Compounds Activity Prediction in Large Imbalanced Datasets with Substructural Relations Fingerprint and EEM

Wojciech Marian Czarnecki and Krzysztof Rataj

***Abstract*—Modern drug design procedures involve the process of virtual screening, a highly efficient filtering step used for maximizing the efficiency of the preselection of compounds which are valuable drug candidates. Recent advances in introduction of machine learning models to this process can lead to significant increase in the overall quality of the drug designing pipeline.**

Unfortunately, for many proteins it is still extremely hard to come up with a valid statistical model. It is a consequence of huge classes disproportion (even 1000:1), large datasets (over 100,000 of samples) and restricted data representation (mostly high-dimensional, sparse, binary vectors). In this paper, we try to tackle this problem through three important innovations. First we represent compounds with 2-dimensional, graph representation. Second, we show how one can provide extremely fast method for measuring similarity of such data. Finally, we use the Extreme Entropy Machine which shows increase in balanced accuracy over Extreme Learning Machines, Support Vector Machines, one-class Support Vector Machines as well as Random Forest.

Proposed pipeline brings significantly better results than all considered alternative, state-of-the-art approaches. We introduce some important novel elements and show why they lead to better model. Despite this, it should still be considered as a proof of concept and further investigations in the field are needed.

***Keywords*—substructural relations fingerprint, compounds activity, virtual screening, Tanimoto similarity, entropy.**

I. Introduction

PREDICTION of chemical compounds biological activity is a crucial element of modern virtual screening techniques. In order to find drugs candidates, one often tries to model the concept of molecule activity towards given protein so it can be used for filtering the huge databases of compounds.

While many researchers claim successful application of machine learning (ML) methods in such task, using for example Support Vector Machines (SVM), it is still a challenging problem due to various characteristics of the data. In particular, one has to deal with huge databases of compounds, which are extremely imbalanced [1], gathered without iid assumption, without generally accepted data representation [2].

In this paper, we try to solve this problem by performing experiments on 16 proteins with highest number of known ligands and compounds of confirmed inactivity (dozens and hundreds of thousands). We focus on three elements of the classification process – data representation, model and finally similarity measure-based selection. We show a combination of three novel approaches combined to achieve best scores in evaluation consisting of 22 models, which might not be definitive but provides the proof of concept of the working pipeline in a hard setting of massive, imbalanced dataset of compounds activity prediction.

W. M. Czarnecki is with the Faculty of Mathematics and Computer Science, Jagiellonian University, Krakow, Poland, e-mail: wojciech.czarnecki@uj.edu.pl.

K. Rataj is with Institute of Pharmacology, Polish Academy of Sciences, Krakow, Poland.

Manuscript received April 19, 2005; revised January 11, 2007.

II. Problem and general idea

The main problem considered in this research is the activity prediction which is here seen as a binary classification problem of the chemical compounds. There are some important features of the problem which need to be considered:

- structures of chemical compounds need to be somehow represented in order to use ML methods;
- datasets are highly imbalanced and are collected with violation of iid assumption made by most of the statistical learning models;
- positive class is rarely a compact part of the feature space, rather it seems to be nearly randomly spread across the space which is completely filled with negative samples (see Figure 1);

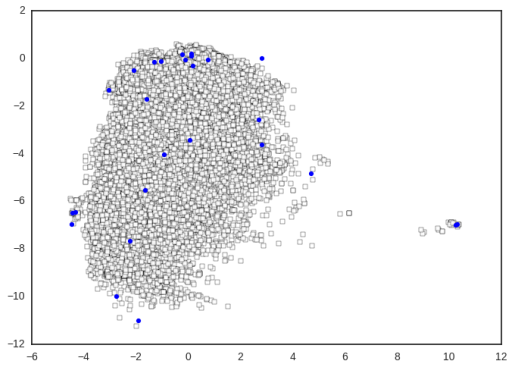

Fig. 1. PCA visualization of Q13148 ligands and known inactive compounds.

One of the most successful models in such problems is SVM. In the next section we show some alternative model, which, as shown in further parts of the paper, can overcome

important limitations of SVM and result in models with much bigger predictive power.

In particular, we want to show that one can efficiently represent molecules as heavily compressed labeled graphs, which followed by a particular random projections can be well modeled by the robust model, called Extreme Entropy Machine (EEM). In other words, we are going to propose a computationally cheap method of representing, measuring similarity and finally classifying specific labeled graph structures, representing chemical molecules.

