Online One-class Classification for Intrusion Detection Based on the Mahalanobis Distance

Patric Nader, Paul Honeine and Pierre Beauseroy [*†]

Institut Charles Delaunay (CNRS), Université de technologie de Troyes
10000 Troyes, France

Abstract. Machine learning techniques have been very popular in the past decade for their ability to detect hidden patterns in large volumes of data. Researchers have been developing online intrusion detection algorithms based on these techniques. In this paper, we propose an online one-class classification approach based on the Mahalanobis distance which takes into account the covariance in each feature direction and the different scaling of the coordinate axes. We define the one-class problem by two concentric hyperspheres enclosing the support vectors of the description. We update the classifier at each time step. The tests are conducted on real data.

1 Introduction

Machine learning techniques provide a powerful tool for estimating nonlinear relations from data [1]. In many disciplines including industrial systems, the majority of the available data refer to a unique class, namely the normal behavior of the system, while the data related to the abnormal/malfunctioning modes are difficult to obtain. This is where comes the role of one-class classification algorithms, which define a decision boundary around the available data that accepts as many positive samples (target class) and rejects the outliers [2]. These algorithms have been successfully applied recently in offline intrusion detection applications [3][4]. Researchers are facing many challenges to elaborate relevant online intrusion detection algorithms, namely in minimizing the time consumption of the algorithm, in reducing the complexity of updating the classifier, in improving the detection accuracy and in minimizing the false alarm rates.

Several incremental and decremental SVM algorithms were proposed for online learning [5][6], where the classifier is updated by adding/removing samples based on the new incoming observations. These multiclass approaches can not be extended for one-class classification problems. Desobry *et al.* proposed in [7] to train the classifier twice at each iteration, and the detection is performed by comparing the present sample set with the immediate past set. The repeated batch training leads to high computational costs. Zhang *et al.* used in [8] a linear optimization of the quarter-sphere SVM to reduce the computational cost of [7]. This approach is faster than [7], but the repeated training results in a delay in the processing of new samples. Another attempt to overcome the quadratic

[*]This work is supported by the "Agence Nationale de la Recherche"(ANR), grant SCALA.
[†]The authors would like to thank Thomas Morris and the Mississippi State University SCADA Laboratory for providing the real datasets.

programming problem is detailed in [9], where an iterative update of the coefficients is used at each time step. This online approach remains greedy in terms of computational cost. Gomez *et al.* introduced in [10] an adaptive online one-class SVM that stores the new samples for many iterations before incorporating them into the training set, which is time consuming. A fast online one-class approach was proposed in [11], where the coherence criterion is used to select the support vectors among the training set and to update the classifier.

In this paper, we extend the offline Mahalanobis-based approach proposed in [3]. The Mahalanobis distance takes into account the covariance in each feature direction and the different scaling of the coordinate axes [12]. We modify the decision function of the classifier to become suitable for online applications by defining two concentric hyperspheres enclosing the support vectors of the description. The main advantages of using two hyperspheres instead of one are the isolation of the outliers and the reduced number of support vectors, which makes the proposed algorithm robust to outliers and reduces its the computational costs. The remainder of this paper is organized as follows. Section 2 describes the proposed online approach. Section 3 discusses the results on the real datasets, and Section 4 provides conclusion and future works.

2 The proposed approach

Given a training dataset $\boldsymbol{x}_i$, for $i = 1, 2, \ldots, n$, in a d-dimensional input space $\mathcal{X}$. Let $\boldsymbol{K}$ be the $n \times n$ kernel matrix with entries $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ for $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathcal{X}$, where $\phi(\boldsymbol{x})$ is the mapping function to the Reproducing Kernel Hilbert Space (RKHS) of some given reproducing kernel $k(\cdot, \cdot)$. The mean of the mapped samples in the feature space, namely $\mathrm{E}[\phi(\boldsymbol{x})]$, can be estimated with the empirical center in that space, namely $c_n = \frac{1}{n} \sum_{i=1}^{n} \phi(\boldsymbol{x}_i)$.

The proposed approach is divided into two phases: an offline training phase and an online detecting/updating phase. In the offline phase, we learn the normal functioning modes of the studied system. First, we compute the quadratic Mahalanobis distance in the feature space between each sample $\boldsymbol{x}_j$ and $\boldsymbol{c}_n$:

$$\|\phi(\boldsymbol{x}_j) - \boldsymbol{c}_n\|^2_{\boldsymbol{\Sigma}} = (\phi(\boldsymbol{x}_j) - \boldsymbol{c}_n)^T \boldsymbol{\Sigma}^{-1} (\phi(\boldsymbol{x}_j) - \boldsymbol{c}_n), \tag{1}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the samples in the feature space given by: $\boldsymbol{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (\phi(\boldsymbol{x}_i) - \boldsymbol{c}_n)(\phi(\boldsymbol{x}_i) - \boldsymbol{c}_n)^T$. The Mahalanobis distance in equation (1) is computed in the RKHS as detailed in [3]: $\sum_{k=1}^{n} \lambda_k^{-1} (\sum_{i=1}^{n} \alpha_i^k \widetilde{k}(\boldsymbol{x}_i, \boldsymbol{x}))^2$, where $n\lambda_k$ and $\boldsymbol{\alpha}^k$ are the eigenvalues and eigenvectors of the centered version of $\boldsymbol{K}$, with entries[1] $\widetilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Secondly, since we are dealing with large volumes of data in the online mode and in order to minimize the computational complexity, we approximate $\boldsymbol{c}_n$ with a sparse center $\boldsymbol{c}_{\mathcal{I}}$. The center $\boldsymbol{c}_{\mathcal{I}}$ depends only on the furthest samples to $\boldsymbol{c}_n$, known as the support vectors, and only these samples are included in the computation of the Mahalanobis distance. The set of support

