

# SCIENTIFIC REPORTS



OPEN

## *Trader* as a new optimization algorithm predicts drug-target interactions efficiently

Yosef Masoudi-Sobhanzadeh<sup>1</sup>, Yadollah Omid<sup>2</sup>, Massoud Amanlou<sup>3</sup> & Ali Masoudi-Nejad<sup>1</sup>

Several machine learning approaches have been proposed for predicting new benefits of the existing drugs. Although these methods have introduced new usage(s) of some medications, efficient methods can lead to more accurate predictions. To this end, we proposed a novel machine learning method which is based on a new optimization algorithm, named *Trader*. To show the capabilities of the proposed algorithm which can be applied to the different scope of science, it was compared with ten other state-of-the-art optimization algorithms based on the standard and advanced benchmark functions. Next, a multi-layer artificial neural network was designed and trained by *Trader* to predict drug-target interactions (DTIs). Finally, the functionality of the proposed method was investigated on some DTIs datasets and compared with other methods. The data obtained by *Trader* showed that it eliminates the disadvantages of different optimization algorithms, resulting in a better outcome. Further, the proposed machine learning method was found to achieve a significant level of performance compared to the other popular and efficient approaches in predicting unknown DTIs. All the implemented source codes are freely available at <https://github.com/LBBSofTrader>.

A drug is referred to a substance, except for the nutrients, which impose a temporary and/or diachronic physiological impact(s) in the body. Based on the mechanism of actions and therapeutic properties of drugs, they can be categorized into several classes such as the anatomical therapeutic chemical classification (ATC) and biopharmaceutics classification systems (BCS). Because of their importance and critical efficacies, many researchers have proposed various methods for the design of a drug<sup>1</sup>. Nonetheless, the design of a new drug is a very costly and time-consuming process, which takes over 15 years. Also, lots of drug discovery and development projects may fail, in large part because of the rigorous controls during drug development phases. Hence, researchers attempted to find other approaches for the treatment of diseases such as drug repurposing method as a cost- and time-effective strategy that offers many new benefits of the existing drugs. Several computational manners have been suggested for the repurposing of medications. These approaches can be categorized into some classes, including:

- i) Molecular docking methods: These methods, which look for ligands that can bind to proteins based on their multi-dimension structures, are the most popular approaches in drug repositioning field<sup>2</sup>. However, the methods cannot be used if the multi-dimension structure of a protein or a ligand is unknown.
- ii) Metabolic pathway-based methods: These procedures are usually used for treating orphan or rare diseases. For this purpose, the metabolic pathways related to the disease are identified. Next, drugs, which can affect the metabolic pathways of the diseases, are investigated<sup>3</sup>, and then, introduced to treat the diseases if they are qualified. Since the metabolic pathways of many orphan and rare diseases are not determined, these methods have a low level of success rate.
- iii) Connectivity-MAP (CMAP) methods: These approaches, which confront lots of genomic data, are used to discover relationships between diseases and genes<sup>4</sup>. For the methods, one can refer to some limitations such as various cell-lines, platforms, etc., which make the data inconsistent.
- iv) Data-mining methods: These methods, which include different procedures such as text-mining, machine

<sup>1</sup>Laboratory of systems Biology and Bioinformatics (LBB), Institute of Biochemistry and Biophysics, University of Tehran, Tehran, Iran. <sup>2</sup>Research Center for Pharmaceutical Nanotechnology, Biomedicine Institute, Tabriz University of Medical Sciences, Tabriz, Iran. <sup>3</sup>Drug Design and Development Research Center, The Institute of Pharmaceutical Sciences (TIPS), Tehran University of Medical Sciences, Tehran, 14176-53955, Iran. Correspondence and requests for materials should be addressed to A.M.-N. (email: [amasoudin@ut.ac.ir](mailto:amasoudin@ut.ac.ir))

learning, etc., are the most powerful ones in finding the novel usages of drugs. Since the methods act based on existing data, they increase the success rate of drug repositioning and many researchers take them into consideration<sup>5</sup>. Nevertheless, the validity of the acquired results remains a primary challenge.

The existing machine learning methods might achieve acceptable results. However, the more effective the approaches are, the better the prediction will be. To this end, we proposed an improved and efficient machine learning method which predicts drug-target interactions (DTIs) efficiently and fits into the fourth category of the groups mentioned above. The proposed method (the so-called ANNTR) is a multi-layer artificial neural network which is trained by a novel optimization algorithm called “*Trader*”. Accordingly, a proper model with a higher predicting ability is acquired. Besides introducing an efficient and improved machine learning approach for predicting DTIs, two other facts motivate us to introduce *Trader* optimization algorithm. First, an efficient algorithm, which eliminates the limitations of the optimization algorithms and can be applied to different fields such as engineering, biology, computer science, etc., is useful and essential. Second, a comprehensive and suitable comparison of optimization algorithms with others can determine their actual performance in the real-world usages.

## Related Works

Our proposed method, which is a combination of artificial neural network and *Trader* optimization algorithm (ANNTR), falls into the data-mining class of drug repositioning and on predicting DTIs. This section is allocated to reviewing the related literature from the data-mining viewpoints. The conducted investigations have been categorized into six classes, as follows:

- i) Learner-based methods: In these studies, learners such as Deep learning<sup>6,7</sup>, Support vector machine<sup>8–11</sup>, Regression algorithms<sup>12</sup>, K-nearest neighbors<sup>13</sup>, Rotation forest learner<sup>11</sup>, and Relevance vector machine<sup>14</sup> aimed to find out the relationships between the input and output using labeled datasets. The acquired model is evaluated and applied to predict unknown DTIs. Since every learner uses a different method for separating samples, their results differ from each other. The biggest weakness of the mentioned literature works is generating negative datasets and obtaining a model based on them. For this reason, the percentage of error goes up due to a possible positive interaction between a drug and a target in the generated negative dataset. To tackle such restriction, one-class classification machine learning approaches can be used<sup>15</sup>. There is a low level of accuracy in the methods used in the related literature despite the fact that their obtained results are acceptable. To enhance the prediction accuracy, we have introduced an efficient machine learning method, which is based on a new optimization algorithm, so-called “*Trader*”, as well as an artificial neural network.
- ii) Network-based methods: This type of literature works formulate drugs and their various targets (genes, proteins, enzymes, metabolic pathways, etc.) and then analyze them for obtaining new information. In a series of related works, the designed network is examined by various algorithms such as Random walk<sup>16,17</sup> and Random forest<sup>18</sup>. Unlike the first class of related works which depends on the negative dataset<sup>19</sup>, the second group only considers the existing information. As a result, the error of the second category is lower than the first one. Nevertheless, the performance of the first category is higher than the second group.
- iii) Prioritization-based methods: These types of researches calculate drug-drug, network-network or target-target similarities. After they are ranked based on acquired scores, the intended drugs are suggested for treating diseases. To compute the scores, chemical information of drugs, topological information of networks, and sequence information of targets are examined<sup>20</sup>. Considering different studies, it can be concluded that the similarity is not an only determinant factor in the repositioning of drugs. Hence, the false positive rates of prioritization-based methods are high. To overcome the restriction, some researches integrate different information and then calculate the similarity scores<sup>21</sup>.
- iv) Mathematics and probabilistic-based methods: This type of studies formulate the problem as a graph and then mine it to obtain new information<sup>22</sup>. These methods run into difficulties when there are orphan nodes in the generated graph. To deal with the existing constraint, a matrix regulation and factorization method may be useful<sup>23</sup>.
- v) Ensemble-based methods: It has been shown that a proper combination of machine learning methods usually leads to better results in computer science problems. Inspired by the combination idea, some researchers have predicted DTIs using a combination of the above-mentioned classes<sup>24–26</sup>. Although these methods enhance the separability power of a drug-target predictor, they increase the error rate and suffer from the disadvantages of the combined methods.
- vi) Review-based approaches: Large numbers of drug-target prediction literature studies are considered just to review articles which have investigated the problem from various viewpoints such as applied tools<sup>27</sup>, methods<sup>28</sup>, databases, software applications<sup>29</sup>, etc. These articles usually include a discussion of the advantages and disadvantages of proposed methods and give some directions to be followed in the future<sup>30</sup>.

## Methods and Materials

**Preparing the datasets.** We integrate chemical and genomic spaces and gather information about drugs and targets as a dataset, similar to the work carried out by Yamanishi *et al.*<sup>31</sup>. The targets are divided into four classes, including enzymes (EN), ion channel proteins (IC), G-protein coupled receptors (GP), and nuclear receptor proteins (NR). To provide the datasets, the following steps can be considered:

- i) The chemical information on drugs and ligands is obtained from KEGG DRUG and KEGG LIGAND databases<sup>32</sup>. Then, the similarity scores between drugs are calculated by Eq. (1)<sup>14</sup>. For this purpose, the

Dataset = [];  
 For every existing drug-target interaction such as (D, T):  
 DR= (A set of drugs which their target set includes T) – D;  
 TA= (A set of targets of D) – T;  
 DR\_SIM= An array which consists of the similarity effect score between D and DR;  
 TA\_SIM= An array which consists of the alignment score between T and TA;  
 N= length (DR); D\_MI= min (DR); D\_MA= max (DR); D\_AVG= mean (DR);  
 M= length (TA); T\_MI= min (TA); T\_MA= max (TA); T\_AVG= mean (TA);  
 R= [N, D\_MI, D\_MAX, D\_AVG, M, T\_MI, T\_MA, T\_AVG];  
 Dataset= Dataset U R;  
 Return Dataset

**Figure 1.** Pseudocodes for generating the dataset. The generated datasets only include positive drug-target interactions and have been obtained based on the chemical similarity score of drugs and smith waterman alignment score of targets.