III. Extreme Entropy Machine

One of the recently proposed alternatives for SVM model is Extreme Entropy Machine [3] (EEM). This model can be seen as a fusion of Random Vector Functional-Link Networks [4] (or very closely related Extreme Learning Machines [5], ELM) and entropy maximization techniques such as Multithreshold Entropy Linear Classifier [6]. Their major advantage is an ability to use arbitrary similarity measures (as opposed to kernels) and the closed form solution of the optimization process, which makes it scale well. For a given dataset of binary labeled points consisting of positive samples $\mathbf{X}^+$, negative $\mathbf{X}^-$, Ledoit-Wolf covariance estimator $\mathrm{cov}_\dagger$ [7] and random projection φ, the EEM optimization problem is defined in a following way.

Extreme Entropy Machine

$$\begin{aligned}
\underset{\boldsymbol{\beta}}{\text{minimize}} \quad & \boldsymbol{\beta}^T\boldsymbol{\Sigma}^+\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{\Sigma}^-\boldsymbol{\beta} \\
\text{subject to} \quad & \boldsymbol{\beta}^T(\mathbf{m}^+ - \mathbf{m}^-) = 2 \\
\text{where} \quad & \boldsymbol{\Sigma}^\pm = \mathrm{cov}_\dagger(\mathbf{H}^\pm) \\
& \mathbf{m}^\pm = \frac{1}{|\mathbf{H}^\pm|} \sum_{\mathbf{h}^\pm \in \mathbf{H}^\pm} \mathbf{h}^\pm \\
& \mathbf{H}^\pm = \varphi(\mathbf{X}^\pm)
\end{aligned}$$

As it is derived from the generative model assuming normal distribution in the projected space, the classification is later based on comparing two pdfs:

$$cl(\mathbf{x}) = \arg \max_{t \in \{-1,+1\}} \mathcal{N}(\mathbf{m}^l\boldsymbol{\beta}, \boldsymbol{\beta}^T\boldsymbol{\Sigma}^l\boldsymbol{\beta})[\varphi(\mathbf{x})\boldsymbol{\beta}].$$

For the best generalization capabilities one often needs control over regularization of given model. The above approach can be parametrized as suggested by Czarnecki et al. [3]. The Ledoit-Wolf covariance estimator is a shrinkage estimator, being a convex combination of empirical covariance and identity matrix. Such approach gives parameter-less model with guaranteed invertability of the estimator. We can further transform it towards identity matrix by putting $\boldsymbol{\Sigma}^\pm_C = \mathrm{cov}_\dagger(\mathbf{H}^\pm) + \frac{1}{2C}\mathbf{I}$ instead of $\boldsymbol{\Sigma}^\pm = \mathrm{cov}_\dagger(\mathbf{H}^\pm)$. Consequently we have an objective function

$$\begin{aligned}
&\boldsymbol{\beta}^T\boldsymbol{\Sigma}^+_C\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{\Sigma}^-_C\boldsymbol{\beta} = \\
&\boldsymbol{\beta}^T(\boldsymbol{\Sigma}^+ + \tfrac{1}{2C}\mathbf{I})\boldsymbol{\beta} + \boldsymbol{\beta}^T(\boldsymbol{\Sigma}^- + \tfrac{1}{2C}\mathbf{I})\boldsymbol{\beta} = \\
&\boldsymbol{\beta}^T\boldsymbol{\Sigma}^+\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{\Sigma}^-\boldsymbol{\beta} + \boldsymbol{\beta}^T(\tfrac{1}{2C}\mathbf{I} + \tfrac{1}{2C}\mathbf{I})\boldsymbol{\beta} = \\
&\boldsymbol{\beta}^T\boldsymbol{\Sigma}^+\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{\Sigma}^-\boldsymbol{\beta} + \tfrac{1}{C}\boldsymbol{\beta}^T\boldsymbol{\beta}.
\end{aligned}$$

While it might be seen as a simple L^2 regularization of linear model weights, we prefer to look at it as a strength of covariance smoothing, which gives analogous, pdf based classification rule:

$$cl(\mathbf{x}) = \arg \max_{t \in \{-1,+1\}} \mathcal{N}(\mathbf{m}^l\boldsymbol{\beta}, \boldsymbol{\beta}^T(\boldsymbol{\Sigma}^l + \tfrac{1}{2C}\mathbf{I})\boldsymbol{\beta})[\varphi(\mathbf{x})\boldsymbol{\beta}],$$

and a training procedure is exactly the same as the original EEM, we only change the definition of covariance.

