

Article

# Bioactive Molecule Prediction Using Extreme Gradient Boosting

Ismail Babajide Mustapha <sup>1</sup> and Faisal Saeed <sup>2,\*</sup>

<sup>1</sup> UTM Big Data Centre, Ibnu Sina Institute for Scientific and Industrial Research, Universiti Teknologi Malaysia, Skudai, Johor 81310, Malaysia; bmismail2@live.utm.my

<sup>2</sup> Information Systems Department, Faculty of Computing, Universiti Teknologi Malaysia, Skudai, Johor 81310, Malaysia

\* Correspondence: faisalsaeed@utm.my; Tel.: +60-7-5532-406

Academic Editor: Leif A. Eriksson

Received: 1 May 2016; Accepted: 22 July 2016; Published: 28 July 2016

**Abstract:** Following the explosive growth in chemical and biological data, the shift from traditional methods of drug discovery to computer-aided means has made data mining and machine learning methods integral parts of today's drug discovery process. In this paper, extreme gradient boosting (Xgboost), which is an ensemble of Classification and Regression Tree (CART) and a variant of the Gradient Boosting Machine, was investigated for the prediction of biological activity based on quantitative description of the compound's molecular structure. Seven datasets, well known in the literature were used in this paper and experimental results show that Xgboost can outperform machine learning algorithms like Random Forest (RF), Support Vector Machines (LSVM), Radial Basis Function Neural Network (RBFN) and Naïve Bayes (NB) for the prediction of biological activities. In addition to its ability to detect minority activity classes in highly imbalanced datasets, it showed remarkable performance on both high and low diversity datasets.

**Keywords:** biological data; drug discovery; virtual screening; prediction of biological activity

## 1. Introduction

Recent advancement in technology has been crucial to the explosive growth in the amount of chemical and biological data available in the public domain. Hence, data driven drug discovery and development process has attracted increased research interest in the last decade with a view not only to design and analyze but apply effective learning methodologies to the rapidly growing data. By leveraging one of the important principles of chemical/molecular similarity [1], where similar biological activities and properties are expected of structurally similar compounds, approaches to drug design through screening of large chemical databases have increased over the years. Virtual Screening (VS), the use of computational approaches and tools through the search of large databases for target or activity prediction, has notably witnessed a shift in trend from the traditional similarity searching, through reference compounds, to the use of machine learning tools to learn from the massive big data by training and prediction of unknown activity. In particular, the compound classification, in which compound label prediction is based on knowledge acquired from a training set, has gained increased research interest and many machine learning tools have been proposed to exploit the increasing big data in drug discovery. Support Vector Machines (SVM) [2,3], DT [4], Random Forest [5], K Nearest Neighbors (K-NN) [6], Naïve Bayes Classifier [7] and Artificial Neural Networks (ANN) [8] are some of the most popular machine learning methods used for activity prediction in compound classification [9].

Despite records of successful application of these methods in cheminformatics and computer aided drug discovery, each method has its peculiar shortcomings and practical constraints; such as predictive accuracy, robustness to high dimensionality and irrelevant descriptors, model interpretability, and

computational efficiency, that hinders its optimal performance. For example, DT is a method that performs fairly well when it comes to most of the afore-stated criteria; however, its low predictive accuracy has inspired methods involving an ensemble of trees to improve this shortcoming. One of such efforts produced Random Forest which has been shown to be a reliable machine learning tool for compound classification as reported in [5]. In the same vein, while bearing in mind the No free Lunch Theorem [10]; that there is no best algorithm for all problems, we present herein the findings on another impressive ensemble of tree method called Extreme Gradient Boosting (Xgboost) for bioactive molecule prediction.

Xgboost is an efficient and scalable variant of the Gradient Boosting Machine (GBM) [11] which has been a winning tool for several Machine learning competitions [12,13] in recent years due to its features such as ease of use, ease of parallelization and impressive predictive accuracy. In addition to the obvious fact that alternative approaches to target prediction gives a wider perspective of the data rather than a single approach [14], we show in this paper that Xgboost not only produces comparable or even better predictive accuracy than the state of art in bioactivity prediction, but possess the intrinsic ability to handle the highly diverse and complex feature space of descriptors, especially in situations where the class distribution is highly imbalanced.