Name	Number of samples	Samples in the training set	Samples in the test set
EN	2,236	1,736	500
IC	1,374	1,074	300
GP	504	404	100
NR	47	27	20

**Table 1.** Properties of the generated datasets.

pharmacological effects of medications on 17109 molecular properties are taken into consideration.

$$\text{SIM}(D, D') = \frac{\sum_{i=1}^n W_i F_i F'_i}{\sqrt{\sum_{i=1}^n W_i F_i^2} \sqrt{\sum_{i=1}^n W_i F'^2_i}} \quad (1)$$

Where, n,  $F_i$ , SIM, and  $W_i$  are a total number of molecular properties (17109), the  $i^{\text{th}}$  molecular feature, the similarity score between two drugs such as D and D', and the weight of the  $F_i$  calculated by Eq. (2)<sup>14</sup>, respectively.

$$W_i = \exp(-d_i^2 / (\sigma^2 h^2)) \quad (2)$$

Where,  $d_i$ ,  $\sigma$ , and h are the frequency of  $i^{\text{th}}$  feature, the standard deviation of  $d_k$  ( $k = 1$  through n), and a constant value 0.1, respectively. Using Eq. (1), a matrix of the effect similarity score for every pair of drugs is created.

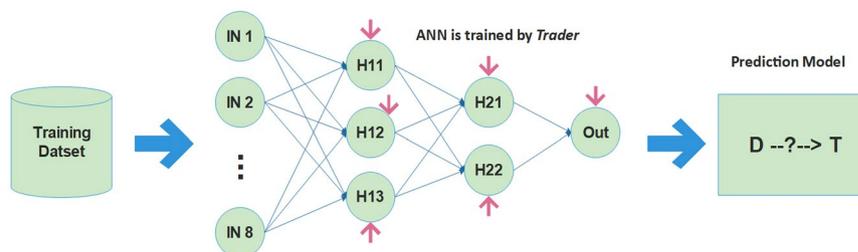
- ii) Amino acid sequences of protein targets are obtained from the DrugBank<sup>33</sup> database and KEGG GENE databases. Further, we have developed an integrated database named DrugR+<sup>34</sup> (<http://www.drugr.ir>) which is a relational database and contains all data of DrugBank and some data of KEGG. Next, the similarity score between every pair of targets is computed by the normalized smith and waterman alignment scoring method<sup>35</sup>, and a matrix is generated for target-target similarity scores.
- iii) The interaction information between drugs and targets is obtained from the DrugR+ database.

For every type of the targets, a dataset is created by the pseudocode presented in Fig. 1. These datasets can be used as gold standard datasets by researchers who want to predict the interaction between drugs and targets using machine learning approaches. In Table 1, the attributes of the generated datasets are also shown.

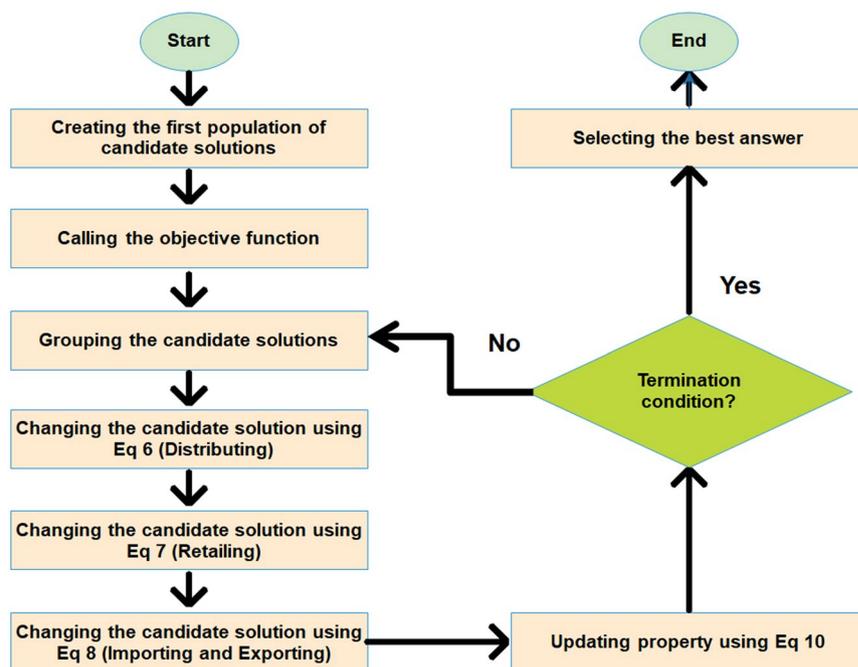
**The machine learning approach.** Our proposed method, whose framework is depicted in Fig. 2, creates a prediction model using a multi-layer perceptron (MLP) artificial neural network (ANN) with two hidden layers. The generated datasets are divided into two sets, including (i) training and (ii) testing sets. For all the generated datasets, the ANN is trained by *Trader* in which every candidate solution consists of 38 variables. There are 8, 3, 2, and 1 neurons in the input, the first hidden, the second hidden, and the output layers, respectively. In the ANN, all the neurons of a layer are connected to all the neurons of the next layer, and hence, the total number of synapses or ANN's edges is  $8*3 + 3*2 + 2*1 = 32$ . Moreover, since there are six biases which are specified in Fig. 2, the total number of variables will be  $32 + 6 = 38$  in a potential answer. In this problem, the objective function is considered as root mean square error (RMSE) which is computed by Eq. (3):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^S (P_i - O_i)^2}{S}} \quad (3)$$

Where, S, P, and O are the total number of samples, predicted and real-world values, respectively.



**Figure 2.** The framework of the proposed method for drug repurposing. After generating the datasets, Trader trains the ANN using datasets. When the ANN is appropriately trained, the model is generated and then applied to the prediction of the unknown drug-target interactions. IN, H, D, and T show neurons of the input layer, and neurons of hidden layers, a drug, and a target, respectively.



**Figure 3.** The flowchart of *Trader*: The proposed optimization algorithm starts with some candidate solutions which each of them determine the weights of the ANN. Next, they are placed into several groups and are improved by Eq. 6 through 8 (see the text for details). The steps of *Trader* are repeated until the termination condition is satisfied. By passing the steps of the algorithm, the value of RMSE is also reduced and a suitable predictor model is acquired.

**Trader optimization algorithm.** Our proposed algorithm, *Trader*, has been inspired by the intelligent behavior of traders who are looking for more profit and property using different operations such as retailing, importing, exporting, and many other activities. In Fig. 3, the flowchart of *Trader* has been shown. *Trader* consists of several steps that are described, as follows:

- i) Creating the first population of candidate solutions: Like other optimization algorithms, *Trader* starts with some potential answers which consist of several variables and can be considered as an array. Equation (4) shows a candidate solution (CS) with  $n$  variables:

$$\text{Variable} = \{v_1, v_2, \dots, v_n, G\} \quad (4)$$

Where  $G$  determines the group of the CS which belongs to a trader,  $v_i$  shows the  $i^{\text{th}}$  variable. The groups are not specified at the beginning of the algorithm. For the drug repurposing problem, a CS determines the weights of the ANN's edges, and the variables show the edges of the ANN. Therefore, the total number of variables and edges of the ANN are the same.

- ii) Calling the objective function: After creating the first population of CSs, the worthiness of each of them is calculated by an objective function (OF), whose worthiness is defined based on a problem nature. For example, the fitness of a CS is computed by the value of the error in the problem of training an artificial neural network (Eq. (3)).

- iii) Grouping the candidate solutions: The groups are constituted based on the number of traders and their properties. At the start of the algorithm, all the traders have a same property which will be updated during the algorithm's iterations. Equation (5) is used to calculate the number of CSs devoted to a specific trader (a group):

$$NB_i = 2 + \text{round} \left( \frac{P_i}{\sum_{j=1}^T P_j} \times (C - 2 \times T) \right) \quad (5)$$

where,  $NB_i$ ,  $P_i$ ,  $C$ , and  $T$  are the total number of CSs assigned to the  $i^{\text{th}}$  trader or group, the property of the  $i^{\text{th}}$  trader, the number of existing CSs, and the number of traders, respectively. Also, the constant value of 2 indicates that none of the traders or groups is eliminated during the algorithm iterations, and at least two CSs remain in every group. Figure S2 has illustrated an example of the competition among traders for getting the CSs.

