# Machine Learning based Predictive Model for Screening Mycobacterium Tuberculosis Transcriptional Regulatory Protein Inhibitors from High-Throughput Screening Dataset

Syed Asif Hassan
Department of Computer Science, Faculty of Computing
and Information Technology Rabigh (FCITR)
King Abdulaziz University,
Jeddah, Saudi Arabia

Tabrej Khan
Department of Information Sciences, Faculty of Computing
and Information Technology Rabigh (FCITR)
King Abdulaziz University,
Jeddah, Saudi Arabia

***Abstract*—In view of the essential role played by dosRS in the survival of Mycobacterium in the infected granuloma cells, dosRS transcriptional regulatory proteins were considered as a validated target for high throughput screening (HTS). However, the cost and time factor involved in screening large compound libraries are an important hurdle in identifying lead compounds. Therefore, the use of computational machine learning techniques to build a predictive model for screening putative drug-like molecule has gained significance. In this regard, a target-based predictive model using machine learning approaches was built to develop fast and efficient virtual screening procedures to screen anti-dosRS molecules. In the present study, we have used various structural and physiochemical attributes of compounds from HTS dataset to train and build a chemoinformatics predictive model based on four state-of-art supervised classifiers (Random forest, SMO, J48, and Naïve Bayes). The trained model was applied to test dataset for validating the robustness, accuracy, and sensitivity of the predictive model in screening active anti-dosRS molecules. The Cost-Sensitive Classifier (CSC) with Random Forest (RF) algorithm based predictive model showed a high sensitivity (100%) and specificity (83.13%) to identify active and inactive molecules, respectively from assay dataset (ID: 1159583). CSC-RF proved to more robust and efficient in classifying active molecule from an imbalanced dataset with highest Balancing Classification Rate (BCR) (91.57%) and maximum Area under the Curve (AUC) value (0.999).**

***Keywords*—Mycobacterium; dosRS-transcriptional regulatory proteins; High Throughput Screening (HTS); virtual screening; machine learning algorithms; classification; predictive chemoinformatics model***

## I. Introduction

Tuberculosis (TB) a highly infectious disease is caused by *Mycobacterium tuberculosis* (*Mtb*) which affects a substantial population across the globe and is among the top 10 causes of death especially in low and middle-income countries. As per latest World Health Organization (WHO) report, nearly10.4 million people were infected with *Mtb* and approximately two million death occurred due to TB in 2015 (which includes nearly half a million people immunocompromised with Human Immunodeficiency Virus (HIV)) [1]. Moreover, propagation and evolution of both Multidrug-Resistant (MDR) and extensively Drug-Resistant (XDR) species of *Mtb* across the globe have turned into a key problem in combating tuberculosis worldwide [2]. An estimated 480000 people have developed MDR-TB worldwide in 2015 [3]. Considering the prevalence of this epidemic around the world, there is a pressing necessity to identify novel efficient and fast hit identification approaches. Discovery and development of novel drug generally comprise of four steps: 1) identifying the target/Screening of molecule from the database; 2) hit identification (3) lead finding and optimization; 3) pre-clinical studies of the optimized lead molecule; and 4) clinical studies. Hit identification is of profound importance for the triumph of all drug discovery programs. In this regard, High Throughput Screening (HTS) has been routinely used for the screening of hit molecule in the most drug discovery protocols. The enormous time and cost involved in HTS is a major hurdle in the discovery and development of novel drugs [4]. Virtual screening methodologies, when compared to traditional experimental HTS are comparatively fast, efficient and cost-effective to screen active hit molecules from thousands of molecules from chemical libraries. Structure-based and ligand-based virtual screening protocols have been adapted to screen and prioritize active hit molecules during early-phase of drug discovery protocols [5]. Moreover, the virtual screening protocol could further be improved using faster and robust algorithm to screen active hit molecules from a huge chemical dataset with higher accuracy and sensitivity.

Machine learning (ML) methods are predominantly robust and effective algorithms. ML algorithms can make an intelligent decision on an independent dataset based on their ability to recognize and learn complex attributes from multi-dimensional bioactivity input data, therefore, they have been recently employed to screen hit molecule during the hit identification phase of drug discovery program [6]-[13]. Since the bioactivity data obtained through HTS provides the necessary attributes both in binary (active/inactive) and a numerical value (namely, IC50). Therefore, the ML algorithms can be trained with binary and numerical values of various attributes of bioactivity dataset to classify molecules as active and inactive from bioactivity data procured from experimental HTS. Recent studies have shown the application of ML algorithms to build predictive computational chemoinformatics model to classify molecule as active or inactive from the bioassay data available on public domain derived from HTS [14]-[16]. These HTS data derived from the biological activity of molecule screened against targets critical for the survival of infectious agents within host cells. The present objective of our study is to develop a binary predictive classification model for screening active anti-dosRS molecules using the fast and efficient ML algorithms. The classification algorithm based chemoinformatics models when subjected to *in silico* selection of novel hit against dosRS molecule from large compound libraries will definitely fast-track the anti-tubercular agents' discovery process. The structure of the paper is as follows: Section II describes the material and method employed in this article. Section III describes the obtained results and the required discussion for the same and Section IV provides the concluding remarks about the present work. An overview of the approach that is employed in this study is represented in Fig. 1.

## II. Materials and Methods

This section describes the data source, the techniques for molecular descriptors generation and pre-processing the biological dataset. It also presents the ML algorithms for model building and appending Cost-sensitive learning methodology. Moreover, this section also describes the model performance statistical evaluators of Weka used in evaluating the currently proposed classification model.