## 2. Methods

### 2.1. Tree Ensemble

As described by Chen and Guestrin [15], Xgboost is an ensemble of  $K$  Classification and Regression Trees (CART)  $\{T_1(x_i, y_i) \dots T_K(x_i, y_i)\}$  where  $x_i$  is the given training set of descriptors associated with a molecule to predict the class label,  $y_i$ . Given that a CART assigns a real score to each leaves (outcome or target), the prediction scores for individual CART is summed up to get the final score and evaluated through  $K$  additive functions, as shown in Equation (1):

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in F \quad (1)$$

where  $f_k$  represents an independent tree structure with leaf scores and  $F$  is the space of all CART. The regularized objective to optimize is given by Equation (2):

$$Obj(\Theta) = \sum_i^n l(y_i, \hat{y}_i) + \sum_k^K \Omega(f_k) \quad (2)$$

The first term is a differentiable loss function,  $l$ , which measures the difference between the predicted  $\hat{y}$  and the target  $y_i$ . The second is a regularization term  $\Omega$  which penalizes the complexity of the model to avoid over-fitting. It is given by  $\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$  Where  $T$  and  $w$  are the number of leaves and the score on each leaf respectively.  $\gamma$  and  $\lambda$  are constants to control the degree of regularization. Apart from the use of regularization, shrinkage and descriptor subsampling are two additional techniques used to prevent overfitting [15].

**Training.** For a training dataset of molecules with vectors of descriptors and their corresponding class labels or (e.g., active/inactive) or activity of interest, the training procedure in Xgboost is summarized as follows;

- i For each descriptor,
  - Sort the numbers
  - Scan the best splitting point (lowest gain)
- ii Choose the descriptor with the best splitting point that optimizes the training objective
- iii Continue splitting (as in (i) and (ii)) until the specified maximum tree depth is reached

- iv Assign prediction score to the leaves and prune all negative nodes (nodes with negative gains) in a bottom-up order
- v Repeat the above steps in an additive manner until the specified number of rounds (trees K) is reached.

Since additive training is used, the prediction  $\hat{y}$  at step  $t$  expressed as

$$\hat{y}_i^{(t)} = \sum_{k=1}^K f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \quad (3)$$

And Equation (2) can be written as

$$Obj(\Theta)^{(t)} = \sum_i^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (4)$$

And more generally by taking the Taylors expansion of the loss function to the second order

$$Obj(\Theta)^{(t)} = \sum_{i=1}^n [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) + \Omega(f_t)] \quad (5)$$

where  $g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$  and  $h_i = \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)})$  are respectively first and second order statistics on the loss function. A simplified objective function without constants at step  $t$  is as follows

$$Obj(\Theta)^{(t)} = \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \quad (6)$$

The objective function can be written by expanding the regularization term as

$$\begin{aligned} Obj(\Theta)^{(t)} &= \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\ &= \sum_{j=1}^T [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \end{aligned} \quad (7)$$

where  $I_j = \{i | q(x_i) = j\}$  is the instance set of leaf  $j$ , for a given structure  $q(x)$  the optimal leaf weight,  $w_j^*$ , and the optimal objective function which measure how good the structure is are given by Equations (8) and (9) respectively

$$w_j^* = -\frac{G_j}{H_j + \lambda} \quad (8)$$

$$Obj^* = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T \quad (9)$$

where  $G_j = \sum_{i \in I_j} g_i$ ,  $G_j = \sum_{i \in I_j} g_i$  and  $H_j = \sum_{i \in I_j} h_i$ .

Equation (10) is used to score a leaf node during splitting. The first, second and third term of the equation stands for the score on the left, right and the original leaf respectively. Moreover, the final term,  $\gamma$ , is regularization on the additional leaf.

$$Gain = \frac{1}{2} \left[ \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (10)$$

## 2.2. Machine Learning Algorithms

The performance of Xgboost was compared with four machine learning algorithms that have been used in the previous studies for activity prediction (Lavecchia 2015): The Support Vector Machine LibSVM (LSVM) [16], Random Forest (RF) [5], Naïve Bayes (NB) [17], and the Radial Basis Function Network (RBFN) [18] Classifiers.

## 3. Experimental Design

### 3.1. Datasets

This work was evaluated on seven carefully selected datasets that have been used to validate fingerprint based molecule classification and activity prediction in the past. A description of COX2 cyclooxygenase-2 inhibitors (COX2) (467 samples), benzodiazepine receptor (BZR) (405 samples) and estrogen receptor (ER) (393 samples) datasets [19,20] is shown in Table 1. The compounds are classified as active or inactive, and divided into training (70%) and validation (30%) sets for the purpose of this work. The table shows the mean pairwise Tanimoto similarity that was calculated based on ECFC\_4 across all pairs of molecules for both active and inactive molecules.

