A Random Fourier Features based Streaming Algorithm for Anomaly Detection in Large Datasets

Deena P. Francis and Kumudha Raimond

Abstract Anomaly detection is an important problem in real-world applications. It is particularly challenging in the streaming data setting where it is infeasible to store the entire data in order to apply some algorithm. Many methods for identifying anomalies from data have been proposed in the past. The method of detecting anomalies based on a low-rank approximation of the input data that are non-anomalous using matrix sketching has shown to have low time, space requirements, and good empirical performance. However, this method fails to capture the non-linearities in the data. In this work, a kernel-based anomaly detection method is proposed which transforms the data to the kernel space using random Fourier features (RFF). When compared to the previous methods, the proposed approach attains significant empirical performance improvement in datasets with large number of examples.

Keywords Streaming data · Anomaly detection · Random Fourier features
Matrix sketching

1 Introduction

Large data are encountered in many real-world applications. Due to the nature of this data, storing and processing of such data as a whole become infeasible. One of the important problems in modern applications is detecting anomalies. An anomaly is a datapoint that does not conform to the same pattern as the other data points

D. P. Francis (✉) · K. Raimond
Department of Computer Sciences Technology, Karunya University,
Coimbatore, Tamil Nadu, India
e-mail: deena.francis@gmail.com

K. Raimond
e-mail: kraimond@karunya.edu

E. B. Rajsingh et al. (eds.), *Advances in Big Data and Cloud Computing*,
Advances in Intelligent Systems and Computing 645,
https://doi.org/10.1007/978-981-10-7200-0_18

in a dataset [1]. Detecting anomalies has become important in areas such as spacecraft systems [2], medicine, and finance [1]. Many approaches for anomaly detection have been proposed in the past. Subspace-based anomaly detection has been used by some works [3–5]. It involves computing a low-rank approximation of the non-anomalous input data points and then projecting the newly arrived points onto it. The anomalous points are discovered, and the non-anomalous points are used to update the low-rank approximation matrix. Huang et al. [6] used a matrix sketching technique for detecting anomalies in streaming data. They proposed a deterministic technique (DetAnom) which achieved better empirical results when compared to other scalable anomaly detection algorithms such as support vector machine (SVM) with linear as well as radial basis function (RBF) kernel, isolation forest [7], mass estimation [8], and unconstrained least-squares importance fitting [9]. They also achieved significant savings in time as well as space requirements. However, due to the non-linearities in data encountered in modern applications, a linear subspace method like [6] fails to capture the behavior of the data. A kernel function maps the data to a non-linear feature space. Since directly applying kernel functions are computationally expensive, RFF method [10] is used to approximate the kernel function. In this work, RFF method [10] is used to transform the data to a feature space, and then the anomalies are identified.

This work is organized as follows. The notations used are described in Sect. 2. The previous related works are described in Sect. 3. The proposed approach is described in Sect. 4. The experimental results and discussion are provided in Sect. 5, and the conclusion is provided in Sect. 6.

2 Preliminaries

For a data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, n is the number of examples and d is the number of attributes of $\mathbf{X}$. $\mathbb{I}_d$ is the identity matrix of size $d \times d$. The singular value decomposition (SVD) of $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{d \times d}$ is an orthogonal matrix, $\mathbf{V} \in \mathbb{R}^{n \times n}$ is an orthogonal matrix, and $\mathbf{\Sigma} \in \mathbb{R}^{d \times n} = \{\sigma_i\}$ is a diagonal matrix. The matrix $\mathbf{\Sigma}$ contains the singular values of $\mathbf{X}$, sorted in the decreasing order, i.e., $\sigma_i \geq \sigma_j$ for $i \leq j$. The data arrives in a streaming fashion. $\mathbf{X}_t$ denotes the data that arrives at time t, where $\mathbf{X}_t \in \mathbb{R}^{d \times n_t}$, d is the number of attributes of the data, and n_t is the number of instances of the data at time t. The matrix $\mathbf{X}_{[t]} \in \mathbb{R}^{d \times n_{[t]}}$ denotes the horizontal concatenation of matrices $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_t \in \mathbb{R}^{d \times n_i}$, where $i = 1, \ldots, t$. The set of non-anomalous points identified at time $t-1$ is denoted by $\mathbf{N}_{[t-1]}$. The rank-k approximation of $\mathbf{N}_{[t-1]}$ is $\mathrm{SVD}_k(\mathbf{N}_{[t-1]_k}) = \mathbf{U}_{(t-1)_k} \mathbf{\Sigma}_{(t-1)_k} \mathbf{V}^T_{(t-1)_k}$.