### A. Data Source

The HTS data for small molecule against dosRS activity (Assay ID: 1159583) was obtained from PubChem a chemical library of National Center for Biotechnology Information (NCBI) [17]. The HTS dataset was built based on the bioactivity of the small molecule against hypoxia-regulated (i.e., dosRS) fluorescent biosensor in Mycobacterium tuberculosis CDC1551 (hspX':: GFP) full-grown in Middlebrook 7H9 medium with a pH 7.0 and further screened using 384-well microtiter plates format. A Compound library consisting of 328,633 small molecules were screened for anti-dosRS activity. According to the protocol definition, molecules that showed > 50 % inhibition of both growth and fluorescence were considered as general inhibitors (active molecule) of dosRS. Moreover, the activity of the molecules under study was scaled from 100 to 0, the scaling values were derived from normalized percentage inhibition, with values 100 or more than 100 corresponding to 100% inhibition (active molecule) and zero or less than zero corresponding to no inhibition (inactive molecule). The Structure-Data File (SDF) of both active and inactive molecules were downloaded from https://pubchem.ncbi.nlm.nih.gov/bioassay/1159583#section=Top.

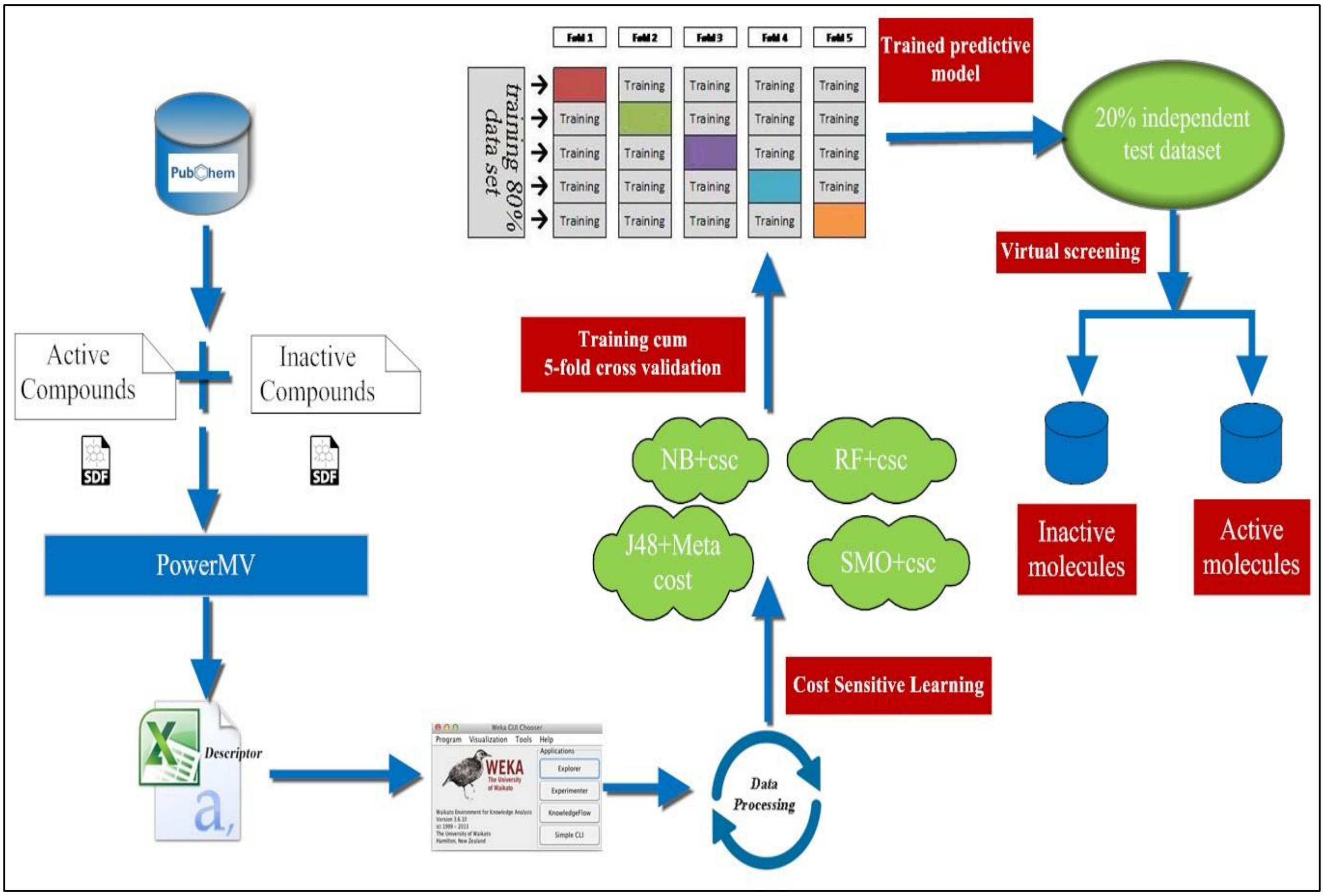

Fig. 1. Workflow for in silico virtual screening of active molecule from HTS dataset using a predictive classification model.