EEM, SVM and nearly any other machine learning model heavily depends on the data representation and similarity measure used by the model (either stated explicitly through definition of the metric like in KNN, or indirectly through definition of kernel space in SVM or random projection in RVFL or EEM). We now focus on these two elements of our problem – how to represent and measure similarity of two compounds in a highly efficient manner.

IV. Two-dimensional fingerprints and similarity

There are many types of fingerprints (fixed length chemical compounds representations) [2], however, one of the simplest and fastest to compute are substructural ones (like SubFP). In such an approach we have a binary[1] vector which simply encodes presence of a particular substructure in a given compound. In other words, given compound c we define a substructural fingerprint fp through sequence of structures $s_1, ..., s_d$.

$$\mathrm{fp}(c) = [1_{s_1 \in c}, \ldots 1_{s_d \in c}]^T \in \{0,1\}^d,$$

where 1_p is a characteristic function equal 1 iff p is true. One of the main advantages of using such representation is its simplicity, easy interpretation of the results as well as speed of generation. However, such method limits practical applications, in particular, results obtained by direct application of ML methods often yield weak results [8]. The need for more complex representations is undeniable, but at the same time their computation should be rapid so developed solution can scale up to millions (and above) of structures.

One of the alternative approaches to substructural fingerprinting is to consider graph representation of compounds. Unfortunately, for general graphs it is quite hard to define similarity measures. While there exist very successful graph kernels [9] which can be plugged into many kernel methods [10], they main shortcoming is their computational complexity which scales at least quadratic with the size of the graph [11] to even sixth power [9]. Furthermore, these methods are just approximations of underlying similarity ideas, as even finding the biggest common subgraph of two graphs is a NP-hard problem. We focus on a technique, which can connect such an approach of deeper understanding of compound structure with simplicity and effectiveness of fingerprints approach, namely the substructural relations fingerprint (SRFP).

The binary SRFP of given substructural fingerprint $s_1, ..., s_d$ is defined as a matrix $\bar{G}$ such that

$$\bar{G}_{ij} = 1 \iff \exists_{a,b} a \text{ is } s_i \wedge b \text{ is } s_j \wedge a \text{ is connected to } b \text{ in } c.$$

[1] There are also non-binary alternatives, but they are beyond the scope of this research.

In other words, we have an undirected graph with labeled vertices, which are substructures of a given fingerprint. The edges are present between two vertices if for at least one pair of such substructures in a given compound they are connected. We will denote the graph G and its adjacency matrix by $\bar{G}$ through the rest of the paper. Notice that in general $\bar{G}$ are very sparse (as a consequence of fp sparseness and typical "tree-like" shapes of compounds)

Once we have a condensed graph representation of compounds we need to efficiently compute whether two graphs G_1 and G_2 are similar to each other. As stated before, there are existing methods such as graph kernels which answer the similarity queries. Unfortunately, they are highly inefficient and often relatively complex to implement. This is a consequence of the fact, that many of them try to count how many paths of arbitrary length are shared by G_1 and G_2. Instead, we consider a simplified approach where we are only interested in paths of length 1, in other words – edges. Such question can be easily answered given adjacency matrices, as number of shared edges is given by a matrix scalar product $\langle \bar{G}_1, \bar{G}_2 \rangle = \mathrm{tr}(\bar{G}_1 \bar{G}_2^T)$. However, use of such similarity measure followed by a linear model would degenerate to the linear model in the input space (as composition of two linear models is still linear). Thus one needs to introduce non-linearity which might be achieved through use of a particular normalization scheme [9] which leads to three basic set-similarity measures known in literature: Tanimoto (also known as Jaccard; $\phi_{\tan}$), Sørensen-Dice (also known as Sørensen; $\phi_{\mathrm{sør}}$) and overlap coefficient (ϕ_{ove}):

$$\phi_{\tan}(G_1, G_2) = \frac{\langle \bar{G}_1, \bar{G}_2 \rangle}{\langle \bar{G}_1, \bar{G}_1 \rangle + \langle \bar{G}_2, \bar{G}_2 \rangle - \langle \bar{G}_1, \bar{G}_2 \rangle} \in [0,1],$$

$$\phi_{\mathrm{ove}}(G_1, G_2) = \frac{\langle \bar{G}_1, \bar{G}_2 \rangle}{\min\{\langle \bar{G}_1, \bar{G}_1 \rangle, \langle \bar{G}_2, \bar{G}_2 \rangle\}} \in [0,1],$$

$$\phi_{\mathrm{sør}}(G_1, G_2) = \frac{2\langle \bar{G}_1, \bar{G}_2 \rangle}{\langle \bar{G}_1, \bar{G}_1 \rangle + \langle \bar{G}_2, \bar{G}_2 \rangle} \in [0,1],$$

where $\langle \cdot, \cdot \rangle$ is defined as above.