3 Previous Works

Density-based methods were used by [11, 12] to detect anomalies. The problem with this approach is that the all pairwise distance computation is expensive and hence cannot be used in large data. Nicolas and McDermott [13] proposed an autoencoder and density estimation method which also has the problem of expensive computation. Many subspace based anomaly detection approaches have been proposed in the past. Such methods construct a low-rank subspace of the non-anomalous data points in order to detect anomalies. Huang et al. [4, 5] used a principal component analysis (PCA)-based method using a sliding window scheme. These approaches suffer from the drawback of poor scalability. In order to overcome the problem of scalability, both deterministic and randomized matrix sketching-based techniques were proposed by [6]. The deterministic method (DetAnom) is based on the frequent directions (FD) algorithm of [14]. In their method, the rank-k approximation of the non-anomalous points observed at time $t-1$, $\mathbf{N}_{(t-1)}$ is computed. Using its left singular vectors $\mathbf{U}_{(t-1)_k}$, the anomaly score of a new data point $\mathbf{x}_i$ is constructed as follows.

$$a_i = \|(\mathbb{I}_d - \mathbf{U}_{(t-1)_k}\mathbf{U}_{(t-1)_k}^T)\mathbf{x}_i\| \tag{1}$$

The points that have anomaly score greater than a threshold are marked as anomalies, and the rest are marked as non-anomalies. The left singular vectors are updated with the newly discovered non-anomalous points using a modified FD algorithm. This algorithm like most of the previous works does not capture the non-linearities in the data. Kernel-based data transformation can be used to overcome the drawback of the previous methods.

4 Proposed Approach

The proposed algorithm first uses RFF method [10], and then applies the FD-based anomaly detection algorithm, **DetAnom** of [6]. The proposed algorithm, RFFAnom, is shown in Algorithm 1. $\mathbf{X}_{(t-1)}$ is the set of data points at time $(t-1)$, and $\mathbf{X}_t$ is the set of points at time t (new points). The algorithm starts with an initial set of non-anomalous points $\mathbf{N}_{t-1}$ using which an initial sketch matrix $\mathbf{B}_{t-1}$ and the matrix $\mathbf{U}_{(t-1)_k}$ are computed. The columns of $\mathbf{X}_{t-1}$ are made to have unit l-2 norm, obtained by normalizing $\mathbf{X}_{t-1}$. As in [6], it is assumed that at any time t, a set of new points $\mathbf{X}_t$ arrives. This batch (set of points) $\mathbf{X}_t$ is transformed to the kernel space using the *FeatureMap* function in the Algorithm 2. Here, m is the number of feature maps to be generated. In this work, m is set to be equal to d, as the aim was not to perform dimensionality reduction, but rather to obtain a better representation of the non-anomalous points. The transformed points, $\mathbf{Y}_t \in \mathbb{R}^{m \times n}$, are obtained as a consequence of applying the *FeatureMap* function. These points are also normalized in order to make its columns to have unit l-2 norm. The anomaly scores a_i are calculated as the distance between the points $\mathbf{y}_i$ in $\mathbf{Y}_t$ and the projection of the points $\mathbf{y}_i$ onto