**Table 1.** Activity Classes for cyclooxygenase-2 (COX2) estrogen receptor (ER) and benzodiazepine receptor (BZR) Datasets.

Datasets	Number of Compounds				Pairwise Similarity (Mean)	
	Active		Inactive		Active	Inactive
	Training	Validation	Training	Validation		
Cyclooxygenase-2 inhibitors	211	92	116	48	0.687	0.690
Benzodiazepine receptor	214	92	70	29	0.536	0.538
Estrogen receptor	86	55	190	62	0.468	0.456

The fourth dataset utilized as a part of this study is Directory of Useful Decoys (DUD), which was presented by [21]. Although recently compiled as a benchmark data, its use in virtual screening can be found in [22,23]. The decoys for each target have been chosen to fulfill a number of criteria to make them relevant and as unbiased as possible. Only 12 subsets of the DUD with only 704 active compounds were considered and divided into training (70%) and validation (30%) set in this study as shown in Table 2.

**Table 2.** Number of Active (Na) compounds for 12 Directory of Useful Decoys (DUD) datasets.

No	Activity Class	Na	
		Training	Validation
1	FGFR1T	90	30
2	FXA	106	40
3	GART	27	13
4	GBP	38	14
5	GR	55	23
6	HIVPR	42	20
7	HIVRT	32	11
8	HMGA	24	11
9	HSP90	24	13
10	MR	10	5
11	NA	35	14
12	PR	22	5

The last three datasets (MDDR1-3), selected from the MDL Drug Data Report MDDR [24], have been previously used for LBVS [22,25] and activity prediction [26]. The MDDR data sets contain

well defined derivatives and biologically relevant compounds that were converted to Pipeline Pilot's ECFC\_4 fingerprints and folded to give 1024 element fingerprints. A detailed description of each dataset showing the training (70%) and validation (30%) sets, activity classes, number of molecules per class, and their average pairwise Tanimoto similarity across all pairs of molecules is given in Tables 3–5. The active molecules for each dataset were used. For instance, the MDDR1 (Table 3) contains a total of 8294 active molecules, which is a mixture of both structurally homogeneous and heterogeneous active molecules (11 classes). The MDDR2 (5083 molecules) and MDDR3 (8568 Molecules) in Tables 4 and 5 respectively, contain 10 homogeneous activity classes and 10 heterogeneous ones respectively [27].

**Table 3.** Activity Classes for MDDR1.

Activity Index	Activity Class	Active Molecules		Pairwise Similarity
		Training	Validation	Mean
31420	renin inhibitors	783	347	0.573
71523	HIV protease inhibitors	535	215	0.446
37110	thrombin inhibitors	561	242	0.419
31432	angiotensin II AT1 antagonists	674	269	0.403
42731	substance P antagonists	859	387	0.339
06233	5HT3 antagonists	530	222	0.351
06245	5HT reuptake inhibitors	257	102	0.345
07701	D2 antagonists	268	127	0.345
06235	5HT1A agonists	589	238	0.343
78374	protein kinase C inhibitors	326	127	0.323
78331	cyclooxygenase inhibitors	427	209	0.268

**Table 4.** Activity Classes for MDDR2.

Activity Index	Activity Class	Active Molecules		Pairwise Similarity
		Training	Validation	Mean
07707	adenosine (A1) agonists	136	71	0.424
07708	adenosine (A2) agonists	119	37	0.484
31420	renin inhibitors	791	339	0.584
42710	monocyclic $\beta$ -lactams	78	33	0.596
64100	cephalosporins	911	390	0.512
64200	carbacephems	115	43	0.503
64220	carbapenems	732	319	0.414
64300	penicillin	88	38	0.444
65000	antibiotic, macrolide	268	120	0.673
75755	vitamin D analogues	323	132	0.569

**Table 5.** Activity Classes for MDDR3.

Activity Index	Activity Class	Active Molecules		Pairwise Similarity
		Training	Validation	Mean
09249	muscarinic (M1) agonists	620	280	0.257
12455	NMDA receptor antagonists	990	410	0.311
12464	nitric oxide synthase inhibitors	348	157	0.237
31281	dopamine $\beta$ -hydroxylase inhibitors	76	30	0.324
43210	aldose reductase inhibitors	663	294	0.37
71522	reverse transcriptase inhibitors	501	199	0.311
75721	aromatase inhibitors	444	192	0.318
78331	cyclooxygenase inhibitors	449	187	0.382
78348	phospholipase A2 inhibitors	430	187	0.291
78351	lipoygenase inhibitors	1478	633	0.365