- iv) Changing the candidate solutions: After grouping candidate solutions, at first, the best CS of each group named *Master CS* is selected, and then its variable values are distributed to the another CS, named *Slave CS*, using Eq. (6):

$$\sum_{j=1}^{C_k} (\sum_{i=1}^R (CS\_slave\_j(\text{rand}(n)) = CS\_master\_k(\text{rand}(n)))) \quad (6)$$

where  $n$  is the total number of variables in a CS,  $R$  is a random integer value between  $[1, n]$ ,  $C_k$  is the number of CSs of the  $k^{\text{th}}$  group,  $CS\_slave\_j$  is the  $j^{\text{th}}$  *Slave CS* of the  $k^{\text{th}}$  group, and  $CS\_master\_k$  is the *Master CS* of the  $k^{\text{th}}$  group. In case Eq. (6) enhances the value of the OF (RMSE), these changes are ignored. Otherwise, they will be accepted. In addition to the Eq. (6) which helps the *Slave CSs* to improve their value of OF, there is another operator that changes the *Slave CSs* based on their contents. These changes are applied to the *Slave CSs* using Eq. (7).

$$\sum_{i=1}^R (CS_{slave(M)} = CS_{slave(M)} + k \times \text{rand}(CS_{slave(M)})) \quad (7)$$

where  $R$  is a random integer value between 1 and  $n/10$ ,  $M$  is a random integer value between 1 and  $n$ ,  $CS_{slave}$  is a *Slave CS*, and  $k$  is an arbitrary value which is selected either 1 or  $-1$ . Like the previous operator, the changes are accepted if they improve the value of the OF. Unlike Eqs (6) and (7) which only change *Slave CSs*, there is another equation (Eq. (8)) which alters *Master CSs*. This operator exchanges values of variables among *Master CSs*. For applying it to the *Master CSs*, some of the values of the best CS of other groups are randomly chosen and then are imported to the selected *Master CS*.

$$\sum_{j=1}^T \sum_{i=1}^R (CS\_master\_j(\text{rand}(n)) = CS\_master\_k(\text{rand}(n))) \quad (8)$$

Where,  $R$  is a random integer value between 1 and  $n$ ,  $j$  and  $k$  indicate the importer and exporter groups,  $CS\_master\_j$  shows the *Master CS* of the importer group, and  $CS\_master\_k$  shows the *Master CS* of the exporter group. The value of  $k$  is calculated by Eq. (9).

$$K = \{a | a \neq j \text{ and } a \text{ is an integer random value in } 1 \leq a \leq n\} \quad (9)$$

Like the other operators of *Trader*, the changes, induced by the Eq. (8), are accepted if the imported values improve the value of the OF. By the Eq. (6) through (8), the weights of the ANN's edges are altered, and a new drug-target predictor is acquired. Provided that the new drug-target predictor reduces the value of the RMSE (Eq. (3)), the changes of weights are admitted. Figures S3 through S5 illustrate how the changes on CSs are applied.

- v) Updating property: The operators of *Trader*, shown by Eq. 6 through 8, may change the CSs. Hence, the total value of the objective functions of a group, which is computed using Eq. (10), varies. Accordingly, the property of the groups must be updated.

$$\text{Property}_i = \left\{ \sum_{j=1}^B \text{OF}(j) | CS(j, G) = G_i \right\} \quad (10)$$

where,  $property_i$  is the property of the  $i^{\text{th}}$  trader or group,  $B$ ,  $G_i$ , and  $G$  are the number of CSs, the  $i^{\text{th}}$  group, and the group which the  $j^{\text{th}}$  CS belongs to it, respectively.

- vi) Termination condition: Like other optimization algorithms, each of the following options can be considered as the termination condition of *Trader*: (i) calling algorithms steps based on a predefined number of iterations; (ii) reaching a determined value of accuracy or error; (iii) elapsing a certain amount of time; (iv) stabilizing of the best answer in recent iterations. For training the ANN, a predefined number of iterations has been selected as the termination condition.
- vii) Selecting the best answer: When the termination condition is satisfied, a CS having the best value of OF will be selected and introduced as the solution to the problem. For the DTIs prediction problem, a CS, which has the minimum value of the RMSE, is chosen as a solution to forecast unknown DTIs. Figure 4 shows the pseudocode of *Trader*.

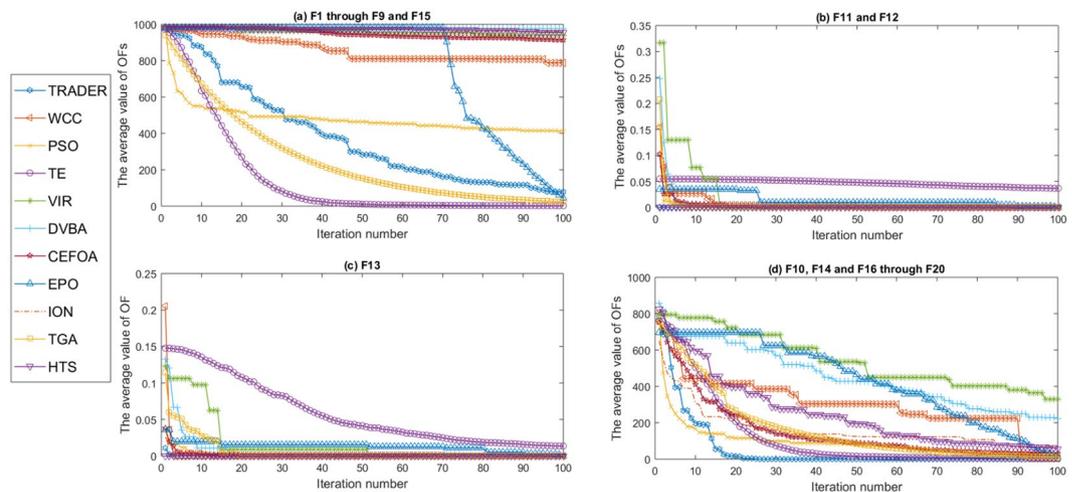
1. **Input**  
 T= a number of traders, B= a number of CS, n= a number of variables;  
 R= range of variables, CS = zeros (B, n+1);  
 NB = zeros (1, T); \\ an array for holding the number of CS in every group
2. **Creating the first population of CSs**  
 For I= 1 to B  
   For J=1 to n  
     CS (I, J) = a random value in the determined rage;  
 For I= 1 to T  
   Update NB (I) using Eq 1;
3. **Calling the OF**  
 For I= 1 to B  
   CS (I).profit = the returned value from OF;
4. **Grouping the CSs**  
 TMP = NB;  
 For I=1 to B  
   K = rand (length (TMP)), CS (I, n+1) = k, TMP (k) = TMP (k) -1;  
   If TMP (K) == 0  
     TMP (:, k) = [];
5. **Changing the candidate solutions using Eq 6: Distributing**  
 For I= 1 to T  
   Master = the best CS of the Ith group;  
   For J= 1 to B  
     If (CS (J, n+1) == I) and (CS (j) != Master)  
       Distribute values from the Master CS to slave CSs based on Eq 6 provided that  
       value of the OF is improved. Else ignore the changes;
6. **Changing the candidate solutions using Eq 7: Retailing**  
 For I= 1 to B  
   Update slave CSs using Eq 7 provided that the value of the OF is improved. Else ignore the  
   changes;
7. **Changing the candidate solutions using Eq 8: Importing & Exporting**  
 For every master CSs  
   Update master CS using Eq 8 provided that the value of the OF is improved. Else ignore  
   the changes;
8. **Updating property**  
   Update property of traders using Eq 10;
9. **Termination condition**  
   If termination condition is satisfied  
     Go to 10;  
   Else  
     Go to 4;
10. **Selecting the best answer**

**Figure 4.** The pseudocode of *Trader*. For training the ANN, *Trader* produces some potential answers which consist of several variables (the edges of the ANN). *Trader* includes three operations, shown by Eq. 6 through 8. These operations change the weight of ANN's edges differently. For instance, Eq. (7) alters them based on their content, or Eq. (8) tries to improve them by importing some values from the best solutions.

## Results

The proposed machine learning approach has been implemented in MATLAB programming language and all the implemented source codes are available at (<https://github.com/LBBSOft/Trader>). This section contains three categories of results as follows:

**Trader in comparison with the other optimization algorithms.** Besides *Trader*, ten state-of-the-art optimization algorithms (PSO<sup>36</sup>, WCC<sup>37</sup>, TGA<sup>38</sup>, TE<sup>39</sup>, EPO<sup>40</sup>, ION<sup>41</sup>, VIR<sup>42</sup>, DVBA<sup>43</sup>, HTS<sup>44</sup>, and CEFOA<sup>45</sup>) were implemented. Then, these algorithms were applied to 20 benchmark functions which are used in various researches in which the above-mentioned optimization algorithms have been introduced. These standard test functions, which are available in Table S1 (Supplementary File), are categorized into unimodal, multimodal, fix dimension, expanded, penalized, and hybrid categories. Since optimization algorithms produce variable results in different executions, these algorithms are recommended to be executed at least 30 times for an intended problem, and then, the final best-obtained result should be reported to answer the problem<sup>46</sup>. Hence, all of the above-mentioned algorithms were executed over 50 individual executions on the determined benchmark functions with high dimensions. Further, the algorithms were executed under similar conditions such as the number of iterations during execution and the number of OF callings, and their parameters are determined in such a way that their performances were maximized. To evaluate the optimization algorithms, the criteria like convergence



**Figure 5.** The convergence of the algorithms on different test functions shown by  $F_i$ . For instance,  $F_i$  presents  $i^{\text{th}}$  test function. (a) The average convergence of the algorithms on F1 through F9 and F15. (b) The average convergence of the algorithms on F11 and F12. (c) The convergence of the algorithms on F13. (d) The average convergence of the algorithms on F10, F14, and F16 through F20. Among the test functions, F10, F14, and F16 through F20 are the benchmark functions with the small sizes, but the others have a large number of variables with a higher range. These diagrams show that *Trader* has more stable behavior than the others on different benchmark functions whereas EPO, TGA, and ION fall into local optima for some of them as F11, F12, and F13. Also, the results state that the performance of the algorithms is almost the same when the size of a problem or the number of variables is small.

and stability of acquired results are considered. Figure 5 shows the convergences of the algorithms on the test functions, which relate to their best result over 50 individual executions. For similar convergence behaviors, the average outcomes were drawn. For example, the results of F11 and F12 were merged into one. Besides, the convergences of the algorithms on each of the benchmark functions are presented in Fig. S6 (Supplementary File).