Algorithm 1 RFFAnom

Input: $\mathbf{X}_t \in \mathbb{R}^{d\times n_t}, \mathbf{U}_{(t-1)_k} \in \mathbb{R}^{d\times k}, \eta \in \mathbb{R}, \mathbf{B}_{t-1} \in \mathbb{d}^{m\times l}, \mathbf{N}_t \leftarrow [], \mathbf{A}_t \leftarrow [], \varsigma \in \mathbb{R}$
Initial $\mathbf{N}_{t-1}$ is used to compute $\mathbf{B}_{t-1}$
for each new set of points $\mathbf{X}_t$ **do**
  $\mathbf{Y}_t = \mathit{FeatureMap}(\mathbf{X}_t, \varsigma)$
  for each point $\mathbf{y}_i$ in $\mathbf{Y}_t$ **do**
    $a_i = \|(\mathbb{I}_d - \mathbf{U}_{(t-1)_k}\mathbf{U}^T_{(t-1)_k})\mathbf{y}_i\|$
    if $a_i \leq \eta$ **then**
      $\mathbf{N}_t \leftarrow [\mathbf{N}_t, \mathbf{y}_i]$
    end if
  end for
  $\mathbf{N}_{[t]} \leftarrow [\mathbf{N}_{[t-1]}, \mathbf{N}_t]$
  $\mathbf{B}_t \leftarrow [\mathbf{B}_{t-1}, \mathbf{N}_{[t]}]$
  $\tilde{\mathbf{U}}_{t_l}\tilde{\mathbf{\Sigma}}_{t_l}\tilde{\mathbf{V}}^T_{t_l} \leftarrow \mathrm{SVD}_l(\mathbf{B}_t)$
  $\mathbf{B}_t \leftarrow \tilde{\mathbf{U}}_{t_l}\mathrm{diag}(\sqrt{\tilde{\sigma}^2_{t_l} - \tilde{\sigma}^2_{t_l}}, \ldots, \sqrt{\tilde{\sigma}^2_{t_{l-1}} - \tilde{\sigma}_{t_l{}^2}}, 0)$
  $\tilde{\mathbf{U}}_{t_k} \leftarrow [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$
end for
return $\mathbf{B}_t$ and $\tilde{\mathbf{U}}_{t_k}$

the rank-k subspace $\mathbf{U}_{(t-1)_k}$ of the non-anomalous points $\mathbf{N}_{t-1}$. If a_i is smaller than a threshold η, then the corresponding point is appended to the set of non-anomalous points $\mathbf{N}_t$. After all the non-anomalous points are obtained, the left singular vectors $\mathbf{U}_{(t)}$ are updated by using the FD algorithm. In this part of the algorithm, the sketch matrix $\mathbf{B}_t \in \mathbb{R}^{m\times l}$ is updated with the new set of non-anomalous points $\mathbf{N}_t$. Here, l is set as $\sqrt{m}$ as suggested by [6]. Finally, the new set of left singular vectors $\tilde{\mathbf{U}}_{t_k}$ is obtained. A diagram describing the proposed method is shown in Fig. 1. The run-

Algorithm 2 *Feature Map*$(\mathbf{X}_t, \varsigma)$

$\mathbf{R} \leftarrow$ generate Gaussian random matrix with standard deviation ς, $\mathbf{R} \in \mathbb{R}^{m\times d}$
$\gamma \leftarrow$ Sampled uniformly at random from $[0, 2\pi]$, $\gamma \in \mathbb{R}^{m\times 1}$
$\mathbf{Y}_t \leftarrow \sqrt{\frac{2}{m}}\cos(\mathbf{R}\mathbf{X}_t + \gamma)$
return $\mathbf{Y}_t$

ning time of *DetAnom* algorithm is $O(\max\{dn_t l, dl^2\})$, and the proposed algorithm is slower by a factor of $\sqrt{d}$. This does not affect the running time in the experiments to a great extend because the datasets considered do not have high dimensionality. By using RFF, the running time of applying kernel functions is reduced significantly. The space required by the algorithm is $O(d.\max_t\{n_t\} + dl)$.