### B. *Molecular Descriptors Generation and Biological Dataset Pre-Processing*

PowerMV [18] publicly available windows based software was used for generating and viewing, 2D molecular descriptors from the available biological dataset (Assay ID: 1159583). Since the number of molecules to be screened in the dataset were large (approximately 3 million) and with a capacity of PowerMV to process a huge data is limited by available memory. Therefore, using a Perl script SplitSDFiles available in MayaChemTools [19], the entire biological HTS dataset was divided into smaller files. Consequently, each SDF files were successively uploaded and processed in PowerMV. Overall, 179 2D molecular descriptors representing the molecular attributes of each compound in the dataset AID1159583 were generated. Details of the complete list of various descriptors used for the construction of a predictive chemoinformatics model are provided in supplementary file 1 (Table III). The molecular descriptors files of each SDF subfile were joined into a lone Comma Separated Values (CSV) file. The bioactivity for each molecule in the dataset was appended to the last column labeled as “outcome” representing an additional feature “class” and a nominal value active or inactive was appended. The merged single CSV file of the descriptors was pre-processed to remove the non-informative or uninstructive descriptors having only zero and one-bit string all through the dataset were filtered out by using un-supervised filter present the Weka software tool [20]. Removal of the attributes having only one value throughout the dataset (uninstructive descriptors) decreased the size of the available bioactivity HTS dataset. Lastly, the instances of the bioactivity dataset were systematically arranged as per class and the processed data was split using a Perl script into 80 % training-cum validation set to build the classification model and 20 % as an independent test set to check the accuracy of the classification predictive model. In cross-validation (CV) a 5-fold CV is assigned to the training dataset for model generation. The processed descriptor file of the training cum validation set was randomly rearranged and split into “n” (here n=5) equal size folds. In successive iteration, one fold of the training cum validation dataset was used for testing and the remaining n-1folds for training the machine learning classifiers. An average of each test fold result was calculated. The mean test value provides cross-validated estimated accuracy of the proposed predictive chemoinformatics model. Finally, the present trained classification based predictive intelligent system was tried with 20 % separate test dataset comprising of molecule entirely unfamiliar to the trained intelligent system (proposed classification model). The test value obtained from 20 % independent test dataset provides the efficacy of the classification model to predict active molecule (inhibitors of dosRS) from an untrained dataset with higher accuracy and sensitivity.

### C. *Machine Learning Algorithms for Model Building*

An algorithm is a procedure to assign a specific class to a given input value. In this context, the classification algorithm based predictive model requires assigning a class (active /inactive) to input molecule well characterized by many Molecular attributes. In the present study, the classification based model was build using Weka work platform. The Weka platform which is a java based open source software required to implement classification and clustering algorithm for data analysis and visualization. In order to build a chemoinformatics classification model with higher accuracy and sensitivity to assign a class (active/inactive) to an independent set of test data, we compared the predictive efficiency of each of the four best-known ML classification algorithm such as J48, Naive Bayes (NB), Sequential minimal optimization (SMO), and Random Forest (RF). A brief description of the above-mentioned ML algorithms is mentioned below:

*1) Random Forest*: Random forest (RF) algorithm [21] is a collection of learning methods for categorization and that functions by generating a combination of decision trees during the training period. Arbitrary vector generated by arbitrarily choosing a subgroup of features to generate each tree. Once all the trees are generated, each tree in the ensemble chooses a class and the most voted class provides the final classification “class” for a given subset of attributes (i.e., individual tree). Random decision forests are fast as well as have the potential to handle huge input variables of the training set without over-fitting. The basic steps involved in the execution of the random forest algorithm are as follows:

```
Step 1: RF is a function with different parameter namely test, train,
min_size, max_depth, sample_size, n_features n_trees.

        def RF (test, train, min_size, max_depth, sample_size,
        n_features n_trees)

Step 2: Create list to store data in the form of tree structure

        trees = list()

Step 3: Start the iterative loop to generate a random sub-instances from the
dataset with substitution and build a decision tree and eventually generate
a prediction with a given content of bagged trees

      for i in range(n_trees):
        sample = subsample(train, sample_size)
        tree = build_tree(sample, n_features, min_size, max_depth)
        trees.append(tree)
      predictions = [bagging_predict(trees, row) for row in test]
    return (predictions)
```

*2) Naïve Bayes*: Naïve Bayes (NB) [22] relies on the assumption that each descriptor (attribute) in the processed training dataset is statistically independent. The NB classifier obtains from the training data, the conditional probability of all molecular features depending upon the class label. Classification is performed based on the principles of Bayes theorem which evaluate the possibility of an outcome happening based on the probability of a previous event. The likelihood of a compound to be either categorized into the active or inactive class is proportional to the percentage of molecules in one or the other class which has similar attribute value. The general likelihood of the activity (active/inactive) of a molecule is evaluated via multiplying their individual probabilities. NB algorithm is one of the simplest and effective classifiers. The NB algorithm can be explained as follows:

Suppose: The probability that a document “c” with vector y = < $y_1$,...,$y_n$> belongs to hypothesis h1 is:

$$P(\frac{h_1}{y_i}) = \frac{P(\frac{y_i}{h_1})P(h_1)}{P(\frac{y_i}{h_1})P(h_1)+P(\frac{y_i}{h_2})P(h_2)} \quad (1)$$

In this case, P(h1) is the previous probability linked with hypothesis h1, while P(h1|$y_i$) is a posterior probability.

Hence, now for “m” different hypotheses, we have

$$P(y) = \sum_{j=1}^{n} P(\frac{y_i}{h_j})P(h_j) \quad (2)$$

Therefore, we have

$$P(\frac{h_1}{x_i}) = \frac{P(\frac{x_i}{h_1})P(h_1)}{P(x_i)} \quad (3)$$

*3) J48*: The principle of C4.5 is a decision tree algorithm is implemented in J48 [23]. A decision tree model is created which moves in a top-down fashion from root to leaves by selecting an appropriate attribute at each decision node. The selection of an appropriate attribute at decision node helps us to decide which branch one should travel from any specific node. The leaf node in a decision tree specifies a class label. The functioning of the algorithm can be represented as follows:

```
Input: X //Training data
Output: Y //Decision tree
XYBUILD (*X)
{
Y=$\varphi$;
Y= Make a root node and tag the same using splitting parameters;
Y= for each split predicate augment arc to root node and name it;
 For each arc perform
X= Database made by implementing spreading out predicate to X;
If decision point is attained for this path, then
X’= generate a leaf node and tag it with suitable class;
Else
X’= XYBUILD(X);
Y= add Y’ to arc;
}
```