The acquired results show the below findings:

- i) *Trader*, TGA, TE, and EPO have more convergence speed than the others and can get better results. However, the convergence speed of EPO is lower than TGA, *Trader*, and TE in early steps and depends on its fourth quarter of iterations in which the range of variables become smaller and smaller (Fig. 5a). Therefore, EPO gets more speed of convergence in the last quarter. TE, TGA, and ION use a similar method as does EPO and limit the range of variables by passing the iteration of the algorithms' steps; and therefore, produce the better results for some special problems such as F1 through F9.
- ii) EPO, TGA, and ION algorithms cannot produce the desired results to some problems such as F11 and F12 (Fig. 5b,c). The other algorithms outperform these three algorithms when their iterations of steps are enhanced and can acquire better results.
- iii) For the small-sized benchmark functions such as F17 through F20, the algorithms have similar performance, and all of them can obtain the optimal solution (Fig. 5d).
- iv) The convergence of VIR, HTS, WCC, and CEFOA are slower than other algorithms for some of the test functions (Fig. 5a). Nonetheless, they can acquire acceptable results with enhancing the allocated time or the number of iterations, but not EPO, TE, and TGA, because of falling into local optima.

For an accurate evaluation of the algorithms, we summarized their findings over 50 distinct executions in Tables S2 through S4 (Supplementary File) with two decimal digits of accuracy using the ANOVA one-way test. We also provided Table 2 which includes the P- values of the algorithms compared to *Trader* as a test base and shows that the null hypothesis can be strongly rejected. For this purpose, the Wilcoxon rank sum test, which states how much the generated results are the same<sup>47</sup>, was done.

Based on the average and standard deviation point of views (Table 3), *Trader* has proper functionality, but its results are close to the outcomes of EPO, TGA, and TE for the test functions F1 through F9. However, they are only suitable for the problems whose optimal answer is 0 because of their operators' nature. For this reason, their performance is the same for all of the benchmark function. From the STD aspect, HTS will be the best algorithm and the best option when the range of variables is small in a problem.

**The proposed machine learning method against the others.** In the second part of this section, the performance of the proposed method (ANNTR) is evaluated based on four gold standard datasets<sup>48</sup>, and then, is compared against three state-of-art methods, including the rotation forest-based drug-target (RFDT) predictor method<sup>11</sup>, the Bayesian (BAY) ranking-based method<sup>22</sup>, and a relevance vector machine-based method<sup>14</sup> (RVM). The datasets, which are named Enzyme, Ion channel, G-protein, and Nuclear receptor, consist of 4,449, 2,029, 1,268, and 168 DTIs samples, respectively. Further, the samples have been marked using positive and negative

	WCC	PSO	TE	VIR	DVBA	CEFOA	EPO	ION	TGA	HTS
F1	1.2823E-17	2.17584E-17	7.96337E-18	1.35201E-17	1.15344E-17	7.7553E-18	9.03396E-18	1.09304E-17	1.38319E-17	1.3884E-17
F2	8.17978E-18	1.47874E-17	1.38295E-17	1.04958E-17	1.27209E-17	8.06865E-18	1.00463E-17	1.35367E-17	1.26639E-17	1.38459E-17
F3	1.16996E-17	3.30908E-16	1.30661E-17	1.36657E-17	1.18621E-17	1.24203E-17	1.23171E-17	9.83758E-18	1.16977E-17	8.27569E-18
F4	1.2055E-17	8.22248E-18	9.02283E-18	7.39232E-18	7.75241E-18	1.28847E-17	1.19758E-17	9.30672E-18	1.37804E-17	7.30947E-18
F5	1.01663E-17	2.09249E-16	1.24753E-17	1.2685E-17	8.38653E-18	1.05268E-17	1.02146E-17	1.1633E-17	1.20785E-17	1.23987E-17
F6	9.01648E-18	4.52061E-16	1.1695E-17	8.2151E-18	7.90692E-18	1.05875E-17	1.38477E-17	9.47126E-18	1.12016E-17	8.64754E-18
F7	1.23746E-17	5.33226E-16	1.06412E-17	1.20058E-17	1.33613E-17	1.38445E-17	1.09327E-17	8.0456E-18	8.12099E-18	8.88564E-18
F8	1.30066E-17	1.42263E-16	1.28199E-17	8.78684E-18	1.36323E-17	9.53908E-18	8.45523E-18	8.84025E-18	1.14191E-17	1.04104E-17
F9	9.55092E-18	4.43485E-15	1.12016E-17	1.09505E-17	1.3547E-17	9.08583E-18	1.24165E-17	1.2392E-17	9.75433E-18	1.10783E-17
F10	7.60206E-18	7.44729E-18	1.08167E-17	1.25717E-17	1.36659E-17	7.984E-18	1.10854E-17	1.03828E-17	7.15017E-18	9.4482E-18
F11	8.21206E-18	1.26785E-17	9.26514E-18	1.08007E-17	8.23656E-18	1.13197E-17	8.92425E-18	1.16878E-17	1.19361E-17	5.78827E-20
F12	1.02496E-17	7.65836E-18	8.68404E-18	1.35198E-17	8.14279E-18	1.29014E-17	1.087E-17	1.41048E-17	7.61847E-18	6.82815E-20
F13	7.81969E-18	1.38629E-17	7.09882E-18	1.25416E-17	1.28412E-17	1.32043E-17	7.6627E-18	9.89096E-18	8.90233E-18	1.27194E-17
F14	1.01145E-17	1.35008E-17	8.35102E-18	8.93012E-18	8.09446E-18	8.02754E-18	1.32086E-17	1.11623E-17	1.09514E-17	8.09033E-18
F15	1.30937E-17	8.17177E-16	9.54593E-18	1.06927E-17	9.90528E-18	7.60286E-18	8.76134E-18	7.93745E-18	8.36558E-18	8.76159E-18
F16	1.00145E-17	7.41693E-18	1.34447E-17	1.3742E-17	1.05346E-17	1.05232E-17	9.45242E-18	1.34259E-17	9.6752E-18	7.85184E-18
F17	1.25794E-17	9.81999E-18	8.77388E-18	9.92014E-18	7.74763E-18	7.9986E-18	1.37227E-17	1.38222E-17	1.11305E-17	7.48848E-18
F18	8.72504E-18	9.56152E-18	1.28687E-17	7.17491E-18	7.37008E-18	8.26017E-18	1.16528E-17	1.22365E-17	1.16431E-17	1.02523E-17
F19	1.09313E-17	9.1599E-18	1.23281E-17	8.40124E-18	1.19189E-17	8.36277E-18	9.66981E-18	1.14867E-17	1.25792E-17	7.63931E-18
F20	1.36332E-17	1.25473E-17	1.05058E-17	1.01459E-17	1.02231E-17	9.23076E-18	1.06592E-17	1.06752E-17	1.28435E-17	1.26824E-17

**Table 2.** The obtained P-values of the algorithms based on their best results in different executions with *Trader* as a test base. WCC: World Competitive Contests; PSO: Particles Swarm Optimization; TE: Thermal Exchange; VIR: Virulence; Dynamic Virtual Bat Algorithm; CEFOA: Co-Evolution Fruit fly Optimization; EPO: Emperor Penguin Optimizer; ION: Ion Motion; TGA: Tree Growth Algorithm; HTS: Heat Transfer Search.

labels, which show whether an intended drug and target have the interaction or not. The acquired results, which present the proposed method outperforms the other methods in the overall state, have been shown in Table 4. For every criterion on the datasets, the best-acquired outcome has been determined using the boldface value.

Figures 6 and 7 show the receiver operating characteristic (ROC) and precision-recall (PR) curves based on 5-fold cross-validation test, respectively. Besides, these data (Figs 6 and 7) represent information about the area under the curve (AUC) which compares the performance of the methods on the datasets. Except for the enzyme dataset, ANNTR achieves better results than three others. Furthermore, the proposed method obtains the average AUC values of 0.9457 and 0.9708 for ROC and PR curves, respectively, which are better than three others. Furthermore, as shown in Figs 6 and 7, RFDT, RVM, and BAY respectively obtain the average AUC values of 0.8736, 0.9216, and 0.9215 for the ROC, and 0.9248, 0.9581, and 0.9634 for the PR.

**The acquired results on the generated datasets.** In the third part of the results, we investigated the performance of *Trader* on the generated DTIs datasets (Table 1). For the testing datasets, a total of 751 (from 800 samples) DTIs were correctly predicted by the proposed method. In an evaluation with details and a comparison with other methods, we compared the proposed method with three other popular and efficient classification methods, including the support vector machine (SVM), the decision tree (DT), and the artificial neural network trained by error back propagation method (ANNEBP)<sup>15</sup>. The acquired results are shown in Table 5. Since the datasets are relating to the known DTIs, the problem is considered to be a one-class classification problem. Thus, true positive and false positive rates are reported in Table 5.