The datasets were divided into training (70%) and validation (30%) sets for the purpose of this experiment. Ten-fold cross-validation was used for the Training set. In this cross-validation, the data set was split into 10 parts; 9 were used for training and the remaining 1 was used for testing. This process is repeated 10 times with a different 10th of the dataset used to test the remaining 9 parts during every run of the 10-fold cross validation. Figure 1 pictorially illustrates the various stages involved in the work under study.

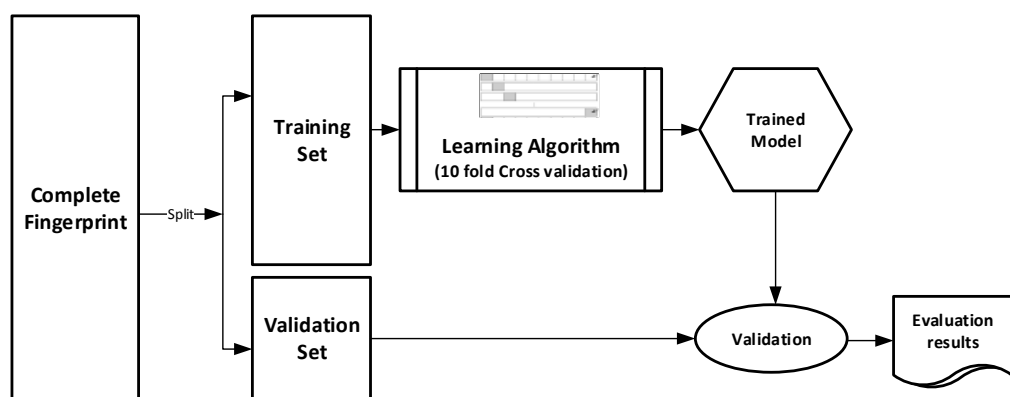


Figure 1. Experimental Design.

### 3.2. Xgboost and Machine Learning Algorithms Parameters

Identifying the optimal parameters for a classifier can be time consuming and tedious and Xgboost is not an exception. This is even more challenging in Xgboost due to the wide range of tuneable parameters for optimal performance; a few of which, using the R [28] implementation of Xgboost, we have restricted our scope to in this work. Thus, by using brute force, we obtained the best performance for Xgboost when eta, gamma, minimum child weight and maximum depth were 0.2, 0.16, 5 and 16 respectively. Where; eta is the step size shrinkage meant to control the learning rate and over-fitting through scaling each tree contribution, gamma is the minimum loss reduction required to make a split, minimum child weight is the minimum sum of instance weight needed in a child and max depth is the maximum depth of a child. Other tree booster parameters like maximum delta step, subsample, column sample and the number of trees to grow per round are left at their default values of 1 respectively. For LSVM, WEKA workbench offers a way to automate the search for optimal parameters. By using grid search, a peak performance with the radial basis kernel was obtained when gamma and cost were  $5.01187233627273 \times 10^4$  and 20 respectively. RF performed best when the maximum depth of tree was not constrained and the number of iteration set to its default value of 100. The NB classifier achieved best performance when kernel estimator parameter is used instead of normal distribution. For RBFN, we converted numeric attributes to nominal and set the minimum standard deviation to 0.1 to get the best performance.

### 3.3. Evaluation Metrics

The choice of performance evaluation for both model building and validation have been carefully selected from the most commonly used metrics in the literature. The selected evaluation metrics includes the accuracy, area under curve (AUC), sensitivity (SEN), specificity (SPC) and F-measure (F-Sc). The one run definition of AUC (Equation (11)) also known as balanced accuracy which is given by the average of the sum of sensitivity and specificity has been used in this work.

$$\text{AUC} = ((\text{SEN} + \text{SPC}))/2 \quad (11)$$

while sensitivity (SEN) (Equation (12)) and specificity (SPC) (Equation (13)) show the ability of the model to correctly classify true positive as positive and true negative as negative respectively, AUC simply describes the tradeoff between them.

