[ \Phi^T B \Phi + A \right]^{-1}
\]

(6.19)

Where \( B = \text{diag}(\beta_1, \beta_2, \ldots, \beta_N) \) is a diagonal matrix with \( \beta_n = \sigma(y(x_n)) [1 - \sigma(y(x_n))] \). The hyper-parameter are then updated in order to maximize their marginal likelihood, according to their efficient update formula.
\[
\alpha_i^{\text{new}} = \frac{1 - \alpha_i \sum_{ii}}{\mu_i^2}
\]  
(6.20)

Where \(\mu_i\) is the \(i^{th}\) posterior mean weight, \(\sum_{ii}\) is the \(i^{th}\) diagonal element of the posterior weight covariance, and the quantity is a measure of the degree to which the associated parameter \(w_i\) is determined by the data. This process is repeated until an appropriate convergence criterion is met. The outcome of this optimization is that many elements of \(\alpha\) go to infinity such that \(w\) will have only a few nonzero weights that will be considered as relevant vector.

### 6.3 RESULTS AND DISCUSSION

#### 6.3.1 Prediction of critical acceleration of slope

The most important parameter is seismic displacement of slope and sloping ground is computation of critical acceleration (\(k_y\)). This is an important parameter for the sliding block model like factor of safety in limit equilibrium method. The \(k_y\) value is interrelated to slope angle, \(\beta\) and FOS. The data base given by Ferentinou and Sakellariou (2007), for 63 slopes analyzed using cyclic failure mechanism. The data base consists of slope angle \(\beta\) and FOS as input and \(k_y\) as the output. The maximum, minimum, average standard deviation of the parameters used in this study is shown in Table 6.1. The results of different ANN models for the above data are shown in Figure 6.3, 6.4 and 6.5, for Bayesian regularization neural network (BRNN), Levenberg-Marquardt neural network (LMNN) and differential evolution neural network (DENN), respectively. The statistical performance in terms of correlation coefficient, coefficient of efficiency (E), root mean square error (RMSE) and maximum absolute error (MAE) of different ANN models are shown in Table 6.2. It can be observed that all the ANN models could predict the \(k_y\) value with high correlation coefficient and with negligible RMSE values. However, based
on R, E, RMSE and MAE values, BRNN found to be better than LMNN and DENN. Ferentinou and Sakellariou (2007) also reported similar findings. After obtaining $k_y$ values the displacement of slopes can be obtained using method suggested by Bray and Rathje (1998) or by applying ANN method (Ferentinou and Sakellariou 2007). Hence, in the present study such an effort has not been made. At this point, it may be emphasized that the ground displacement due to liquefaction induced is more catastrophic and complex than individual slopes. Hence, the study pertaining to prediction of liquefaction induced sloping ground is elaborated as follows.

Table 6.1 Parameters of the data considered for the present study

<table>
<thead>
<tr>
<th>Parameters</th>
<th>$\beta$</th>
<th>$F$</th>
<th>$k_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum</td>
<td>63.00</td>
<td>2.26</td>
<td>0.68</td>
</tr>
<tr>
<td>Minimum</td>
<td>34.00</td>
<td>1.22</td>
<td>0.10</td>
</tr>
<tr>
<td>Average</td>
<td>47.33</td>
<td>1.61</td>
<td>0.35</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>12.03</td>
<td>0.27</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Figure 6.3 The predicted and observed values $k_y$ for the BRNN model
Figure 6.4  The predicted and observed values $k_y$ for the LMNN model

Figure 6.5  The predicted and observed values $k_y$ for the DENN model
Table 6.2 Different ANN models and their statistical performances

<table>
<thead>
<tr>
<th>ANN Models</th>
<th>Correlation coefficient (R)</th>
<th>Coefficient of determination (E)</th>
<th>RMSE value</th>
<th>Maximum absolute error(MAE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
<td>Testing</td>
<td>Training</td>
<td>testing</td>
</tr>
<tr>
<td>BRNN</td>
<td>0.998</td>
<td>0.990</td>
<td>0.995</td>
<td>0.881</td>
</tr>
<tr>
<td>LMNN</td>
<td>0.995</td>
<td>0.987</td>
<td>0.990</td>
<td>0.873</td>
</tr>
<tr>
<td>DENN</td>
<td>0.990</td>
<td>0.987</td>
<td>0.979</td>
<td>0.865</td>
</tr>
</tbody>
</table>