*4) Sequential Minimal Optimization (SMO)*: The algorithm SMO is applied for solving Quadratic Programming (QP) that arises during the training of Support Vector Machine (SVM) [24]. A hyperplane (i.e., SVM) divides members belonging to two distinct classes fairly apart from each other thus enabling proper classification. Contrary to the typical SVM that uses numerical QP optimization as an inner loop to solve large QP optimization problem, which arises all through the training of SVM with the training dataset. SMO breakdowns the large QP optimization case into minor QP case. These small QP cases are eventually resolved by SMO in an analytical manner. Therefore, SMO is comparatively is cost-effective in terms of computation time for solving large QP and also the ability to handle large dataset. The execution of SMO algorithm in sequential form is summarized as follow:

**Step 1:** Initialize $f_i = -y_i, \alpha_i = 0$, Dual=0, i=0, 1,…, $l$

Where $\boldsymbol{\alpha_i}$ is the Lagrange multiplier which needs optimization and

$$\boldsymbol{f_i} = \sum_{j=1}^{l} \alpha_j y_j k(X_j, X_i) - y_i \quad (4)$$

**Step 2:** Work out DualityGap, $b_{up}, I_{up}, b_{low}, I_{low}$

Where $b_{up} = \min\{f: i \in I_0 \cup I_1 \cup I_2\}$ (5)

Here $I_0, I_{1,}I_2$ denotes the guide of training data patterns

$$b_{low} = \max\{f: i \in I_0 \cup I_1 \cup I_2\},$$
$$I_{up} = arg.minf_i \ , \quad I_{low} = arg.maxf_i$$

DualityGap, representing the difference between the dual objective function and the primal

**Until *DualityGap* $\leq \boldsymbol{\tau}|\boldsymbol{Dual}|$**

*Where* $\tau = 10^{-6}$

***1.* Optimize $\boldsymbol{\alpha I_{up}, \alpha I_{low}}$ ;**

**2. Update $\boldsymbol{f_i}, \quad \boldsymbol{i = 1, \ldots, l}$;**

**3. Calculate$\boldsymbol{b_{up}, I_{up}, b_{low}, I_{low}}$, *DualityGap*** and update ***Dual*.**

**Repeat**

## *D. Cost-Sensitive Learning*

Cost-sensitive learning (CSL) is used to train classification model against imbalance class problem associated with HTS bioassay data. Imbalance class problem arises when at least one of the classes in a dataset are represented by much less number of instances when compared to others. In case of HTS biological data, the dataset is termed imbalance since the number of molecules that are active is less in number when compared to the number of inactive molecules. The minority class is represented by active compounds and while the majority class is associated with inactive compounds. This imbalance class problem in HTS dataset adds more complexity to the classification process [25]. Consequently, when machine learning classification algorithms are applied to imbalanced HTS biological dataset may result in biased prediction resulting in higher False Negative (FN) rate. Hence, many strategies in the past were offered and implemented to develop appropriate classification rules for class imbalance dataset. Since the interest of the present study was to correctly classify the minority class (True positives (TP)). Therefore, implementation of misclassification cost on FN instances makes the currently available original base classifiers cost-sensitive and enhances the TP predictive capability of the classifiers. There is no generalized rule for setting misclassification cost and is always subjective to user’s desired threshold. There are primarily two techniques of introducing misclassification cost in error-based base classifiers to overcome the problem of class imbalance problem in a given HTS biological dataset. The first method is to create either cost-sensitive classifier (CSC) namely Inexpensive Classification with Expensive Tests (ICET) [26] or decision tree algorithms proposed by Ling et al. [27] and the other method is to build a wrapper class which can convert the current available error-based base classifier into cost-sensitive one namely cost-sensitive classifier [28] and MetaCost [29]. The second method is generally referred to as meta-learning and is used in Weka software tool to introduce cost-sensitive learning in base classifiers. Meta-learning methods introduce a bias by setting a high misclassification cost for FN in the cost matrix C (a, b), here "a" is the real class and "b" is the anticipated class label. We have used meta-learning of Weka for implementing cost-sensitivity in the base classifier algorithms. MetaCost implements bagging iteration while reclassifying training data with minimum expected cost and eventually applying the base classifiers to the modified training dataset, to generate trustworthy probability calculation on the training dataset. This implementation works well for imbalance class problem associated with HTS biological dataset. On the other hand, CSC employs two measures to implement cost sensitivity: (1) prediction of classes with minimum expected misclassification cost and (2) the training data are reweighted depending upon the total cost associated with individual classes.

In the present study, we have applied meta-cost method were the unpruning option was set to true for implementing cost sensitivity in the J48 base classifier. On the other hand, we have used CSC with the *MinimizedExpectedCost* option set to be false for NB, RF, and SMO. While in SMO an additional option of buildlogisticmodels was employed. Previous studies have shown that these setting (J48-unpruning option-true and CSC-*minimize expected cost-false)* have given better accuracy with their corresponding cost-sensitive classifiers [30]-[32]. A 2x2 (for the binary class problem) cost matrix was used for implementing cost-sensitivity in base classifiers. The four sections of the 2x2 Weka cost matrix are: (1) True Positive (TP) – inhibitor compound of HTS dataset accurately predicted as active; (2) False Positive (FP) – non-inhibitor (inactive) compound of HTS dataset falsely anticipated as active compound; (3) True Negative (TN) – non-inhibitor (inactive) molecule of HTS dataset appropriately predicted as inactive; (4) False Negative (FN) – Inhibitor (active) molecules of HTS dataset inaccurately anticipated as inactive. Considering our case, if the inhibitors (TP) of dosRS are incorrectly classified as inactive molecule (FN) that is more expensive when compared to non-inhibitors (TN) of dosRS classified as inhibitors (FP). Therefore, the fraction of FN is considered more important than the fraction of FP during the development of the classification model and the misclassification cost has been implemented upon FN. Increasing the misclassification cost for FN would enable an enhancement in the number of both TP and FP, respectively. For maintaining the percentage of FP under check, we limit the FP rate to ≤ 20 %. Until the limit for FP is reached we can increase the misclassification cost for FN such that maximum number of TP (inhibitors of dosRS) are predicted.