$$\text{SEN} = \text{tp}/(\text{tp} + \text{fn}) \quad (12)$$

$$\text{SPC} = \text{tn}/(\text{tn} + \text{fp}) \quad (13)$$

where tp, tn, fp and fn are true positive, true negative, false positive and false negative respectively. In addition to the accuracy (Equation (14)) which is the sum of the correctly classified divided by the total number of classes, F-measure (FSc) (Equation (15)), which is the harmonic mean of precision and recall is included to serve as measure the model's accuracy.

$$\text{ACC} = ((\text{tp} + \text{tn})/(\text{tp} + \text{tn} + \text{fn} + \text{fp})) \quad (14)$$

$$\text{F Sc} = 2(\text{precision} \times \text{recall})/(\text{precision} + \text{recall}) \quad (15)$$

This work aims to introduce Xgboost for activity prediction through its performance on known datasets in drug discovery. To achieve this aim, the performance of Xgboost was compared with four state of the art machine learning algorithms used in drug discovery based on the afore-stated evaluation metrics. The prediction performances of the different machine learning algorithms on the datasets under study are tabulated in Tables 6–12. The best values for each metric is shaded.

**Table 6.** Sensitivity, Specificity, Area under Curve, Accuracy and F-measure on MDDR1 Dataset.

ML Algorithm	Training					Validation				
	SEN	SPC	AUC	ACC	F-Sc	SEN	SPC	AUC	ACC	F-Sc
XGB	0.9484	0.9958	0.9721	0.9575	0.9830	0.9579	0.9960	0.9769	0.9594	0.9536
RF	0.9474	0.9963	0.9718	0.9621	0.9514	0.9502	0.9957	0.9730	0.9590	0.9525
LSVM	0.9258	0.9943	0.9600	0.9425	0.9264	0.9357	0.9948	0.9653	0.9497	0.9371
RBFN	0.7566	0.9773	0.8670	0.7719	0.7451	0.7751	0.9777	0.8764	0.7746	0.7553
NB	0.7648	0.9781	0.8715	0.7826	0.7578	0.7488	0.9762	0.8625	0.7626	0.7383

**Table 7.** Sensitivity, Specificity, Area under Curve, Accuracy and F-measure on MDDR2 Dataset.

ML Algorithm	Training					Validation				
	SEN	SPC	AUC	ACC	F-Sc	SEN	SPC	AUC	ACC	F-Sc
XGB	0.9779	0.9981	0.9880	0.9834	0.9689	0.9820	0.9983	0.9902	0.9849	0.9673
RF	0.9562	0.9979	0.9771	0.9837	0.9689	0.9468	0.9977	0.9723	0.9823	0.9597
LSVM	0.9590	0.9978	0.9784	0.9817	0.9667	0.9436	0.9974	0.9705	0.9790	0.9547
RBFN	0.9507	0.9961	0.9734	0.9646	0.9402	0.9420	0.9960	0.9690	0.9658	0.9312
NB	0.9546	0.9963	0.9755	0.9677	0.9458	0.9401	0.9967	0.9684	0.9724	0.9403

**Table 8.** Sensitivity, Specificity, Area under Curve, Accuracy and F-measure on MDDR3 Dataset.

ML Algorithm	Training					Validation				
	SEN	SPC	AUC	ACC	F-Sc	SEN	SPC	AUC	ACC	F-Sc
XGB	0.9407	0.9937	0.9672	0.9440	0.9348	0.9493	0.9937	0.9715	0.9447	0.9448
RF	0.9209	0.9929	0.9569	0.9409	0.9350	0.9316	0.9928	0.9622	0.9397	0.9405
LSVM	0.8800	0.9885	0.93425	0.904651	0.8948	0.8983	0.9902	0.9443	0.9171	0.9120
RBFN	0.7053	0.9643	0.8348	0.680613	0.6597	0.7254	0.9657	0.8456	0.6890	0.6710
NB	0.6803	0.9613	0.8208	0.657276	0.6402	0.6636	0.9594	0.8115	0.6415	0.6211



**Table 9.** Sensitivity, Specificity, Area under Curve, Accuracy and F-measure on DUD Dataset.

ML Algorithm	Training					Validation				
	SEN	SPC	AUC	ACC	F-Sc	SEN	SPC	AUC	ACC	F-Sc
XGB	0.8677	0.9920	0.9298	0.9113	0.8616	0.8569	0.9953	0.9261	0.9471	0.8673
RF	0.8861	0.9935	0.9397	0.9294	0.8908	0.9078	0.9951	0.9515	0.9471	0.9123
LSVM	0.8659	0.9919	0.9289	0.9113	0.8683	0.8738	0.9941	0.9340	0.9375	0.8862
RBFN	0.8228	0.9895	0.9061	0.8871	0.8344	0.8503	0.9931	0.9217	0.9279	0.8537
NB	0.8783	0.9910	0.9346	0.9032	0.8730	0.9177	0.9942	0.9559	0.9375	0.9193