As reported in Table 5, ANNTR displays a higher detection capability of DTIs relative to the others. We used 10-fold cross-validation test in which a dataset has been divided into ten distinct sets for the comprehensive evaluation of the methods. In 10 iterations, four sets are considered as the training set; and the remaining one is used as the test set. There is also the convergence behavior of *Trader* on the generated datasets (Fig. 8).

Furthermore, we generated all the potential drug-target interactions dataset using pseudocode (Fig. 1), resulting in 119,743 records. Then, we applied the obtained models to the dataset of the potential DTIs. For all the possible DTIs dataset (119,743 samples), ANNTR predicted 47 new DTIs (Table 6).

## Discussion

The proposed machine learning ANNTR method, which is based on the new optimization algorithm, was developed and compared with the well-known and efficient machine learning methods. Then, the acquired results were analyzed. Although many optimization algorithms have been proposed, they suffer from some limitations. Our proposed algorithm, which eliminates the shortcomings of other algorithms, shows stable behavior much more than the others do, and trains the ANN appropriately. The findings also show that *Trader*, VIR, HTS, DVBA, CEFOA, ION, PSO, and WCC have better performance in comparison with TGA, TE, and EPO. The main reasons for such performance are as follows: (i) They lack any assumptions about the optimal answer to a problem with their operators, whereas TGA, TE, and EPO include operators making the range of variables smaller and smaller and therefore can reach optimal answers in a faster manner for some of the problems. (ii) Their behaviors

	Trader	WCC	PSO	TE	VIR	DVBA	CEFOA	EPO	ION	TGA	HTS
F1	<b>253.1526 ± 91.38444</b>	841.7066 ± 31.64796	477.6466 ± 18.94408	267.309375 ± 95.324573	959.4107 ± 5.94061	984.8087 ± 3.682139	950.7273 ± 11.24574	296.88529 ± 53.116245	976.356 ± 3.531292	301.05265 ± 124.06458	967.8077 ± 3.767768
F2	241.6875 ± 87.64757	839.1354 ± 34.11527	477.5999 ± 16.72475	264.262762 ± 0.2740584	960.1196 ± 5.092073	985.5377 ± 2.865412	948.5616 ± 12.12369	247.59292 ± 11.962224	977.6307 ± 2.102615	<b>24.02758 ± 0.0752910</b>	968.2255 ± 4.680692
F3	247.4043 ± 98.93201	850.3819 ± 37.43831	478.2538 ± 19.38222	235.281881 ± 0.3718979	959.8414 ± 6.465041	985.1427 ± 3.265755	948.7011 ± 10.60998	247.20669 ± 25.180205	976.7359 ± 4.132642	<b>224.04293 ± 0.0750508</b>	967.5905 ± 3.589742
F4	254.8232 ± 70.79901	847.9573 ± 42.10005	480.1671 ± 16.30009	262.355633 ± 96.352460	959.6194 ± 6.010377	985.1864 ± 3.512979	948.769 ± 11.73818	<b>74.56383 ± 26.100591</b>	976.3783 ± 3.778037	154.0341 ± 0.0900080	967.3514 ± 3.921301
F5	278.0891 ± 101.3598	848.5671 ± 37.93419	475.6191 ± 18.63996	<b>216.298908 ± 0.4138509</b>	959.3906 ± 5.015376	985.3614 ± 3.462716	952.145 ± 12.92692	366.49984 ± 17.768409	977.4732 ± 2.668033	297.03107 ± 39.083350	967.0254 ± 5.551571
F6	<b>250.9852 ± 81.59639</b>	838.3908 ± 35.51018	479.2404 ± 17.91999	299.244924 ± 102.31963	959.7162 ± 5.834993	985.3645 ± 3.358163	946.9678 ± 12.91631	305.46687 ± 67.032709	977.2192 ± 3.438642	300.01504 ± 49.081595	967.4208 ± 4.269639
F7	251.3358 ± 108.8897	841.1641 ± 39.66158	479.4888 ± 14.50753	<b>2.367119 ± 0.3282396</b>	961.0597 ± 6.120448	985.2691 ± 4.110173	948.9477 ± 11.33569	47.58055 ± 2.014534	976.1429 ± 3.420905	24.02266 ± 0.0841355	967.9891 ± 3.429372
F8	253.9236 ± 92.58586	846.1772 ± 39.51895	476.0835 ± 16.47731	29.276181 ± 3.40998	959.46 ± 6.371902	984.5531 ± 3.213782	949.5816 ± 11.30967	47.08275 ± 2.257234	977.3502 ± 2.663908	<b>24.03074 ± 0.0772274</b>	967.606 ± 4.560412
F9	260.6851 ± 102.8401	855.9507 ± 29.35375	479.3728 ± 21.17523	260.311927 ± 25.333143	959.9554 ± 5.546612	984.9158 ± 3.203462	947.1605 ± 13.9521	<b>165.73816 ± 21.956847</b>	977.6 ± 3.671616	233.0349 ± 44.073838	967.3254 ± 4.316765
F10	<b>1.7736e-32 ± 1.253e-31</b>	145.9045 ± 83.05286	29.76548 ± 5.045027	4.743712 ± 0.6323218	525.0646 ± 51.02185	288.105 ± 35.46705	51.30359 ± 11.50775	13.47461 ± 1.575858	99.40116 ± 15.27536	17.35812 ± 1.41808	74.66727 ± 10.12832
F11	5.1198e-72 ± 2.634e-71	0.03253717 ± 0.0231875	2.2715e-08 ± 1.502e-08	1.025448 ± 0.245046	0.2861472 ± 0.2577325	0.06544539 ± 0.0364154	0.00349084 ± 0.0016383	0.00548362 ± 0.0031898	0.01487591 ± 0.0094282	0.00577078 ± 0.0053602	<b>0 ± 0</b>
F12	<b>0 ± 0</b>	0.00220911 ± 0.0021401	1.7413e-09 ± 1.302e-09	0.08037839 ± 0.0154615	0.02280964 ± 0.0188935	0.00475016 ± 0.0028673	0.00034019 ± 0.0001791	0.00035483 ± 0.0001949	0.00093730 ± 0.0005838	0.00057121 ± 0.0006454	<b>0 ± 0</b>
F13	<b>4.8004e-63 ± 2.968e-62</b>	0.00166645 ± 0.0016165	7.6484e-09 ± 3.888e-09	0.01628576 ± 0.001229	0.01011512 ± 0.0110870	0.00235245 ± 0.000852	0.00022412 ± 9.119e-05	0.00036907 ± 0.0001605	6.8835e-05 ± 8.987e-05	0.00047189 ± 0.0004179	5.928e-12 ± 0
F14	<b>4.6657e-36 ± 3.262e-35</b>	147.8289 ± 92.29859	30.23713 ± 4.95025	4.773464 ± 0.6929691	508.2296 ± 54.1219	281.2291 ± 45.63714	52.49834 ± 14.60957	13.56846 ± 1.401647	101.3417 ± 17.26803	17.00028 ± 1.862734	76.718 ± 10.14181
F15	258.9922 ± 107.4624	852.0461 ± 41.73354	480.807 ± 19.48127	222.342933 ± 0.3350069	960.0506 ± 5.331034	984.5826 ± 3.527149	945.3247 ± 13.84791	<b>47.6192 ± 2.178289</b>	977.731 ± 2.914132	334.02932 ± 0.0848916	967.8714 ± 4.844288
F16	<b>2.3816e-32 ± 1.426e-31</b>	157.1937 ± 88.66653	30.07814 ± 5.658763	4.887573 ± 0.5766644	524.5626 ± 47.10672	278.6976 ± 36.47537	54.55871 ± 16.3804	13.60383 ± 1.502604	101.7394 ± 17.00099	17.07861 ± 1.607762	74.4641 ± 11.54133
F17	<b>5.3552e-28 ± 3.786e-27</b>	169.7363 ± 88.32107	30.85688 ± 5.256397	4.706465 ± 0.6551469	515.5718 ± 53.29598	278.9005 ± 35.72971	53.5587 ± 13.27707	13.91615 ± 1.720058	98.84173 ± 16.28048	17.30378 ± 1.511352	78.45175 ± 11.01959
F18	1.0635e-28 ± 7.520e-28	151.4619 ± 89.5036	29.90848 ± 5.2751	4.76432 ± 0.7214757	499.9788 ± 55.48298	282.8907 ± 42.3109	54.82297 ± 11.80817	13.47419 ± 1.497829	98.37421 ± 15.05355	17.385 ± 1.539785	<b>0 ± 0</b>
F19	<b>4.6354e-35 ± 2.960e-34</b>	156.6053 ± 61.30125	30.35593 ± 5.681104	4.930016 ± 0.9168929	516.1155 ± 47.98521	278.2041 ± 38.58286	52.01463 ± 10.74207	13.48662 ± 1.561745	99.91899 ± 16.78486	17.39819 ± 1.618368	73.81935 ± 10.6238
F20	<b>1.4363e-33 ± 8.344e-33</b>	161.686 ± 91.38573	30.15713 ± 5.526973	4.607396 ± 0.487521	511.36 ± 55.23571	285.4784 ± 40.22137	53.02054 ± 15.04198	13.6743 ± 1.550832	97.17674 ± 14.64385	17.10843 ± 1.49111	78.12226 ± 10.41774