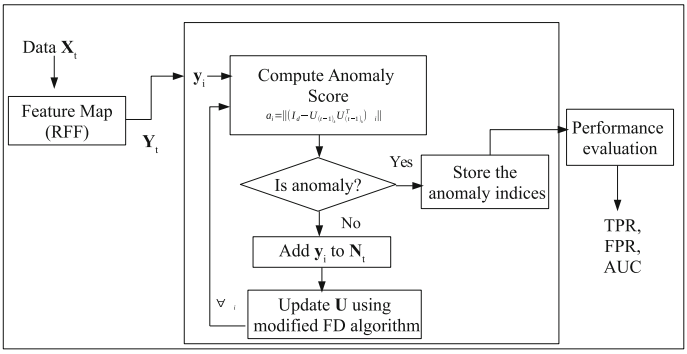

Fig. 1 Proposed method for anomaly detection from a set of new input points

5 Experimental Results

All experiments were carried out in a Linux machine with 3.5 GHz Intel Core i7 processor and 16 GB of RAM. For all the experiments, the proposed algorithm **RFFAnom** is compared against deterministic algorithm **DetAnom** of [6]. The *DetAnom* algorithm has been shown to have better empirical performance than many other scalable anomaly detection algorithms [6]. Here, non-anomalous points are labeled as 0 and the anomalous points are labeled as 1. From the set of non-anomalous data points, 2000 points are drawn at random and they comprise the initial set of non-anomalous points. The size of the data arriving as input to the algorithm at each time t is set as 5000 as suggested by [6].

5.1 Datasets Used

- COD-RNA [15]: contains 488,565 genome sequences with eight attributes. The anomalies in this case are the set of non-coding RNAs. The number of examples in classes 0 and 1 are 325710 and 162855, respectively, and the percentage of anomalies is 33%.
- Forest [16]: contains 286048 instances of forest cover types. The data were obtained from http://odds.cs.stonybrook.edu/forestcovercovertype-dataset/ and contained 10 attributes. The number of examples in classes 0 and 1 are 283301 and 2747, respectively, so the dataset has 0.9% anomalies.
- Protein-Homology [17]: contains 145751 instances and 74 attributes. The number of class 0 and 1 instances are 144455 and 1296 respectively. It has 0.8% anomalies.
- Shuttle: contains nine attributes and 49095 instances out of which 3511 are outliers. The number of non-anomalous points is 45586, so the percentage of anomalies is 7%. The data were obtained from http://odds.cs.stonybrook.edu/shuttle-dataset/.

- MNIST [18]: contains a total of 7603 instances and 100 attributes. The number of class 0 and 1 instances is 6903 and 700, respectively. It has 9% anomalies. The data were obtained from http://odds.cs.stonybrook.edu/mnist-dataset/.
- HTTP: contains 41 attributes and 200000 instances. The number of class 0 and 1 instances is 160555 and 39445, respectively, and it has 19% anomalies. The data were obtained from UCI repository [19].

5.2 Performance Metrics

The metrics used to evaluate the result of the algorithm are described below. *True Positive Rate (TPR)*: It is the proportion of correctly identified instances. Here, it is the proportion of anomalies that have been correctly identified.

$$\text{TPR} = \frac{TP}{(TP + FN)} \tag{2}$$

False Positive Rate (FPR): It is the proportion of negative instances that have been correctly identified. Here, it is the proportion of non-anomalous points that have been correctly identified.

$$\text{FPR} = \frac{FP}{FP + TN} \tag{3}$$

where TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives, and FN is the number of false negatives.
Area Under the Curve (AUC): It is a metric computed from the plot of TPR and FPR. If this value is close to 1, then the performance of the algorithm is good, and if it is less than 0.5, the performance is poor.

5.3 Results

The receiver operating characteristic (ROC) plots of the algorithms *DetAnom* and *RFFAnom* are shown in the Figs. 2, 3 and 4. For the cod-RNA and Forest datasets, the proposed algorithm, *RFFAnom*, performs much better than *DetAnom*. In particular, *DetAnom* performs suboptimally for small values of FPR, whereas *RFFAnom* has better results. The AUC values and the time taken for each dataset are shown in Table 1.