**Table 10.** Sensitivity, Specificity, Area under Curve, Accuracy and F-measure on COX2 Dataset.

ML Algorithm	Training					Validation				
	SEN	SPC	AUC	ACC	F-Sc	SEN	SPC	AUC	ACC	F-Sc
XGB	0.9361	0.9444	0.9403	0.9388	0.9535	0.9570	0.9362	0.9466	0.9500	0.9622
RF	0.9763	0.8879	0.9321	0.9450	0.9581	0.9783	0.8750	0.9266	0.9429	0.9574
LSVM	0.9526	0.9138	0.9332	0.9388	0.9526	0.9565	0.8958	0.9262	0.9357	0.9514
RBFN	0.9293	0.7203	0.8248	0.8379	0.8658	0.9250	0.7000	0.8125	0.8286	0.8605
NB	0.6777	0.9569	0.8173	0.7768	0.7967	0.7065	1.0000	0.8533	0.8071	0.8280

**Table 11.** Sensitivity, Specificity, Area under Curve, Accuracy and F-measure on BZR Dataset.

ML Algorithm	Training					Validation				
	SEN	SPC	AUC	ACC	F-Sc	SEN	SPC	AUC	ACC	F-Sc
XGB	0.9764	0.9028	0.9396	0.9577	0.9718	0.9884	0.8000	0.8942	0.9339	0.9551
RF	0.9720	0.9143	0.9431	0.9577	0.9720	0.9674	0.8966	0.9320	0.9504	0.9674
LSVM	0.9579	0.8714	0.9147	0.9366	0.9579	0.9348	1.0000	0.9674	0.9504	0.9663
RBFN	0.9947	0.7263	0.8605	0.9049	0.9330	1.0000	0.6444	0.8222	0.8678	0.9048
NB	0.9112	0.8571	0.8842	0.8979	0.9308	0.8478	0.9655	0.9067	0.8760	0.9123

**Table 12.** Sensitivity, Specificity, Area under Curve, Accuracy and F-measure on ER Dataset.

ML Algorithm	Training					Validation				
	SEN	SPC	AUC	ACC	F-Sc	SEN	SPC	AUC	ACC	F-Sc
XGB	0.7671	0.8522	0.8097	0.8297	0.7044	0.8837	0.7703	0.8270	0.8120	0.7755
RF	0.6860	0.8895	0.7878	0.8261	0.7108	0.6364	0.8226	0.7295	0.7350	0.6931
LSVM	0.6628	0.9316	0.7972	0.8478	0.7308	0.6727	0.9194	0.7960	0.8034	0.7629
RBFN	0.7089	0.8477	0.7783	0.8080	0.6788	0.8478	0.7746	0.8112	0.8034	0.7723
NB	0.9767	0.6368	0.8068	0.7428	0.7029	0.9818	0.5645	0.7732	0.7607	0.7941

The classification performance of the MDDR1-3, DUD, COX2, BZR and ER datasets are reported in Tables 6–12 respectively.

The experimental results on MDDR1-3 Validation datasets (Tables 6–8) shows that Xgboost produced the best accuracy, sensitivity, specificity, AUC and F-Sc across all the activity classes compared to the other machine learning methods (RF, LSVM, RBFN and NB) despite the obvious imbalance distribution of activity classes in the most of the datasets. Hence, the Xgboost method performed well for the high diverse dataset (MDDR3), and these results are particularly interesting since the MDDR3 is made up of heterogeneous activity classes which are more challenging for most machine learning algorithms.

For DUD Validation dataset (Table 9), Xgboost and RF produced the best accuracy (0.9471) compared to the other methods. In addition, Xgboost produced the best specificity across all DUD sub datasets. However, NB obtained the best sensitivity, AUC and F-Sc results.



For COX2, ER and BZR Validation datasets (Tables 10–12), it is shown that Xgboost performed well and produced the best accuracy and AUC for COX2 and ER datasets. In addition, it obtained the best F-Score results for COX2 dataset compared to the other state-of-art methods.