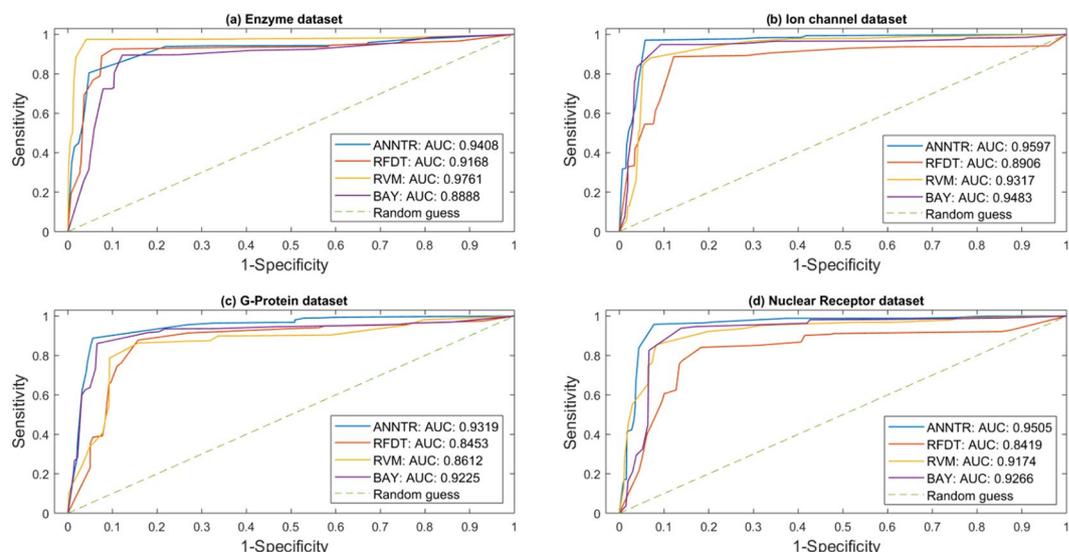
**Table 3.** The obtained mean and standard deviation values of the algorithms with [mean] ± [standard deviation] pattern. WCC: World Competitive Contests; PSO: Particles Swarm Optimization; TE: Thermal Exchange; VIR: Virulence; DVBA: Dynamic Virtual Bat Algorithm; CEFOA: Co-Evolution Fruit fly Optimization; EPO: Emperor Penguin Optimizer; ION: Ion Motion; TGA: Tree Growth Algorithm; HTS: Heat Transfer Search.

	Enzyme				Ion channel				G-protein coupled receptor				Nuclear receptor				Average			
	ACC	SEN	SPC	PRE	ACC	SEN	SPC	PRE	ACC	SEN	SPC	PRE	ACC	SEN	SPC	PRE	ACC	SEN	SPC	PRE
ANNTR	94.2	92.92	95.24	94.46	<b>96.2</b>	<b>96.61</b>	<b>95.32</b>	<b>95.74</b>	<b>93.5</b>	<b>92.15</b>	<b>94.24</b>	<b>93.17</b>	94.6	95.28	<b>94.43</b>	<b>93.06</b>	<b>94.62</b>	<b>94.24</b>	<b>94.80</b>	<b>94.10</b>
RFDT	91.3	92.02	91.34	92.56	89.1	88.92	89.21	89.46	84.1	84.21	84.86	85.21	71.1	71.16	71.88	70.13	83.09	84.07	84.32	84.34
RVM	<b>97.73</b>	<b>97.44</b>	<b>97.78</b>	<b>98.01</b>	93.12	93.32	93.02	92.96	86.78	84.89	87.36	87.91	87.78	92.63	87.51	85.19	91.35	92.07	91.41	91.01
BAY	89.04	88.73	89.04	90.52	95.3	94.47	95.19	94.44	92.64	91.26	93.25	92.92	<b>94.8</b>	<b>95.37</b>	94.06	92.66	92.94	92.45	92.88	92.63

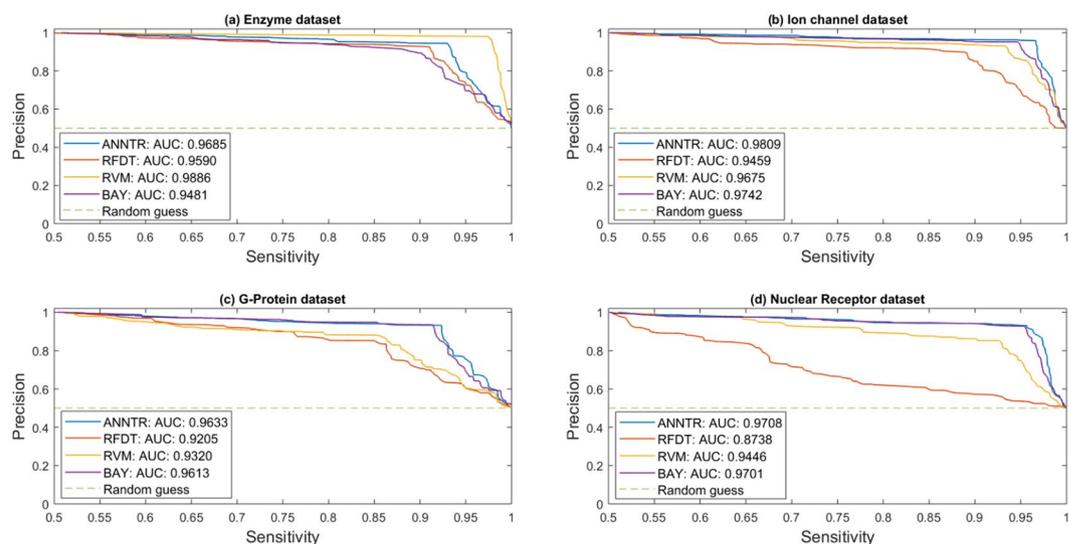
**Table 4.** A comprehensive comparison between the 5-fold cross-validation results of the proposed method and the others. ANNTR: Trader-based Artificial Neural network; RFDT: Rotation Forest-based Drug-Target predictor; RVM: Relevance Vector Machine; BAY: Bayesian ranking-based; ACC: Accuracy; SEN: Sensitivity; SPC: Specificity; PRE: Precision.

are almost the same on the different benchmark functions, whereas TE, EPO, and TGA fall into local optima positions for some test functions.

As a case study, we applied the proposed ANNTR method to some biological datasets and all the potential DTIs data to find drugs which may affect targets, with the result that 47 DTIs were discovered. The predicted results can be used in three manners. First, they propose some unknown DTIs. In case a disease is due to the



**Figure 6.** The ROC curve of the methods on the four gold-standard datasets. (a) The ROC curves of the algorithms on the enzyme dataset. (b) The ROC curves of the algorithms on the ion channel dataset. (c) The ROC curves of the algorithms on the G-protein dataset. (d) The ROC curves of the algorithms on the nuclear receptor dataset. Besides the four plots, there are also the values of the AUC. Except for the enzyme dataset, the proposed method has obtained better results than others. Furthermore, *Trader's* average value of the AUC is higher than four others. ANNTR: *Trader*-based Artificial Neural network; RFDT: Rotation Forest-based Drug-Target predictor; RVM: Relevance Vector Machine; BAY: Bayesian ranking-based.

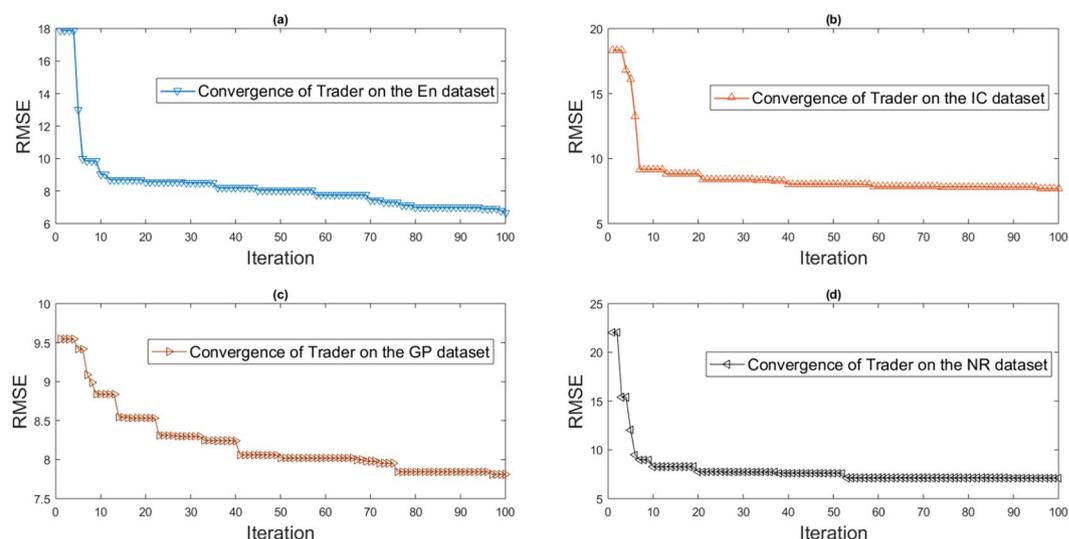


**Figure 7.** The PR curve of the methods on the gold-standard datasets. (a) The PR curves of the algorithms on the enzyme dataset. (b) The PR curves of the algorithms on the ion channel dataset. (c) The PR curves of the algorithms on the G-protein dataset. (d) The PR curves of the algorithms on the nuclear receptor dataset. The size of the positive and negative datasets is the same. The PR curves show the proper performance of the proposed method relative to the others. The average value of *Trader's* AUS is also higher than them. ANNTR: *Trader*-based Artificial Neural network; RFDT: Rotation Forest-based Drug-Target predictor; RVM: Relevance Vector Machine; BAY: Bayesian ranking-based.

intended target, the related drug can be introduced as an option for the treatment of disease. Further, the side effects of drugs can be determined by investigating the predicted relation between a drug and a target. Second, they show that some of the medications, like D00086 and D00145, have an identical predicted target. There is a possibility that they have similar functionality and can be used alternatively. Also, these results can be useful to chemical pharmacists who look for the novel potential efficacy of drugs and researchers who want to validate their predicted outcomes. Third, the predicted DTIs might reveal the real mechanism of actions (MOA) of drugs<sup>49</sup> which show the pharmacological effect(s) of a drug.