### E. Model Performances Estimation

To estimate the performance of the classification model, various statistical performance evaluators were used to estimate our results. The fraction of predicted true positive (active molecule) to the total number of the active molecule (TP/TP+FN) is designated as True Positive Rate (TPR). Similarly, the fraction of projected false actives (FP) to a real number of inactive molecules (FP/FP+TN) is termed as False Positive Rate (FPR). Specificity (TN/TN+FP) is the ability of the classification model to screen non-inhibitors compounds predicted as true negative and false positives while sensitivity is calculated as (TP/TP+FN) which demonstrate the ability of the model to screen inhibitors (active) compounds predicted as True Positives and False Negatives. A test evaluation showing higher sensitivity and specificity values always have a minimum error percentage. Accuracy specifies the overall nearness of measured test value to its factual value. In this case, accuracy is the overall evaluation of correctly predicted active and inactive molecule from an independent test dataset. It is generally estimated as ([TP + TN] / [TP+TN+FP+FN]). Balance Classifier Rate (BCR) calculated as an average of specificity and sensitivity (0.5x (sensitivity + specificity)) and the observed BCR values provides a stable accuracy while classifying a class bias dataset. Receiver Operating Characteristic (ROC) plot is used to assess the reliability of a classifier by using the Area under the Curve (AUC) value. The AUC values are obtained by plotting a graph between the False Positive Rate (FPR) plotted in the "x" axis and True Positive Rate (TPR) in "y" axis. AUC value is a probability that a classifier will give a greater score to a randomly chosen positive instance (active molecule) as compared to a randomly chosen negative instance (inactive molecule).

## III. Results and Discussion

A confirmatory high throughput screen bioassay dataset (AID 1159583) performed to screen active dosRS inhibitors. The AID 1159583 dataset containing 312 active 300891 inactive were used to generate 179 molecular descriptors using PowerMV (supplementary file 1). A set of twenty-five noninformative molecular descriptors were deleted during the preprocessing of the dataset as mentioned in prior in the methodology section. Finally, the remaining 154 molecular descriptors were employed for building classification based model. The descriptors deleted during the preprocessing of the dataset are enlisted in supplementary file 1. Consequently, after preprocessing the dataset was divided into 20% independent test data and the remaining 80% of the dataset was employed for training-cum validation. The test and the training dataset were transformed into *Attribute-Relation File Format (*arff) using Weka. Initially, the training data in .arff format was loaded and processed in Weka. Since the training dataset file was large therefore a heap size of 8 GB was used to initiate the processing of the dataset in Weka. Firstly, standard base classifiers were employed to build the classification based predictive model. The predictive models built using base classification algorithm had a low number of TP due to the imbalanced nature of HTS dataset where the base classifier showed a preference for the majority class i.e., inactive compounds. Therefore, cost-sensitive learning was introduced using cost matrix where misclassification cost for FN was raised keeping the false positive rate under a threshold limit of ≤ 20%. Thus a number of classification model was trained with incremental FN cost. The FN misclassification cost of the best-trained model for each classifier is tabulated in Table I. As per Table I, the misclassification cost appended for FN to increase the instances of TP's keeping the FP under threshold (i.e., ≤ 20%) was minimum for NB-CSC as compared to other cost-sensitive base classifiers.

TABLE I. MISCLASSIFICATION COST ASSIGNED TO FALSE NEGATIVE (FN) FOR EACH COST-SENSITIVE CLASSIFIER (CSC)

| Cost-Sensitive Classifier (CSC) | Misclassification Cost on False Negative (FN) |
|---|---|
| *Random Forest (RF-CSC)* | 212315 |
| *J48-MetaCost* | 3228 |
| *Naïve Bayes (NB-CSC)* | 270 |

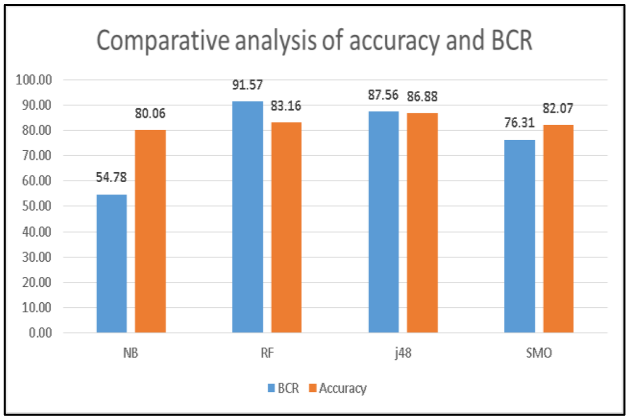

Fig. 2. Comparative study of Balanced Classification Rate (BCR) and Accuracy (Ac) of each cost-sensitive classifier based model.