In order to use such measure of similarity in random projection based models we need to randomly generate one of the arguments (as φ function is unary). While one could perform inference of compounds SRFP distribution, this task is computationally expensive. Instead, we can easily approximate sampling from such distribution by sampling from uniform distribution over training set (as we are dealing with massive datasets, the training samples should fill the whole sample space quite densely). In other words we put

$$\varphi_m(G) = [\phi_m(G, F_1), \ldots, \phi_m(G, F_h)]^T \in [0,1]^h,$$

where $m \in \{\tan, \mathrm{ove}, \mathrm{sør}\}$, $F_i \sim \mathrm{uniform}(\{G_1, \ldots, G_N\})$ and h is a fixed parameter of the projection size.

One can easily observe that Tanimoto and Sørensen coefficient behave quite similarly (up to some nonlinear scaling differences). In particular, using basic algebra one can prove following observation.

Observation 1. *Two pairs of graphs share the same Tanimoto coefficient if and only if they share the same Sørensen coefficient. For any G_1, G_2, G_1', G_2' we have*

$$\phi_{\tan}(G_1, G_2) = \phi_{\tan}(G_1', G_2') \iff \phi_{\mathrm{sør}}(G_1, G_2) = \phi_{\mathrm{sør}}(G_1', G_2')$$

This relation will be further used during an evaluation and will help to understand similarities between the results of these two projections. It is worth noting that analogical result do not hold for overlap projection and any of the above methods.

Let us assume that we are given a training set in the form of SRFPs, meaning that we have a tensor $\mathbf{X}$ where $\mathbf{X}_{ijk}$ denotes whether in ith compound, jth and kth substructure is connected. Furthermore, $\mathbf{X}^+$ and $\mathbf{X}^-$ denotes positive samples (active compounds) and negative samples (inactive compounds) respectively. For given regularization parameter C, size of the hidden layer (number of preselected compounds) h and one of the above projections m we can provide extremely simple algorithms for training our model in Algorithm 1. Operation $A \otimes B$ in Algorithm 2 denotes the tensor product with summation over two indices, which is equivalent in Einstein notation to $A_{kij}B_{lij}$. Consequently, due to the SRFP symmetry $(A \otimes B)_{kl} = \langle A_k, B_l \rangle$, as desired.

Algorithm 1 Regularized Extreme Entropy Machine

TRAIN$(\mathbf{X}^+, \mathbf{X}^-)$
- **build** φ **using Algorithm 2**
- $\mathbf{H}^{\pm} \leftarrow \varphi(\mathbf{X}^{\pm})$
- $\mathbf{m}^{\pm} \leftarrow 1/|\mathbf{H}^{\pm}| \sum_{\mathbf{h}^{\pm} \in \mathbf{H}^{\pm}} \mathbf{h}^{\pm}$
- $\mathbf{\Sigma}_C^{\pm} \leftarrow \mathrm{cov}_{\dagger}(\mathbf{H}^{\pm}) + \mathbf{I}/(2C)$
- $\boldsymbol{\beta} \leftarrow 2\left(\mathbf{\Sigma}_C^+ + \mathbf{\Sigma}_C^-\right)^{-1}(\mathbf{m}^+ - \mathbf{m}^-)/\|\mathbf{m}^+ - \mathbf{m}^-\|_{\mathbf{\Sigma}_C^+ + \mathbf{\Sigma}_C^-}$
- $\mathrm{F}(x) = \arg\max_{t \in \{+,-\}} \mathcal{N}(\boldsymbol{\beta}^T \mathbf{m}^t, \boldsymbol{\beta}^T \mathbf{\Sigma}_C^t \boldsymbol{\beta})[x]$
- **return** $\boldsymbol{\beta}, \varphi, \mathrm{F}$