Dataset	Method	True positive	False negative	Accuracy
EN	ANNTR	406	94	81.20
	SVM	306	194	61.20
	DT	255	245	51.00
	ANNEBP	294	206	58.80
IC	ANNTR	241	59	80.33
	SVM	203	97	67.67
	DT	151	149	50.33
	ANN_EBP	189	111	63
GP	ANNTR	87	13	87
	SVM	72	28	72
	DT	57	43	57
	ANNEBP	75	25	75
NR	ANNTR	16	4	80
	SVM	15	5	75
	DT	13	7	65
	ANNEBP	15	5	75

**Table 5.** Acquired results using 10-fold cross-validation test on the generated datasets. ANNTR: *Trader*-based Artificial Neural Network; ANN: Artificial neural network; SVM: Support Vector Machine; DT: Decision Tree; ANNEBP: ANN based on Error Back-Propagation.



**Figure 8.** The convergence behavior of *Trader* on all the generated datasets in training of the ANN. (a) The Convergence of *Trader* on the EN dataset. (b) The Convergence of *Trader* on the IC dataset. (c) The Convergence of *Trader* on the GP dataset. (d) The Convergence of *Trader* on the NR dataset. The results relate to the best-obtained outcomes from 50 distinct executions. For all the datasets, *Trader* has led to an acceptable value of the RMSE.

For example, we predicted that diazoxide interacts with the angiotensin-I-converting enzyme (ACE). An investigation in the clinical impact of diazoxide and ACE together with some previous studies reveals that diazoxide can be used for the treatment of severe hypertension<sup>50</sup>, while ACE is responsible for controlling blood pressure<sup>51</sup>. Therefore, for the first time, we have shown that diazoxide can affect the ACE, while the MOA of diazoxide has been reported differently by others<sup>52</sup>. Also, similar to diazoxide, Ketotifen, used to reduce conjunctivitis allergic effects, can also interact with the ACE. Likewise, in a study conducted by Sanchez-Patan *et al.*<sup>53</sup>, Ketotifen was shown to decrease hypertension in rats.

Another example is erlotinib which is used for treating epithelial lung cancer. Our proposed method has predicted that erlotinib interacts with Muscle, skeletal, receptor tyrosine kinase (MuSK) which its antibodies are found in neuromuscular diseases. The disease leads to various phenotypes such as less eye involvement, weakness, and pain in the neck. Some researches related to the side effects of ertolinib<sup>54</sup> may validate the predicted interaction.

No	Drug ID	Drug name	Target ID	Target function
1	D00086	Thimerosal	hsa5152	Phosphodiesterase 9A [EC:3.1.4.17]
2	D00145	Trimethoprim	hsa5152	Phosphodiesterase 9A [EC:3.1.4.17]
3	D00160	Epsilon-Aminocaproic acid	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
4	D00169	Meclofenamate sodium	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
5	D00227	Aminophylline	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
6	D00231	Amrinone	hsa3156	3-hydroxy-3-methylglutaryl-Coenzyme A reductase [EC:1.1.1.34]
7	D00294	Diazoxide	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
8	D00325	Fluocinonide	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
9	D00380	Tolbutamide	hsa43	Acetylcholinesterase (Yt blood group) [EC:3.1.1.7]
10	D00394	Trimipramine	hsa231	Aldo-keto reductase family 1, member B1 (aldose reductase)
11	D00394	Trimipramine	hsa239	Arachidonate 12-lipoxygenase [EC:1.13.11.31]
12	D00394	Trimipramine	hsa242	Arachidonate 12-lipoxygenase, 12 R type [EC:1.13.11.-]
13	D00410	Metypapone	hsa246	Arachidonate 15-lipoxygenase [EC:1.13.11.33]
14	D00437	Nifedipine	hsa247	Arachidonate 15-lipoxygenase, type B [EC:1.13.11.33]
15	D00451	Sumatriptan	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
16	D00459	Quinapril hydrochloride	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
17	D00475	Probenecid	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
18	D00505	Phenelzine sulfate	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
19	D00566	Sodium salicylate	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
20	D00577	Diethylstilbestrol	hsa43	Acetylcholinesterase (Yt blood group) [EC:3.1.1.7]
21	D00596	Rosiglitazone maleate	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
22	D00623	Moexipril hydrochloride	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
23	D00650	Bendroflumethiazide	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
24	D00749	Leflunomide	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
25	D00781	Entacapone	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
26	D00813	Ketorolac tromethamine	hsa231	Aldo-keto reductase family 1, member B1 (aldose reductase)
27	D00885	Oxiconazole nitrate	hsa239	Arachidonate 12-lipoxygenase [EC:1.13.11.31]
28	D00960	Anastrozole	hsa476	ATPase, Na <sup>+</sup> /K <sup>+</sup> transporting, alpha 1 polypeptide [EC:3.6.3.9]
29	D00969	Meloxicam	hsa242	Arachidonate 12-lipoxygenase, 12 R type [EC:1.13.11.-]
30	D01276	Atazanavir sulfate	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
31	D01332	Ketotifen fumarate	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
32	D01364	Ciclopirox olamine	hsa246	Arachidonate 15-lipoxygenase [EC:1.13.11.33]
33	D01811	Salicylamide	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
34	D02290	Flurbiprofen sodium	hsa247	Arachidonate 15-lipoxygenase, type B [EC:1.13.11.33]
35	D02323	Tolrestat	hsa3156	3-hydroxy-3-methylglutaryl-Coenzyme A reductase [EC:1.1.1.34]
36	D02375	Terbinafine	hsa43	Acetylcholinesterase (Yt blood group) [EC:3.1.1.7]
37	D02451	Fadrozole hydrochloride	hsa5152	Phosphodiesterase 9A [EC:3.1.4.17]
38	D02559	Toloxatone	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
39	D02563	Befloxatone	hsa43	Acetylcholinesterase (Yt blood group) [EC:3.1.1.7]
40	D03601	Crivastatin	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
41	D03689	Deracoxib	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
42	D03717	Parecoxib sodium	hsa4593	Muscle, skeletal, receptor tyrosine kinase [EC:2.7.10.1]
43	D03787	Nepicastat hydrochloride	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
44	D03806	Ponalrestat	hsa476	ATPase, Na <sup>+</sup> /K <sup>+</sup> transporting, alpha 1 polypeptide [EC:3.6.3.9]
45	D04023	Erlotinib hydrochloride	hsa4593	Muscle, skeletal, receptor tyrosine kinase [EC:2.7.10.1]
46	D05341	Palmitic acid	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1
47	D06238	Trimetrexate	hsa1636	Angiotensin I converting enzyme (peptidyl-dipeptidase A) 1

**Table 6.** The detected drug-target interactions.

## Conclusion

A new optimization algorithm, named *Trader*, was introduced and compared with ten state-of-art optimization algorithms based on various statistical criteria. The results show that *Trader* outperforms other optimization algorithms and eliminates their limitations. As an empirical, yet smart evaluation, we examined the performance of *Trader* in the training of a multi-layer perceptron artificial neural network to discover potential DTIs on the gold-standard and generated datasets. The predicting model obtained from *Trader* achieved 94.62%, 94.24%, 94.80%, and 94.10% of the average 5-fold cross-validation respectively for the accuracy, sensitivity, specificity, and precision of the model. These values appeared to be better than the acquired results from other methods. Furthermore, the proposed method predicted 47 potential DTIs. We envision that the outcomes obtained by

the proposed model may be used for managing possible side-effects of medications, understanding the MOA of drugs, and finding new research opportunities. Taken all, this study may pave the way in terms of de novo applications of computer-aided methods in drug discovery and development.

## Data Availability

All the source codes and the datasets are available in the following link: <https://github.com/LBBSofT/Trader>.