In our present study, we observed that NB was able to build classification model at a faster rate as compared with another cost sensitive base classifier. The best-trained model for each cost-sensitive classifier was evaluated on 20% independent test dataset with different statistical evaluators of Weka. All the best-trained models had a controlled rate of FP instances (i.e., within 20% of the total number of molecules tested). The performance statistics of the best-predicted model for each cost sensitive base classifier on independent test data are tabulated in Table II. Due to the class imbalance nature of the dataset, the overall accuracies alone which were above 80% for all the four cost-sensitive classifier may not be sufficient to measure the efficacy of the classification model. Therefore, Balanced Classification Rate (BCR) another model performance measure was used to evaluate the robustness and efficacy of the model. BCR provides stability to the classification model by calculating the mean of specificity and sensitivity. As shown in Fig. 2, BCR value is highest RF-CSC as compared to all other cost-sensitive classifiers.

A measure of specificity and sensitivity was used to access the ability of the classification model to accurately predict the actual biological activity of the molecule i.e., actual positive and negative instances in the dataset. An ideal classification model is a system which achieves 100% specificity and sensitivity, respectively. As shown in Fig. 3 all the cost-sensitive classifier were highly specific in predicting negative results (predictive specificity ≥ 80%) and in terms of sensitivity random forest-CSC was found to be an ideal cost-sensitive classifier with a predictive measure of 100%, while Naïve Bayes-CSC had the lowest sensitive percentage among all the classifier used in the current study.

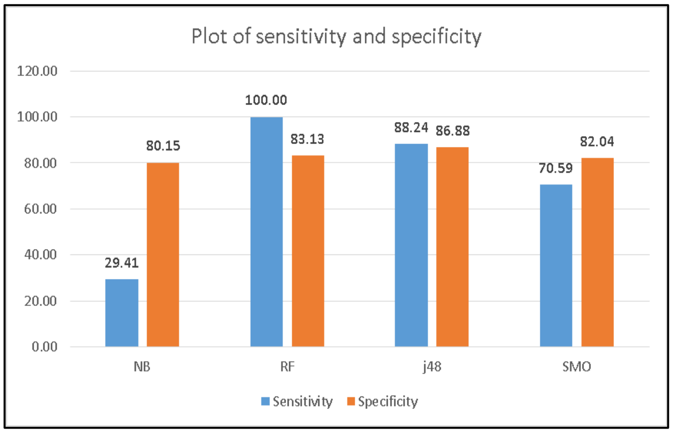

Fig. 3. Comparative study of sensitivity (Sn) and Specificity (Sp) of each cost-sensitive classifier based model.

Evaluation of the discriminatory power of the predictive model using AUC value was generated by drawing a Receiver Operating Characteristic (ROC) plotted between FP rate and TP rate as shown in Fig. 4.

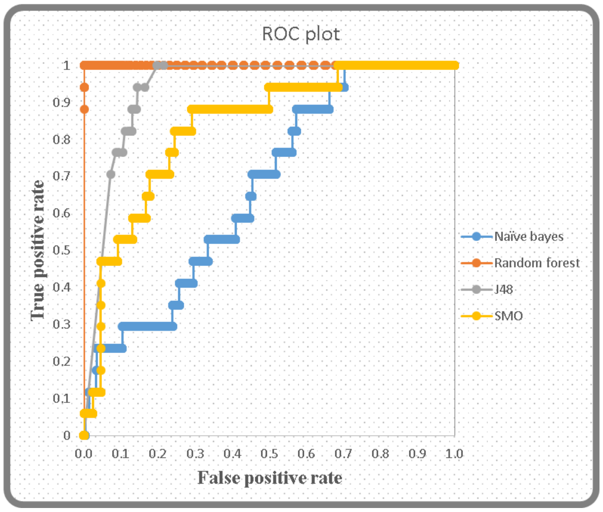

Fig. 4. Receiver Operating Characteristic (ROC) curve plot represent the significant Area Under the curve (AUC) values for Random Forest (RF), J48, Naïve Bayes (NB), and Sequential Minimal Optimization (SMO).

TABLE II. THE PERFORMANCE STATISTICS OF COST-SENSITIVE CLASSIFIERS TESTED ON 20% INDEPENDENT TEST DATASET

| Classifier | TN % | TP % | FP % | | FN % | Ac | ROC | Sn | Sp | BCR |
|---|---|---|---|---|---|---|---|---|---|---|
| *J48-Metacost* | 86.9 | 88.2 | 13.1 | | 11.8 | 86.9 | 0.937 | 88.24 | 86.88 | 87.56 |
| *RF-CSC* | 83.1 | 100 | 16.9 | | 0.0 | 83.2 | 0.999 | 100 | 83.13 | 91.57 |
| *NB-CSC* | 80.2 | 29.4 | 19.8 | | 70.6 | 80.1 | 0.667 | 29.41 | 80.15 | 54.78 |
| *SMO-CSC* | 82.0 | 70.6 | 18 | | 29.4 | 82.0 | 0.736 | 70.59 | 82.04 | 76.31 |

Where TN = True Negative; TP = True positive; FP = False Positive; FN = False Negative; Ac = Accuracy; ROC = Receiver Operating Characteristic; Sn = Sensitivity; Sp = Specificity BCR= Balanced Classification Rate.

The analysis of ROC curve is an appropriate and consistent method for evaluating the relative classifier performance in virtual screening using predictive classification model. The ROC curve analysis shows that random forest spread over a maximum area under the curve (AUC value: 0.999) as compared to other classifiers. In data analytics, an AUC value which is nearer to 1 is considered important. Though all the model based on four states of art classifiers were observed to have equivalent predictive accuracy, random forest-CSC proved to more efficient among all with high specificity, sensitivity, highest BCR rate and maximum AUC value.