PREDICT$(\mathbf{X})$
- **return** $\mathrm{F}(\boldsymbol{\beta}^T \varphi(\mathbf{X}))$

Algorithm 2 φ building for SRFPs

$\phi_{\tan}(\mathbf{X}, \mathbf{W}) = (\mathbf{X} \otimes \mathbf{W})/(\mathbf{X} \otimes \mathbf{X} + \mathbf{W} \otimes \mathbf{W} - \mathbf{X} \otimes \mathbf{W})$
$\phi_{\mathrm{sør}}(\mathbf{X}, \mathbf{W}) = 2(\mathbf{X} \otimes \mathbf{W})/(\mathbf{X} \otimes \mathbf{X} + \mathbf{W} \otimes \mathbf{W})$
$\phi_{\mathrm{ove}}(\mathbf{X}, \mathbf{W}) = (\mathbf{X} \otimes \mathbf{W})/\min\{\mathbf{X} \otimes \mathbf{X}, \mathbf{W} \otimes \mathbf{W}\}$

BUILD $\varphi(\mathbf{X}, h, m)$
- $\mathbf{w}_i \sim \mathrm{uniform}(\mathbf{X})$ **for** $i \in \{1, ..., h\}$
- $\varphi(\mathbf{X}) = \phi_m(\mathbf{X}, [\mathbf{w}_1, \ldots, \mathbf{w}_h])$
- **return** φ

V. Experiments

In order to perform experiments on hard, massive, cheminformatics problems we use 15 proteins from ChEMBL database which have highest number of confirmed active/inactive compounds[2] Table I summarizes these datasets. As one can see they contain dozens of thousands of inactive

[2]We ignore proteins for which the number of actives compounds is smaller than 10.

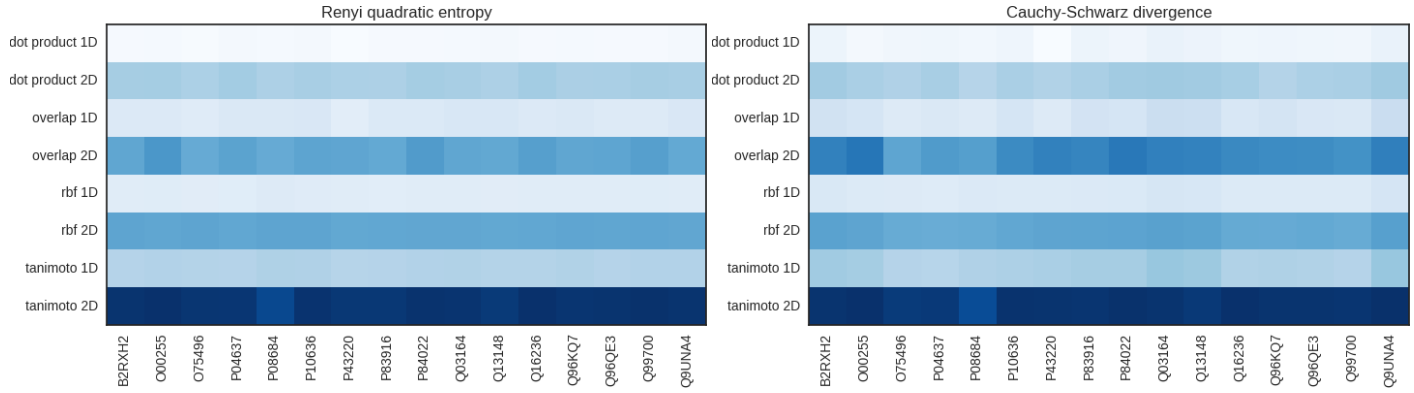

Fig. 2. Entropy of 100 of random projections through various activation functions with SRFP.

compounds and from few dozens to few thousands active compounds.

TABLE I. SUMMARY OF USED DATASETS.

dataset	protein name	$\lvert\mathbf{X}_+\rvert$	$\lvert\mathbf{X}_-\rvert$
B2RXH2	Lysine-specific demethylase 4E	74	39694
O00255	Menin	43	46700
O75496	Geminin	5315	106798
P04637	Cellular tumor antigen p53	2026	41946
P08684	Cytochrome P450 3A4	476	10061
P10636	Microtubule-associated protein tau	544	88102
P43220	Glucagon-like peptide 1 receptor	59	104223
P83916	Chromobox protein homolog 1	162	91518
P84022	Mothers against decapentaplegic homolog 3	121	59708
Q03164	Histone-lysine N-methyltransferase 2A	29	62948
Q13148	TAR DNA-binding protein 43	26	37903
Q16236	Nuclear factor erythroid 2-related factor 2	995	85915
Q96KQ7	Histone-lysine N-methyltransferase EHMT2	487	86527
Q96QE3	ATPase family AAA domain-containing protein 5	642	115129
Q99700	Ataxin-2	1714	45117
Q9UNA4	DNA polymerase iota	34	113792

The huge disproportion is not the only reason for the problem complexity. Samples are also not iid [12] as distribution of samples is a result of chemists new drugs candidates search strategies, which are highly non-random.