## References

- Csermely, P., Korcsmáros, T., Kiss, H. J., London, G. & Nussinov, R. Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. *Pharmacology & therapeutics* **138**, 333–408 (2013).
- Luo, H., Mattes, W., Mendrick, D. L. & Hong, H. Molecular docking for identification of potential targets for drug repurposing. *Current topics in medicinal chemistry* **16**, 3636–3645 (2016).
- Wu, Z., Wang, Y. & Chen, L. Network-based drug repositioning. *Molecular BioSystems* **9**, 1268–1281 (2013).
- Qu, X. A. & Rajpal, D. K. Applications of Connectivity Map in drug discovery and development. *Drug discovery today* **17**, 1289–1298 (2012).
- Zhang, M., Luo, H., Xi, Z. & Rogaeva, E. Drug repositioning for diabetes based on omics' data mining. *PLoS one* **10**, e0126082 (2015).
- You, J., McLeod, R. D. & Hu, P. Predicting Drug-Target Interaction Network Using Deep Learning Model. *Computational Biology and Chemistry* (2019).
- Xie, L., He, S., Song, X., Bo, X. & Zhang, Z. Deep learning-based transcriptome data classification for drug-target interaction prediction. *BMC genomics* **19**, 667 (2018).
- Ho, Q.-T., Phan, D.-V. & Ou, Y.-Y. Using word embedding technique to efficiently represent protein sequences for identifying substrate specificities of transporters. *Analytical Biochemistry* (2019).
- Song, D. *et al.* Similarity-based machine learning support vector machine predictor of drug-drug interactions with improved accuracies. *Journal of clinical pharmacy and therapeutics* **44**, 268–275 (2019).
- Keum, J. & Nam, H. Self-blmm: Prediction of drug-target interactions via self-training svm. *PLoS one* **12**, e0171839 (2017).
- Wang, L. *et al.* RfDT: A rotation forest-based predictor for predicting drug-target interactions using drug structure and protein sequence information. *Current Protein and Peptide Science* **19**, 445–454 (2018).
- Cichonska, A. *et al.* Computational-experimental approach to drug-target interaction mapping: a case study on kinase inhibitors. *PLoS computational biology* **13**, e1005678 (2017).
- Cai, C. *et al.* In silico prediction of ROCK II inhibitors by different classification approaches. *Molecular diversity* **21**, 791–807 (2017).
- Meng, F.-R., You, Z.-H., Chen, X., Zhou, Y. & An, J.-Y. Prediction of drug-target interaction networks from the integration of protein sequences and drug chemical structures. *Molecules* **22**, 1119 (2017).
- Masoudi-Sobhanzadeh, Y., Motieghader, H. & Masoudi-Nejad, A. FeatureSelect: a software for feature selection based on machine learning approaches. *BMC bioinformatics* **20**, 170 (2019).
- Lee, I. & Nam, H. Identification of drug-target interaction by a random walk with restart method on an interactome network. *BMC bioinformatics* **19**, 208 (2018).
- Yan, X.-Y., Zhang, S.-W. & He, C.-R. Prediction of drug-target interaction by integrating diverse heterogeneous information source with multiple kernel learning and clustering methods. *Computational biology and chemistry* **78**, 460–467 (2019).
- He, L. *et al.* Patient-customized drug combination prediction and testing for t-cell prolymphocytic leukemia patients. *Cancer research* **78**, 2407–2418 (2018).
- Zheng, Y. *et al.* Predicting adverse drug reactions of combined medication from heterogeneous pharmacologic databases. *BMC bioinformatics* **19**, 517 (2018).
- Lu, Y., Guo, Y. & Korhonen, A. Link prediction in drug-target interactions network using similarity indices. *BMC bioinformatics* **18**, 39 (2017).
- Ji, X., Freudenberg, J. M. & Agarwal, P. In *Computational Methods for Drug Repurposing* 203–218 (Springer, 2019).
- Peska, L., Buza, K. & Koller, J. Drug-target interaction prediction: A Bayesian ranking approach. *Computer methods and programs in biomedicine* **152**, 15–21 (2017).
- Ezzat, A., Zhao, P., Wu, M., Li, X.-L. & Kwok, C.-K. Drug-target interaction prediction with graph regularized matrix factorization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* **14**, 646–656 (2017).
- Gu, W., Xie, X., He, Y. & Zhang, Z. Drug-target protein interaction prediction based on AdaBoost algorithm. *Sheng wu yi xue gong cheng xue za zhi = Journal of biomedical engineering = Shengwu yixue gongchengxue zazhi* **35**, 935–942 (2018).
- Ezzat, A., Wu, M., Li, X. & Kwok, C.-K. In *Computational Methods for Drug Repurposing* 239–254 (Springer, 2019).
- Sharma, A. & Rani, R. BE-DTI: Ensemble framework for drug target interaction prediction using dimensionality reduction and active learning. *Computer methods and programs in biomedicine* **165**, 151–162 (2018).
- Rifaioğlu, A. S. *et al.* Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases. *Brief. Bioinform* **10** (2018).
- Ding, Y., Tang, J. & Guo, F. The computational models of drug-target interaction prediction. *Protein and peptide letters* (2019).
- Lai, H.-Y. *et al.* A Brief Survey of Machine Learning Application in Cancerlectin Identification. *Current gene therapy* **18**, 257–267 (2018).
- Zhang, W. *et al.* Recent Advances in the Machine Learning-Based Drug-Target Interaction Prediction. *Current drug metabolism* (2019).
- Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W. & Kanehisa, M. Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* **24**, i232–i240 (2008).
- Kanehisa, M. & Goto, S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic acids research* **28**, 27–30 (2000).
- Wishart, D. S. *et al.* DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research* **34**, D668–D672 (2006).
- Masoudi-Sobhanzadeh, Y., Omidi, Y., Amanlou, M. & Masoudi-Nejad, A. DrugR+: A comprehensive relational database for drug repurposing, combination therapy, and replacement therapy. *Computers in Biology and Medicine* (2019).
- Smith, T. F. & Waterman, M. S. Comparison of biosequences. *Advances in applied mathematics* **2**, 482–489 (1981).
- Coello Coello, C. & Lechuga, M. In *Proc., Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*. 1051–1056.
- Masoudi-Sobhanzadeh, Y. & Motieghader, H. World Competitive Contests (WCC) algorithm: A novel intelligent optimization algorithm for biological and non-biological problems. *Informatics in Medicine Unlocked* **3**, 15–28 (2016).
- Cheraghalipour, A., Hajiaghahi-Keshteli, M. & Paydar, M. M. Tree Growth Algorithm (TGA): A novel approach for solving optimization problems. *Engineering Applications of Artificial Intelligence* **72**, 393–414 (2018).
- Kaveh, A. & Dadras, A. A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Advances in Engineering Software* **110**, 69–84 (2017).
- Dhiman, G. & Kumar, V. Emperor Penguin Optimizer: A Bio-inspired Algorithm for Engineering Problems. *Knowledge-Based Systems* (2018).

41. Javidy, B., Hatamlou, A. & Mirjalili, S. Ions motion algorithm for solving optimization problems. *Applied Soft Computing* **32**, 72–79 (2015).
42. Jaderyan, M. & Khotanlou, H. Virulence Optimization Algorithm. *Applied Soft Computing* **43**, 596–618 (2016).
43. Topal, A. O. & Altun, O. A novel meta-heuristic algorithm: Dynamic Virtual Bats Algorithm. *Information Sciences* **354**, 222–235 (2016).
44. Patel, V. K. & Savsani, V. J. Heat transfer search (HTS): a novel optimization algorithm. *Information Sciences* **324**, 217–246 (2015).
45. Han, X., Liu, Q., Wang, H. & Wang, L. Novel fruit fly optimization algorithm with trend search and co-evolution. *Knowledge-Based Systems* **141**, 1–17 (2018).
46. Mernik, M., Liu, S.-H., Karaboga, D. & Črepinšek, M. On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation. *Information Sciences* **291**, 115–127 (2015).
47. Abdi, Y. & Seyfari, Y. Search Manager: A Framework for Hybridizing Different Search Strategies. *International journal of advanced computer science and applications* **9**, 525–540 (2018).
48. Yamanishi, Y., Kotera, M., Kanehisa, M. & Goto, S. Drug–target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics* **26**, i246–i254 (2010).
49. Baker, N. C., Ekins, S., Williams, A. J. & Tropsha, A. A bibliometric review of drug repurposing. *Drug discovery today* (2018).
50. Sridharan, K. & Sequeira, R. P. Drugs for treating severe hypertension in pregnancy: a network meta-analysis and trial sequential analysis of randomized clinical trials. *British journal of clinical pharmacology* **84**, 1906–1916 (2018).
51. Tartibian, B., Botelho Teixeira, A. M. & Baghaiee, B. Moderate Intensity Exercise is Associated With Decreased Angiotensin-converting Enzyme, Increased  $\beta$ 2-adrenergic Receptor Gene Expression, and Lower Blood Pressure in Middle-Aged Men. *Journal of aging and physical activity* **23**, 212–220 (2015).
52. Altszuler, N., Hampshire, J. & Moraru, E. On the mechanism of diazoxide-induced hyperglycemia. *Diabetes* **26**, 931–935 (1977).
53. Sánchez-Patán, F. *et al.* Mast cell inhibition by ketotifen reduces splanchnic inflammatory response in a portal hypertension model in rats. *Experimental and Toxicologic Pathology* **60**, 347–355 (2008).
54. Celik, T. & Kosker, M. Ocular side effects and trichomegaly of eyelashes induced by erlotinib: a case report and review of the literature. *Contact Lens and Anterior Eye* **38**, 59–60 (2015).

### Author Contributions

Yosef Masoudi-Sobhanzadeh: Conceptualization, implementation, formal analysis, investigation, writing, editing, and revising the manuscript. Yadollah Omidi: Results analysis, validation, Conceptualization, writing, editing, and revising the manuscript. Massoud Amanlou: Validation, data analysis, Editing-manuscript. Ali Masoudi-Nejad: Conceptualization, Supervision, Project administration, writing, editing, and revising the manuscript. All authors have read and approved the manuscript.

### Additional Information

**Supplementary information** accompanies this paper at <https://doi.org/10.1038/s41598-019-45814-8>.

**Competing Interests:** The authors declare no competing interests.

**Publisher's note:** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2019