6.3.2 Prediction of liquefaction induced ground displacement

As described in Chapter 2, presently the multi linear regression (MLR) model given by Youd et al. (2002) is most widely method for prediction of liquefaction induced lateral displacement. They proposed separate equations as per the geometry of the ground as (i) free face case and (ii) sloping ground, as shown in Figure 2.2b. It can be seen from the Wang and Rahman (1999) and Chiru-Danzer (2001) also developed separate ANN models for the same, but the above two studies are based on uncorrected field data given by Barlett and Youd (1995). Javadi et al. (2006) developed separate genetic programming (GP) model using the corrected data of Youd et al. (2002). As it can be seen from Figure 2.2b, the sloping ground model is valid for very mild slope (Youd et al. 2002) and the free face model can be simulated to that of settlement close to wide river embankment. Figure 6.6 is presented here to emphasize the same. The difference between the two types of ground condition is reflected in the geometry of ground surface at the location of the displacement vectors. As shown in Figure 6.6 the free face case is characterized by the free face ratio ($H_{ff}/L_{ff}$), where $H_{ff}$ is the height of the free face (i.e., difference between the crest and toe elevations) and $L_{ff}$ is the horizontal distance from the toe of free
face to the displacement vector. Hence, in the present study similar only free face models are developed using different ANN, SVM and RVM models.

Figure 6.6 Schematic Examples of free face liquefaction induced ground displacement

Data base and preprocessing

The case histories involving the lateral displacement towards a free face as given by Youd et al. (2002) have been considered. The database consisting of 228 cases are related to free face ground condition, out of which 182 are considered as training data set and remaining 46 for the validation (testing). The data base is normalized in the range [0,1] and the ANN, SVM and RVM are implemented using Matlab (Math Works 2001).

The performance of different ANN (LMNN, BRNN, and DENN) models as discussed in Chapter 3, are presented in Table 6.3. It can be seen that based on R value of the testing data results BRNN model is found to be better than the other model. Similarly, coefficient of determination, E also is found to be more for BRNN model. The RMSE and MAE values are found to be minimum for the BRNN model. The RMSE value indicates the variation in total data base but MAE value indicate the error value for individual data. The performance of the training and testing data of the developed BRNN model is found not to be that efficient like GP model developed by Javadi et al. (2007). However, as the exact data points that considered for training and testing by Javadi et al. (2007) is known, the results are not shown in Table 6.3
Table 6.3 Different ANN models and their statistical performances to predict the horizontal ground displacement of free face model

<table>
<thead>
<tr>
<th>ANN Models</th>
<th>Correlation coefficient (R)</th>
<th>Coefficient of determination (E)</th>
<th>RMSE value</th>
<th>Maximum absolute error (MAE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
<td>Testing</td>
<td>Training</td>
<td>testing</td>
</tr>
<tr>
<td>BRNN</td>
<td>0.912</td>
<td>0.882</td>
<td>0.828</td>
<td>0.696</td>
</tr>
<tr>
<td>LMNN</td>
<td>0.924</td>
<td>0.868</td>
<td>0.855</td>
<td>0.674</td>
</tr>
<tr>
<td>DENN</td>
<td>0.964</td>
<td>0.872</td>
<td>0.756</td>
<td>0.674</td>
</tr>
</tbody>
</table>

The predicted and observed values of horizontal ground displacement are shown in Figure 6.7 and 6.8, for training and testing data set respectively. It can be seen that the model is efficient for ‘large’ displacement and the ratio of predicted to observed values are within 0.5:1-1:2. However, the data points are very much scattered for ‘small to moderate’ displacement. Similar observation has been made by Javadi et al. (2007), while using GP model.

![Graph showing predicted vs. measured horizontal displacement](image)

Figure 6.7 Measured Vs. predicted horizontal displacement for the free face model using training data set as per ANN models
Figure 6.8 Measured Vs. predicted horizontal displacement for the free face model using testing data set as per ANN models

The variation in MAE, AAE and RMSE value for the developed ANN models corresponding to training and testing data are shown in Figure 6.9 and 6.10 respectively. Based on the above values it may be concluded that among the ANN models BRNN model is more efficient compared to LMNN and DENN models.
As discussed above the performance of SVM models depend upon the optimum capacity factor C and optimum error insensitive zone ε and the type of kernel function used (Samui 2008). In the present study such a procedure was made and it was observed that SVM model with radial basis function (Gauss), kernel function found to be better than polynomial and spline kernel function. The statistical performance of the developed model is shown in Table 6.4. In this study SVM employs about 65.4% of the training data as support vector. It is worth to mention here that, these support vectors represent prototypical examples. The prototypical examples exhibit the essential features of the information content of the data, and thus are able to transform the input data into the specified target. This has the advantages in terms of sparsity, i.e. a significant number of weights (SVM model parameters) are zero, thereby producing compact and efficient models.