In Fig. 3a, for the Protein-Homology dataset, *DetAnom* performs slightly better than *RFFAnom*. It can be seen from Fig. 4a that for the MNIST dataset, the proposed algorithm performs better than *DetAnom*. In Fig. 4b, for the HTTP dataset, the AUC value of the proposed algorithm is 0.995, which is significantly better than that of *DetAnom*. The figure also shows how well the proposed algorithm performs since

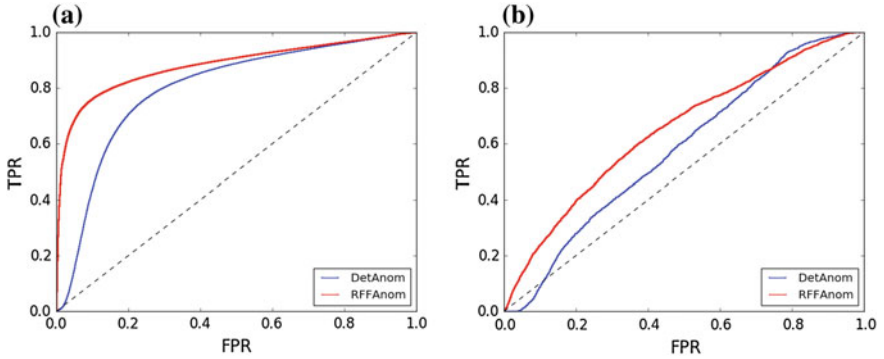

Fig. 2 ROC curves of RFFAnom and DetAnom algorithms for **a** cod-RNA (left), **b** Forest (right)

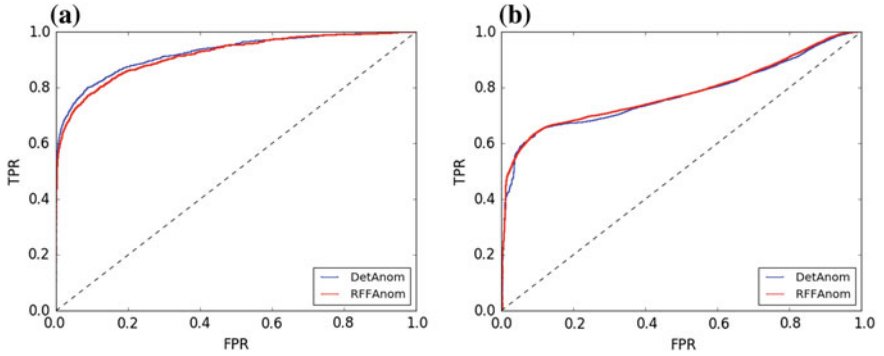

Fig. 3 ROC curves of RFFAnom and DetAnom algorithms for **a** Protein-Homology (left), **b** Shuttle (right)

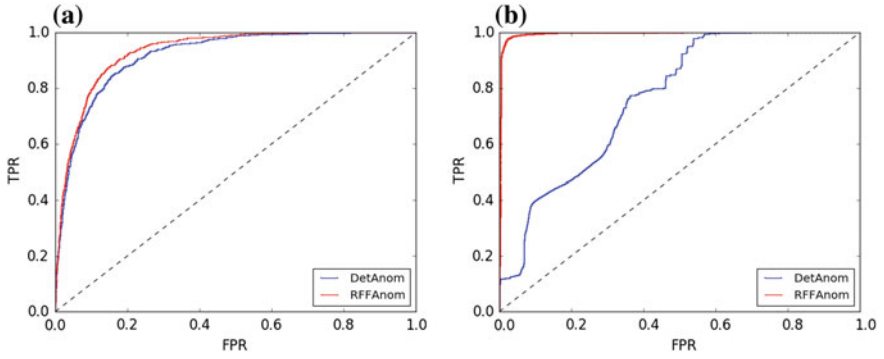

Fig. 4 ROC curves of RFFAnom and DetAnom algorithms for **a** MNIST (left), **b** HTTP (right)

Table 1 AUC values and time taken by DetAnom and RFFAnom algorithms for various datasets

Dataset	Algorithm	AUC	Time taken (s)
COD-RNA	DetAnom	0.797620	3.3825
	RFFAnom	**0.883272**	3.3717
Forest	DetAnom	0.581079	2.0930
	RFFAnom	**0.651525**	2.2101
Protein-Homology	DetAnom	**0.924820**	5.6235
	RFFAnom	0.917973	6.1562
Shuttle	DetAnom	0.773587	0.3910
	RFFAnom	**0.781500**	0.4075
MNIST	DetAnom	0.917452	0.6308
	RFFAnom	**0.933240**	0.7404
HTTP	DetAnom	0.764248	4.0326
	RFFAnom	**0.995234**	4.4200

its graph lies close to the y-axis. In general, the proposed approach performs much better than *DetAnom* for datasets with large number of instances. The results indicate that the feature space transformation improves the anomaly detection capability of the proposed algorithm. Many datasets that are available today have some kind of non-linearity present. The kernel feature space transformation (RFF) used in this work effectively exploits this nature of the data.