Let us focus first on some general behavior of considered random projections. We can take the distribution of values returned by each φ_i and compute two characteristics

1) mean Renyi's quadratic entropy of $P(\varphi_i(\mathbf{X}))$ to measure how diverse are data representations in the feature space, the higher the entropy, the more completely and uniformly is feature space filled with data points;
2) mean Cauchy-Schwarz divergence of $P(\varphi_i(\mathbf{X}^{\pm}))$, measuring how divergent are representations of each class in marginal sense (we compute just a single projection and take a mean divergence, so we do not consider relations between multiple projections).

Figure 2 visualizes these two characteristics over all considered projections and all tested proteins. We do not include results of Sørensen projection, as due to Observation 1 they are identical to the one obtained by Tanimoto.

One can notice two important aspects. First, 2-dimensional fingerprint yields consistently higher entropy and divergence over its 1-dimensional counterpart. This means (to some extent, as the experiment considers simplified model) that such fingerprint leads to more uniformly filled feature space and furthermore that it is more discriminative in terms of used labels. Second, similar observation can be made for activation functions, which are consistently sorted in terms of entropy and divergence from dot product based (linear, sigmoid) through rbf, overlap and finally Tanimoto, which yield the best scores.

Let us now proceed to main evaluation part. We consider ten machine learning models, namely: SVM with rbf kernel (SVM_{rbf}), Weighted Extreme Learning Machine [13] with Tanimoto (WELM_{tan}), Sørensen ($\text{WELM}_{\text{sør}}$) and overlap (WELM_{ove}) activation functions, Extreme Entropy Machine with Tanimoto (EEM_{tan}), Sørensen ($\text{EEM}_{\text{sør}}$) and overlap (EEM_{ove}) activation functions, Random Forest [14] (RF), one-class SVM with rbf (oSVM_{rbf}), Tanimoto (oSVM_{tan}) and Sørensen kernel ($\text{oSVM}_{\text{sør}}$). We do not use SVM with neither Tanimoto nor Sørensen kernel as it failed to converge in reasonable time using on-demand kernel computation and we were unable to precompute kernel matrices (as they were of size $100,000^2 = 10,000,000,000$). We could not use overlap as a kernel for SVM as it does not meet the Mercer condition. Each method has fitted hyperparameters: C regularization for SVM ($C \in [10^{-}1, ..., 10^{6}]$) with RBF, EEM with all activation functions and WELM with all activation functions ($C \in [10^{2}, ..., 10^{6}]$); $\gamma \in [10^{-}6, ..., 10^{1}]$ for each method using rbf kernel; number of trees for Random Forest ($[10, 100, ..., 1000]$); $\mu \in \{0.1, 0.2, ..., 0.9\}$ for each one-class SVM. Experiments were performed in 5-fold cross validation with Balanced Accuracy used as evaluation metric

$$\text{BAC}(\text{TP}, \text{FP}, \text{TN}, \text{FN}) = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP}+\text{FN}} + \frac{\text{TN}}{\text{TN}+\text{FP}}\right).$$

The code is written in python with use of numpy [15], scipy and scikit-learn [16]. For SVM we use scikit-learn binding to LIBSVM (for both two-class SVM and one-class SVM), similarly for RF. WELM and EEM are implemented from scratch using basic algebraic operations. Each method was evaluated using SubFP and SRFP (based on SubFP), resulting in total 20 models tested on 16 datasets.

There are three general observations. First, introduction of SRFP leads to increase in model quality over the one-

TABLE II. SUMMARY OF BAC SCORES FOR EACH METHOD. SCORE ON THE LEFT INDICATES THE ONE OBTAINED FOR SUBFP REPRESENTATION AND ON THE RIGHT WITH SRFP. BOLDED VALUES INDICATE BEST RESULT FOR A GIVEN PROTEIN.