Visual inspection of the results shows that Xgboost produced the best accuracy for all used datasets (except for BZR dataset which produced the second best accuracy). While the performance of Xgboost on most activity classes in terms of accuracy and AUC remains the best, it still produces the best average performance across all evaluation metrics. In addition, the good performance of Xgboost is not only restricted to homogenous activity classes since it also performed well on the heterogeneous dataset.

Moreover, a quantitative approach using Kendall W test of concordance was used to rank the effectiveness of all used methods as shown in Table 13. This test shows whether a set of raters make comparable judgments on the ranking of a set of objects. Hence, the XGB, RF, LSVM, RBFN and NB methods were used as the raters, and the accuracy measure (using MDDR1-3, DUD, COX2, BZR and ER datasets respectively) were used as the ranked objects. The outputs of this test are the Kendall coefficient ( $W$ ) and the associated significance level ( $p$  value). In this paper, if the value is significant at a cutoff value of 0.01, then it is possible to give an overall ranking for the methods.

The results of the Kendall analysis for the seven datasets are shown in Table 13. The columns show the evaluation measure, the value of the Kendall coefficient ( $W$ ), the associated significance level ( $p$  value), and the ranking of prediction methods. The overall rankings of the four methods show that Xgboost significantly outperforms the other methods using accuracy measure across all datasets.

**Table 13.** Rankings of Prediction Methods based on Kendall W Test Using Accuracy Measure.

Measure	W	P	Ranks
Accuracy	0.65	0.001	XGBOOST > RF > LSVM > RBFN > NB

#### 4. Conclusions

This paper investigated the performance of Xgboost on bioactivity prediction and found out that Xgboost is indeed a robust predictive algorithm. Experimental results show that Xgboost is not only effective as a predictive model for homogeneous dataset but can replicate such effectiveness on structurally heterogeneous dataset. Experimental results show that Xgboost produces an impressive predictive accuracy, ranging from 94.47% accuracy in the heterogeneous data to 98.49% in the homogeneous one. In addition to the obvious fact that Xgboost has been shown in this work to be a good predictive tool for bioactive molecule, we are hopeful that by this Xgboost would be seen as an invaluable addition to already known computational approaches to target prediction and thus leading to a wider perspective of the data rather than a single approach.

**Acknowledgments:** This work is supported by the Ministry of Higher Education (MOHE) and Research Management Centre (RMC) at the Universiti Teknologi Malaysia (UTM) under the Research University Grant Category (VOT Q.J130000.2528.14H75).

**Author Contributions:** I.M. is a researcher and conducted this project research under the supervision of F.S. All authors read and approved the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Johnson, M.A.; Maggiora, G.M. *Concepts and Applications of Molecular Similarity*; John Wiley & Sons: New York, NY, USA, 1990.
2. Wassermann, A.M.; Geppert, H.; Bajorath, J. Searching for target-selective compounds using different combinations of multiclass support vector machine ranking methods, kernel functions, and fingerprint descriptors. *J. Chem. Inf. Mod.* **2009**, *49*, 582–592. [[CrossRef](#)] [[PubMed](#)]