The basic idea of performing simulated screening protocol is to acquire a substantial number of true positive (active molecules) from a chemical dataset of variable sizes. To evaluate the enrichment for TP obtained by using *in-silico* screening based on predictive classification model, Enrichment Factor (EF) was calculated on a dataset of variable sizes. Generally, EFs are calculated at 1%, 2%, 5% and 10% of the dataset to be screened. The EF with random forest-CSC was found to be 3.5 (EF 1%), 4.6 (EF 2%), 3.4 (EF 5%) and 3.2 (EF 10%). These EF values show that our best classification model (random forest) could achieve an enrichment of 3-4 folds for TP's as compared with any random screening protocol. Therefore, Random Forest is suggested to be a reliable and efficient classifier for screening inhibitors of dosRS from HTS dataset.

## IV. Conclusion and Future Scope

In this study, we have shown that ML algorithms can be effectively used to construct a supervised classification model for screening inhibitors (active molecule) of dosRS from the publicly available chemical compound dataset. Comparative study of various statistical performance evaluators on four important base classifiers such as Random Forest, Naïve Bayes, SMO, and J48 show that random forest shows the highest sensitivity, BCR rate, and AUC value, thus RF is statistically efficient in screening active molecule (inhibitor of dosRS) from the independent imbalance chemical dataset. This study also suggests through a supervised cost-sensitive machine learning algorithm i.e., Random forest, in this case, can cause 3-4 folds enrichment of screening true positives (active inhibitors of dosRS) from large chemical libraries (dataset). Therefore, the future scope of the current proposed cost-sensitive learning-based model can be employed in virtual screening approaches which will speed up the target based anti-tubercular drug discovery program against tuberculosis. Further, substructure analysis of the screened lead chemical molecules can be performed to identify important substructure which would allow us to screen more potent inhibitors of dosRS target macromolecule from chemical libraries will be implemented in future studies.

## Acknowledgement

We are thankful to the Faculty of Computing and Information Technology Rabigh (FCITR) of King Abdulaziz University, Jeddah for providing necessary software, server and platform to perform our experiments.

## References

[1] World Health Organization. Communicable Diseases Cluster. Stop TB Department., Towards universal access to diagnosis and treatment of multidrug-resistant and extensively drug-resistant tuberculosis by 2015 : WHO progress report 2011. World Health Organization, 2011.

[2] M. D. Iseman, "Evolution of drug-resistant tuberculosis: a tale of two species.," Proc. Natl. Acad. Sci. U. S. A., vol. 91, no. 7, pp. 2428–2429, Mar. 1994.

[3] "WHO | Towards universal access to diagnosis and treatment of multidrug-resistant and extensively drug-resistant tuberculosis by 2015," WHO, 2015.

[4] Lahana, "How many leads from HTS?," Drug Discov. Today, vol. 4, no. 10, pp. 447–448, Oct. 1999.

[5] B. Waszkowycz, T. D. J. Perkins, R. A. Sykes, and J. Li, "Large-scale virtual screening for discovering leads in the postgenomic era," IBM Syst. J., vol. 40, no. 2, pp. 360–376, 2001.

[6] J.-P. Vert and L. Jacob, "Machine Learning for In Silico Virtual Screening and Chemical Genomics: New Strategies," Comb. Chem. High Throughput Screen., vol. 11, no. 8, pp. 677–685, Sep. 2008.

[7] 6. J. Melville, E. Burke, and J. Hirst, "Machine Learning in Virtual Screening," Comb. Chem. High Throughput Screen., vol. 12, no. 4, pp. 332–343, May 2009.

[8] P. Vasanthanathan, O. Taboureau, C. Oostenbrink, N. P. E. Vermeulen, L. Olsen, and F. S. Jorgensen, "Classification of Cytochrome P450 1A2 Inhibitors and Noninhibitors by Machine Learning Techniques," Drug Metab. Dispos., vol. 37, no. 3, pp. 658–664, Mar. 2009.

[9] R. Lowe, R. C. Glen, and J. B. O. Mitchell, "Predicting Phospholipidosis Using Machine Learning," Mol. Pharm., vol. 7, no. 5, pp. 1708–1714, Oct. 2010.

[10] E. March-Vila, L. Pinzi, N. Sturm, A. Tinivella, O. Engkvist, H. Chen, and G. Rastelli, "On the Integration of In Silico Drug Design Methods for Drug Repurposing," Front. Pharmacol., vol. 8, p. 298, 2017.

[11] M. Wójcikowski, P. J. Ballester, and P. Siedlecki, "Performance of machine-learning scoring functions in structure-based virtual screening," Sci. Rep., vol. 7, p. 46710, Apr. 2017.

[12] M. Wang, P. Li, and P. Qiao, "The Virtual Screening of the Drug Protein with a Few Crystal Structures Based on the Adaboost-SVM," Comput. Math. Methods Med., vol. 2016, pp. 1–9, 2016.

[13] P. B. Jayaraj, M. K. Ajay, M. Nufail, G. Gopakumar, and U. C. A. Jaleel, "GPURFSCREEN: a GPU based virtual screening tool using random forest classifier," J. Cheminform., vol. 8, no. 1, p. 12, Dec. 2016.

[14] C. Schierz, "Virtual screening of bioassay data," J. Cheminform., vol. 1, no. 1, p. 21, 2009.

[15] Q. Li, T. Cheng, Y. Wang, and S. H. Bryant, "PubChem as a public resource for drug discovery," Drug Discov. Today, vol. 15, no. 23–24, pp. 1052–1057, Dec. 2010.

[16] B. Chen and D. J. Wild, "PubChem BioAssays as a data source for predictive models," J. Mol. Graph. Model., vol. 28, no. 5, pp. 420–426, Jan. 2010.

[17] National Center for Biotechnology Information. PubChem BioAssay Database; AID=1159583, https://pubchem.ncbi.nlm.nih.gov/bioassay/1159583.