	SVM_{rbf}	$oSVM_{rbf}$	$oSVM_{tan}$	$oSVM_{sør}$	EEM_{tan}	ELM_{tan}	EEM_{ove}	ELM_{ove}	$EEM_{sør}$	$ELM_{sør}$	RF
B2RXH2	50.0 / 53.4	52.7 / 57.3	54.1 / 52.9	54.1 / 53.4	62.2 / 62.9	60.9 / 60.3	59.3 / 62.3	57.1 / 59.1	61.2 / **64.4**	59.0 / 58.1	51.4 / 50.0
O00255	50.0 / 89.4	61.1 / 66.0	76.6 / 74.4	75.9 / 73.7	87.9 / **90.1**	87.3 / 89.9	84.7 / 88.6	86.2 / 88.5	86.2 / 88.9	86.7 / 88.5	79.1 / 79.0
O75496	50.0 / 50.0	51.5 / 51.0	51.0 / 51.4	51.0 / 51.3	59.3 / 61.4	59.1 / 60.8	58.5 / 60.9	58.7 / 60.1	58.7 / **61.5**	58.6 / 61.3	51.2 / 51.4
P04637	50.0 / 50.0	52.2 / 50.8	55.5 / 57.2	55.5 / 56.8	64.0 / **69.5**	63.9 / 69.3	63.4 / 66.0	63.0 / 65.0	63.6 / 66.0	63.6 / 66.0	52.7 / 53.7
P08684	50.0 / 59.9	51.9 / 53.6	56.5 / 64.8	56.2 / 63.1	72.2 / 86.3	70.1 / **86.6**	72.7 / 71.0	71.7 / 70.4	72.7 / 69.5	72.4 / 70.2	61.6 / 65.2
P10636	50.0 / 50.0	52.1 / 51.7	51.7 / 50.2	52.0 / 50.1	56.7 / **60.1**	55.7 / 59.1	55.9 / 56.7	55.7 / 55.6	56.4 / 56.6	56.0 / 56.5	50.5 / 50.5
P43220	50.0 / 94.9	66.3 / 78.9	76.4 / 77.9	79.6 / 82.7	92.5 / 98.2	92.8 / **99.1**	91.4 / 92.2	91.5 / 92.2	92.5 / 92.1	92.4 / 93.9	70.6 / 71.6
P83916	50.0 / 50.0	51.7 / 57.8	47.2 / 49.1	47.2 / 48.4	60.8 / **63.0**	59.6 / 61.3	58.6 / 60.9	56.3 / 59.9	59.6 / 61.4	59.8 / 58.7	50.3 / 50.0
P84022	50.0 / 50.6	52.5 / 51.0	54.1 / 53.7	54.3 / 54.1	60.4 / **61.2**	56.1 / 60.8	56.5 / 59.3	56.1 / 57.7	59.7 / 58.3	58.6 / 57.0	50.4 / 50.0
Q03164	50.0 / 53.2	53.7 / 57.7	52.5 / 46.1	53.5 / 47.2	56.6 / 61.2	56.4 / 58.1	59.8 / 60.2	59.4 / 61.7	57.7 / **64.2**	53.1 / 62.7	50.0 / 50.0
Q13148	50.0 / 55.4	54.3 / 56.2	56.5 / 50.6	**67.3** / 57.0	**67.3** / 59.8	63.1 / 58.5	65.1 / 56.3	63.0 / 53.5	66.6 / 58.6	62.3 / 58.9	50.0 / 50.0
Q16236	50.0 / 50.0	50.0 / 51.1	49.3 / 49.7	49.6 / 49.7	58.1 / **60.5**	58.0 / 60.2	57.1 / 59.4	56.4 / 58.3	57.2 / 59.9	56.8 / 59.8	53.3 / 53.7
Q96KQ7	50.0 / 50.0	53.5 / 56.5	53.4 / 50.8	53.5 / 50.4	67.3 / **76.7**	67.0 / 76.5	67.5 / 76.4	68.0 / 72.8	65.0 / 76.1	65.3 / 74.7	51.6 / 51.6
Q96QE3	50.0 / 50.0	50.0 / 51.5	51.0 / 49.3	50.3 / 49.4	61.5 / 63.6	61.5 / 63.2	59.6 / 63.4	59.9 / 63.5	59.5 / 63.7	59.4 / **64.0**	53.8 / 56.3
Q99700	50.0 / 50.0	51.9 / 52.7	51.8 / 50.4	51.5 / 50.3	60.1 / **63.0**	59.7 / 61.5	59.1 / 62.2	58.6 / 60.8	59.2 / 62.8	59.4 / 62.0	50.6 / 50.5
Q9UNA4	50.0 / 50.0	56.0 / 63.6	55.5 / 51.6	55.5 / 50.3	68.0 / 71.9	70.3 / 68.1	73.2 / 70.5	72.8 / 67.0	69.5 / **78.7**	66.8 / 69.9	50.0 / 50.0
wins	0	0	0	1	**9**	2	0	0	4	1	0