3. Yang, Z.R. Biological applications of support vector machines. *Brief. Bioinform.* **2004**, *5*, 328–338. [[CrossRef](#)] [[PubMed](#)]
4. Deconinck, E.; Zhang, M.H.; Coomans, D.; Vander Heyden, Y. Classification tree models for the prediction of blood-brain barrier passage of drugs. *J. Chem. Inf. Mod.* **2006**, *46*, 1410–1419. [[CrossRef](#)] [[PubMed](#)]
5. Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J.C.; Sheridan, R.P.; Feuston, B.P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947–1958. [[CrossRef](#)] [[PubMed](#)]
6. Kauffman, G.W.; Jurs, P.C. QSAR and k-nearest neighbor classification analysis of selective cyclooxygenase-2 inhibitors using topologically-based numerical descriptors. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1553–1560. [[CrossRef](#)] [[PubMed](#)]
7. Koutsoukas, A.; Lowe, R.; KalantarMotamedi, Y.; Mussa, H.Y.; Klaffke, W.; Mitchell, J.B.; Glen, R.C.; Bender, A. In silico target predictions: Defining a benchmarking data set and comparison of performance of the multiclass naïve bayes and parzen-rosenblatt window. *J. Chem. Inf. Mod.* **2013**, *53*, 1957–1966. [[CrossRef](#)] [[PubMed](#)]
8. Krenker, A.; Kos, A.; Bešter, J. *Introduction to the Artificial Neural Networks*; INTECH Open Access Publisher: Rijeka, Croatia, 2011.
9. Lavecchia, A. Machine-learning approaches in drug discovery: Methods and applications. *Drug Discov. Today* **2015**, *20*, 318–331. [[CrossRef](#)] [[PubMed](#)]
10. Wolpert, D.H. The supervised learning no-free-lunch theorems. In *Soft Computing and Industry*; Springer: London, UK, 2002; pp. 25–42.
11. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
12. Adam-Bourdarios, C.; Cowan, G.; Germain-Renaud, C.; Guyon, I.; Kégl, B.; Rousseau, D. The Higgs Machine Learning Challenge. *J. Phys. Conf. Ser.* **2015**. [[CrossRef](#)]
13. Phoboo, A.E. Machine Learning wins the Higgs Challenge. *CERN Bull.* **2014**. Available online: <http://cds.cern.ch/journal/CERNBulletin/2014/49/News%20Articles/1972036> (accessed on 24 April 2016).
14. Harper, G.; Bradshaw, J.; Gittins, J.C.; Green, D.V.; Leach, A.R. Prediction of biological activity for high-throughput screening using binary kernel discrimination. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1295–1300. [[CrossRef](#)] [[PubMed](#)]
15. Chen, T.; Guestrin, C. Xgboost: A Scalable Tree Boosting System. *arXiv:1603.02754* **2016**. [[CrossRef](#)]
16. Chang, C.-C.; Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27. [[CrossRef](#)]
17. John, G.H.; Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–20 August 1995.
18. Bugmann, G. Normalized Gaussian radial basis function networks. *Neurocomputing* **1998**, *20*, 97–110. [[CrossRef](#)]
19. Sutherland, J.J.; O'Brien, L.A.; Weaver, D.F. Spline-Fitting with a Genetic Algorithm: A Method for Developing Classification Structure–Activity Relationships. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1906–1915. [[CrossRef](#)] [[PubMed](#)]
20. Helma, C.; Cramer, T.; Kramer, S.; de Raedt, L. Data Mining and Machine Learning Techniques for the Identification of Mutagenicity Inducing Substructures and Structure Activity Relationships of Noncongeneric Compounds. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1402–1411. [[CrossRef](#)] [[PubMed](#)]
21. Huang, N.; Shoichet, B.K.; Irwin, J.J. Benchmarking sets for molecular docking. *J. Med. Chem.* **2006**, *49*, 6789–6801. [[CrossRef](#)] [[PubMed](#)]
22. Al-Dabbagh, M.M.; Salim, N.; Himmat, M.; Ahmed, A.; Saeed, F. A Quantum-Based Similarity Method in Virtual Screening. *Molecules* **2015**, *20*, 18107–18127. [[CrossRef](#)] [[PubMed](#)]
23. Cross, S.; Baroni, M.; Carosati, E.; Benedetti, P.; Clementi, S. FLAP: GRID molecular interaction fields in virtual screening. Validation using the DUD data set. *J. Chem. Inf. Mod.* **2010**, *50*, 1442–1450. [[CrossRef](#)] [[PubMed](#)]

24. BIOVIA. MDDR. Retrieved 15–07, 2015. Available online: <http://accelrys.com/products/databases/bioactivity/mddr.html> (accessed on 15 July 2015).
25. Abdo, A.; Saeed, F.; Hamza, H.; Ahmed, A.; Salim, N. Ligand expansion in ligand-based virtual screening using relevance feedback. *J. Comput. Aided Mol. Design* **2012**, *26*, 279–287. [[CrossRef](#)] [[PubMed](#)]
26. Abdo, A.; Leclère, V.; Jacques, P.; Salim, N.; Pupin, M. Prediction of New Bioactive Molecules using a Bayesian Belief Network. *J. Chem. Inf. Model.* **2014**, *54*, 30–36. [[CrossRef](#)] [[PubMed](#)]
27. Hert, J.; Willett, P.; Wilton, D.J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. New methods for ligand-based virtual screening: Use of data fusion and machine learning to enhance the effectiveness of similarity searching. *J. Chem. Inf. Mod.* **2006**, *46*, 462–470. [[CrossRef](#)] [[PubMed](#)]
28. The R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2013.

**Sample Availability:** Not Available.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).