[18] K. Liu, J. Feng, and S. S. Young, "PowerMV: A Software Environment for Molecular Viewing, Descriptor Generation, Data Analysis and Hit Evaluation," J. Chem. Inf. Model., vol. 45, no. 2, pp. 515–522, Mar. 2005.

[19] M. Sud, "MayaChemTools." 2010.

[20] R. R. Bouckaert, E. Frank, M. a Hall, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "WEKA - Experiences with a Java Open Source Project," J. Mach. Learn. Res., vol. 11, pp. 2533–2541, 2010.

[21] R. E. Schapire, L. Breiman, and R. E. Schapire, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[22] N. Friedman, "Bayesian Network Classifiers," Mach. Learn., vol. 163, no. 29, pp. 131–163, 1997.

[23] J. R. Quinlan, "C4. 5: Programs for machine learning Morgan Kaufmann Publishers San Francisco," CA Google Sch., 1993.

[24] J. Platt, “Sequential minimal optimization: A fast algorithm for training support vector machines,” 1998.

[25] N. Japkowicz, “The class imbalance problem: Significance and strategies,” in Proc. of the Int’l Conf. on Artificial Intelligence, 2000.

[26] P. D. Turney, “Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm,” J. Artif. Intell. Res., vol. 2, pp. 369–409, 1995.

[27] C. X. Ling, Q. Yang, J. Wang, and S. Zhang, “Decision trees with minimal costs,” in Proceedings of the twenty-first international conference on Machine learning, 2004, p. 69.

[28] H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.

[29] P. Domingos, “Metacost: A general method for making classifiers cost-sensitive,” in Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, 1999, pp. 155–164.

[30] S. Jamal, V. Periwal, V. Scaria, O. Consortium, and others, “Computational analysis and predictive modeling of small molecule modulators of microRNA,” J. Cheminform., vol. 4, no. 1, p. 16, 2012.

[31] V. Periwal, S. Kishtapuram, and V. Scaria, “Computational models for in-vitro anti-tubercular activity of molecules based on high-throughput chemical biology screening datasets,” BMC Pharmacol., vol. 12, no. 1, p. 1, 2012.

[32] H. Kaur, M. Ahmad, and V. Scaria, “Computational Analysis and In silico Predictive Modeling for Inhibitors of PhoP Regulon in S. typhi on High-Throughput Screening Bioassay Dataset,” Interdiscip. Sci. Comput. Life Sci., vol. 8, no. 1, pp. 95–101, 2016.

## SUPPLEMENTARY FILE 1

TABLE III. DIVISION OF MOLECULAR DESCRIPTORS CALCULATED FOR THE DATASET (ID: 1159583)

| **Sl. No.** | **Molecular Descriptor category*** | **Number of molecular descriptors generated** | **Molecular Descriptors prior to data processing** | | | **Molecular Descriptors deleted after data processing** |
|---|---|---|---|---|---|---|
| 1. | ***Pharmacophore fingerprints*** | 147 | NEG_01_NEG – NEG_07_NEG<br>NEG_01_POS – NEG_07_POS<br>NEG_01_HBD – NEG_07_HBD<br>NEG_01_HBA – NEG_07_HBA<br>NEG_01_ARC – NEG_07_ARC<br>NEG_01_HYP – NEG_07_HYP<br>POS_01_POS – POS_07_POS<br>POS_01_HBD – POS_07_HBD<br>POS_01_HBA – POS_07_HBA<br>POS_01_ARC – POS_07_ARC<br>POS_01_HYP – POS_07_HYP<br>HBD_01_HBD – HBD_07_HBD<br>HBD_01_HBA – HBD_07_HBA<br>HBD_01_ARC – HBD_07_ARC<br>HBD_01_HYP – HBD_07_HYP<br>HBA_01_HBA – HBA_07_HBA<br>HBA_01_ARC – HBA_07_ARC<br>HBA_01_HYP – HBA_07_HYP<br>HYP_01_HYP – HYP_07_HYP | | | NEG_01_POS<br>NEG_02_POS<br>NEG_01_HBA<br>NEG_02_HBA<br>NEG_01_ARC<br>NEG_01_HYP<br>POS_01_POS<br>POS_02_POS<br>POS_01_HBD<br>POS_01_HBA<br>POS_02_HBA<br>POS_01_ARC<br>POS_01_HYP<br>HBD_01_HBD<br>HBD_02_HBD<br>HBD_01_HBA<br>HBD_02_HBA<br>HBD_01_ARC<br>HBD_01_HYP<br>HBA_01_HBA<br>HBA_02_HBA<br>HBA_01_ARC<br>HBA_02_ARC<br>HBA_01_HYP<br>ARC_01_HYP |
| 2. | ***Weighted Burden Number*** | 24 | WBN_GC_L_0.25<br>WBN_GC_H_0.25<br>WBN_GC_L_0.50<br>WBN_GC_H_0.50<br>WBN_GC_L_0.75<br>WBN_GC_H_0.75<br>WBN_GC_L_1.00<br>WBN_GC_H_1.00 | WBN_EN_L_0.25<br>WBN_EN_H_0.25<br>WBN_EN_L_0.50<br>WBN_EN_H_0.50<br>WBN_EN_L_0.75<br>WBN_EN_H_0.75<br>WBN_EN_L_1.00<br>WBN_EN_H_1.00 | WBN_LP_L_0.25<br>WBN_LP_H_0.25<br>WBN_LP_L_0.50<br>WBN_LP_H_0.50<br>WBN_LP_L_0.75<br>WBN_LP_H_0.75<br>WBN_LP_L_1.00<br>WBN_LP_H_1.00 | None |
| 3. | ***Properties*** | 8 | XLogP, PSA, NumRot, NumHBA, NumHBD, MW, BBB, BadGroup | | | None |