dimensional one. This is mostly a consequence of introduction of much more information in the graph induced by SRFP representation, while at the same time there is not much redundancy which might lead to underfitting of the model. Second, Tanimoto projection leads to the highest scores over all considered similarity measures. It appears that it is well suited for the set-like representation of chemical compounds. It replicates the results from experiment summarized in Figure 2 where entropic measures showed Tanimoto (and Sørensen) advantages over remaining methods. Interestingly, Sørensen projection seems to be the "second call" in these experiments. Both Tanimoto and Sørensen share the same entropic results, but it seems that Tanimoto normalization better controls scale of the parameters helping classifiers create discriminative models. Finally, EEM achieves better scores than very similar WELM approach using exactly the same representation and projection.

It is quite surprising that one-class models (in case of rbf kernel trained once on majority and once on minority classs, with better results on minority; and for the remaining kernels – solely on minority) yielded very bad scores even though classes imbalance should favor such methods. Our hypothesis is that the classification problem considered in this paper is very complex, and positive class cannot be simply modeled using boundary-based approach used by this one-class model. It appears that in order to make valid decision (even if its only barely better than the random one) we need a relation to the second class. One possible alternative would be to use one-class model connected with some sophisticated manifold learning techniques such as deep learning [17].

VI. CONCLUSION

We showed a proof of concept solution for the classification of large dataset of extremely imbalanced compounds. The proposed concept is based on the connection of three important elements, namely substructural relations fingerprints, regularized Extreme Entropy Machines and Tanimoto random projection. We show that this particular set of elements lead to the best results as compared with alternative approaches such as simple structural fingerprints, different machine learning models, and different projections. While achieved results are promising and show significant increase in tha classification quality, the problem is still very complex and unsolved. Future research in the area is needed to address hardly modelable relations in positive classes.

ACKNOWLEDGMENT

Work of the first author was partially financed by National Science Centre Poland grant no. 2014/13/B/ST6/01792.

REFERENCES

[1] S. Smusz, R. Kurczab, and A. J. Bojarski, "The influence of the inactives subset generation on the performance of machine learning methods." *J. Cheminformatics*, vol. 5, p. 17, 2013.

[2] C. W. Yap, "Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints," *Journal of computational chemistry*, vol. 32, no. 7, pp. 1466–1474, 2011.

[3] W. M. Czarnecki and J. Tabor, "Extreme entropy machines: Robust information theoretic classification," *arXiv preprint arXiv:1501.05279*, 2015.

[4] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.

[5] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 2. IEEE, 2004, pp. 985–990.

[6] W. M. Czarnecki and J. Tabor, "Multithreshold entropy linear classifier: Theory and applications," *Expert Systems with Applications*, vol. 42, pp. 5591–5606, 2015.

[7] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *Journal of multivariate analysis*, vol. 88, no. 2, pp. 365–411, 2004.

[8] S. Smusz, R. Kurczab, and A. J. Bojarski, "A multidimensional analysis of machine learning methods performance in the classification of bioactive compounds," *Chemometrics and Intelligent Laboratory Systems*, vol. 128, pp. 89–100, 2013.

[9] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," *The Journal of Machine Learning Research*, vol. 11, pp. 1201–1242, 2010.

[10] L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi, "Graph kernels for chemical informatics," *Neural Networks*, vol. 18, no. 8, pp. 1093–1110, 2005.

[11] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: Hardness results and efficient alternatives," in *Learning Theory and Kernel Machines*. Springer, 2003, pp. 129–143.

[12] S. Jastrzebski and W. M. Czarnecki, "Analysis of compounds activity concept learned by svm using robust jaccard based low-dimensional embedding," *Schedae Informaticae*, 2015.

[13] W. Zong, G.-B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.

[14] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[15] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[17] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 448–455.