Similarly for the results of the final RVM model is also presented in Table 6.4. It can be seen that compared to SVM model the performance of the RVM model is better in terms of correlation coefficient (R) and coefficient of
determination (E) for both training and testing data and particularly for the testing data. In this study RVM employs only 35.5% of the training data as support vector in comparison to 65.4% as in SVM model.

Table 6.4 Statistical performances of SVM and RVM model to predict the horizontal ground displacement of free face model

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Training performance</th>
<th>Testing performance</th>
<th>Number of support vector</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Correlation coefficient (R)</td>
<td>Coefficient of determination (E)</td>
<td>Correlation coefficient (R)</td>
</tr>
<tr>
<td>SVM (Gauss, width=0.19)</td>
<td>0.920</td>
<td>0.937</td>
<td>0.906</td>
</tr>
<tr>
<td>RVM(0.001)</td>
<td>0.962</td>
<td>0.982</td>
<td>0.979</td>
</tr>
</tbody>
</table>

The predicted and observed values of horizontal ground displacement are shown in Figure 6.11 and 6.12, for training and testing data set respectively. It can be seen that RVM model is more efficient in comparison to SVM model. The R value for training and testing data are, 0.962 and 0.979 respectively. Similarly the coefficient of determination is 0.982 and 0.960, respectively for training and testing data. Unlike, ANN models, the SVM and RVM models are found to be equally efficient for all magnitudes of displacement and the ratio of predicted to observed values are within 0.5:1- 1:2. The above statement is valid for both training and testing data. This has implication in terms of developing different models for ‘small to moderate’ and ‘large’ displacement as per Javadi et al. (2007). The performance of the training and testing data of the developed RVM model is found to be more efficient than GP model developed by Javadi et al. (2007). However, as the exact data points considered for training and testing used by Javadi et al. (2007) is known, the results are not shown in Table 6.4
Figure 6.11 Measured Vs. predicted horizontal displacement for the free face model using training data set as per SVM and RVM models

Figure 6.12 Measured Vs. predicted horizontal displacement for the free face model using testing data set as per SVM and RVM models
The variation of error values MAE, AAE and RMSE due to SVM and RVM models are presented in Figure 13 and Figure 14 respectively, for training and testing data. It can be seen that both RVM and SVM models are efficient in comparison to all the ANN models as shown in Figure 6.9 and 6.10. However, RVM model is found to be more efficient in comparison to SVM model. It may be mentioned here that the efficiency of artificial intelligence techniques are problem dependant and hence, needs critical evaluation before its use in any specific problem.

Figure 6.13 Variation of MAE, AAE and RMSE for SVM and RVM model for the training data

Figure 6.14 Variation of MAE, AAE and RMSE for SVM and RVM model for the testing data
6.4 CONCLUSIONS

This chapter pertains to the study of prediction of critical acceleration (k_y) of slopes and liquefaction induced lateral spread of free face ground. Based on results of different models (BRNN, LMNN and DENN), it was observed that ANN models can predict the k_y value with high correlation coefficient (R value > 0.99) and hence more sophisticated SVM and RVM methods are not employed.

However, the performance of developed ANN models (BRNN, LMNN and DENN) for the prediction of liquefaction induced displacement for the free face model are found to have correlation coefficient values varying from 0.862 to 0.880. The models were found to be efficient for ‘large’ displacement and the ratio of predicted to observe values are within 0.5:1-1:2. However, the data points are very much scattered for ‘small to moderate’ displacement. The developed SVM and RVM models are found to be efficient in comparison to ANN models. The RVM model is found to be more efficient comparison to SVM model. The R value for training and testing data are, 0.962 and 0.979 respectively. Similarly the coefficients of determination values are 0.982 and 0.960, respectively for training and testing data. It may be mentioned here that the efficiency of artificial intelligence techniques are problem dependant and hence, needs critical evaluation before its use in any specific problem. Another important observation is made that, the developed RVM model is equally efficient in ‘large’ and ‘small to moderate’ displacement. Hence, there is no need to develop separate model depending upon the dimension of the displacement.