6 Conclusion

Detecting anomalies from streaming data is an important application in many areas. In the past, many methods for identifying anomalies from data have been proposed. But most of these algorithms suffer from the problem of poor scalability. In this work, a RFF-based anomaly detection method is proposed. It makes use of a kernel feature space transformation of the data points and a FD-based anomaly detection scheme. The proposed method has a low running time and space requirements and is hence applicable to large datasets. Empirical results indicate that a significant improvement in the performance was obtained for large datasets when compared to the previous method.

Acknowledgements The authors would like to thank the financial support offered by the Ministry of Electronics and Information Technology (MeitY), Govt. of India under the Visvesvaraya Ph.D Scheme for Electronics and Information Technology.

References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**(3) (2009). https://doi.org/10.1145/1541880.1541882
2. Fujimaki, R., Yairi, T., Machida, K.: An approach to spacecraft anomaly detection problem using kernel feature space. In Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 401–410. ACM (2005). https://doi.org/10.1145/1081870.1081917
3. Lakhina, A., Crovella, M., Diot, C.: Characterization of network-wide anomalies in traffic flows. In: SIGCOMM (2004). https://doi.org/10.1145/1028788.1028813
4. Huang, L., Nguyen, X., Garofalakis, M., Jordan, M.I., Joseph, A., Taft, N.: In-network PCA and anomaly detection. In: NIPS, pp. 617–624 (2006)
5. Huang, L., Nguyen, X., Garofalakis, M., Hellerstein, J.M., Jordan, M.I., Joseph, A.D., Taft, N.: Communication-efficient online detection of network-wide anomalies. In: INFOCOM (2007). https://doi.org/10.1109/INFCOM.2007.24
6. Huang, H., Kasiviswanathan, S.P.: Streaming anomaly detection using randomized matrix sketching. Proc. VLDB Endow. **9**(3), 192–203 (2015)
7. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: IEEE ICDM, pp. 413–422 (2008). https://doi.org/10.1109/ICDM.2008.17
8. Ting, K.M., Zhou, G.T., Liu, F.T., Tan, J.S.: Mass estimation and its applications. In: ACM SIGKDD (2010). https://doi.org/10.1145/1835804.1835929
9. Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., Kanamori, T.: Statistical outlier detection using direct density ratio estimation. KAIS **26**(2) (2011)
10. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: Advances in Neural Information Processing Systems, pp. 1177–1184 (2007)
11. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM Sigmod Record, vol. 29, pp. 93–104. ACM (2000). https://doi.org/10.1145/342009.335388
12. Tang, J., Chen, Z., Fu, A.W.C., Cheung, D.W.: Enhancing effectiveness of outlier detections for low density patterns. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 535–548. Springer, Berlin, Heidelberg (2002). https://doi.org/10.1007/3-540-47887-6_53
13. Nicolau, M., McDermott, J.: A hybrid autoencoder and density estimation model for anomaly detection. In: International Conference on Parallel Problem Solving from Nature, pp. 717–726. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-45823-6_67
14. Liberty, E.: Simple and deterministic matrix sketching. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 581–588. ACM (2013). https://doi.org/10.1145/2487575.2487623
15. Uzilov, A.V., Keegan, J.M., Mathews, D.H.: Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change. BMC bioinform. **7**(1) (2006)
16. Blackard, J.A., Dean, D.J.: Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. Comput. electron. agric. **24**(3), 131–151 (1999)
17. Caruana, R., Joachims, T., Backstrom, L.: KDD-Cup 2004: results and analysis. ACM SIGKDD Explor. Newslett. **6**(2), 95–108 (2004)
18. Lecun, Y., Cortes, C.: The MNIST database of handwritten digits. (2009). http://yann.lecun.com/exdb/mnist/
19. UCI repository. https://archive.ics.uci.edu/ml/machine-learning-databases/kddcup99-mld/ (1999)