[1] The kernel function $\widetilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \widetilde{k}_{ij}$ is the centered version of $k_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and it is computed as follows: $\widetilde{k}_{ij} = k_{ij} - \frac{1}{n} \sum_{r=1}^{n} k_{ir} - \frac{1}{n} \sum_{r=1}^{n} k_{rj} + \frac{1}{n^2} \sum_{r,s=1}^{n} k_{rs}$.

vectors $\mathcal{I}$ is given by: $\mathcal{I} = \{i, \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}_n\|_{\boldsymbol{\Sigma}} > R_{sparse}\}$, where R_{sparse} is the threshold based on the number of these samples. The sparse center is a linear combination of the support vectors, namely $\boldsymbol{c}_{\mathcal{I}} = \sum_{i\in\mathcal{I}} \beta_i \phi(\boldsymbol{x}_i)$. The coefficients β_i are computed by minimizing the error of approximating $\boldsymbol{c}_n$ with $\boldsymbol{c}_{\mathcal{I}}$, namely $\boldsymbol{\beta} = \boldsymbol{K}_{\mathcal{I}}^{-1}\boldsymbol{k}$, where the entries of the kernel matrix $\boldsymbol{K}_{\mathcal{I}}$ are $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $i, j \in \mathcal{I}$, and $\boldsymbol{k}$ is the column vector with entries $\frac{1}{n}\sum_{k\in\mathcal{I}} k(\boldsymbol{x}_i, \boldsymbol{x}_k)$. The quadratic Mahalanobis distance in the feature space between each sample and $c_{\mathcal{I}}$, namely $\|\phi(\boldsymbol{x}) - \boldsymbol{c}_{\mathcal{I}}\|^2_{\boldsymbol{\Sigma}}$, is computed as follows:

$$\sum_{k=1}^{m} \frac{1}{\lambda_k} \Big(\sum_{i=1}^{n} \alpha_i^k k(\boldsymbol{x}_i, \boldsymbol{x}) - \sum_{i=1}^{n}\sum_{j\in\mathcal{I}} \alpha_i^k \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{n} \frac{\alpha_i^k}{n} \sum_{j=1}^{n} k(\boldsymbol{x}_j, \boldsymbol{x}) + \sum_{i=1}^{n} \frac{\alpha_i^k}{n} \sum_{j=1}^{n}\sum_{l\in\mathcal{I}} \beta_l k(\boldsymbol{x}_j, \boldsymbol{x}_l) \Big)^2.$$

We modify the one-class problem by defining two concentric hyperspheres enclosing the support vectors of the description as illustrated in Fig 1. We fix two thresholds, $R_{detection}$ and R_{sparse}. The first threshold is fixed based on the predefined number of outliers, and R_{sparse} depends on the remaining support vectors. This new definition of the one-class problem allows to separate the outliers from the support vectors, and it has many advantages. The first one is the isolation of the outliers outside the decision boundary without including these samples in the classifier, which makes the proposed algorithm robust to outliers. The second advantage is the small number of support vectors which leads to reducing the computational costs of the algorithm. In the online phase, we have a new sample at each time step. The classifier tests each new sample by computing its Mahalanobis distance to $\boldsymbol{c}_{\mathcal{I}}$, namely $\|\phi(\boldsymbol{x}_t) - \boldsymbol{c}_{\mathcal{I}}\|_{\boldsymbol{\Sigma}}$ for any $t > n$, and we can encounter three possible cases depending on this distance:

1. **First case:** $\|\phi(\boldsymbol{x}_t) - \boldsymbol{c}_{\mathcal{I}}\|_{\boldsymbol{\Sigma}} > R_{detection}$
   In this case, the new sample is considered as an outlier and an alarm is activated. The classifier must not be updated and this sample must not be included in the learning process; this prevents it from affecting the decision function of the classifier and may lead to inaccurate results.

2. **Second case:** $R_{sparse} < \|\phi(\boldsymbol{x}_t) - \boldsymbol{c}_{\mathcal{I}}\|_{\boldsymbol{\Sigma}} \leq R_{detection}$
   In this case, the new sample $\boldsymbol{x}_t$ is considered as a support vector, and it is included into the set $\mathcal{I}$. The number of support vectors is incremented, and the new kernel matrix is updated from $\boldsymbol{K}_{\mathcal{I}}$ as follows:

   $$\begin{bmatrix} \boldsymbol{K}_{\mathcal{I}} & \boldsymbol{b} \\ \boldsymbol{b}^T & k(\boldsymbol{x}_t, \boldsymbol{x}_t) \end{bmatrix},$$

   where $\boldsymbol{b}$ is the column vector with entries $k(\boldsymbol{x}_i, \boldsymbol{x}_t)$ for all $i \in \mathcal{I}$. Also, $\boldsymbol{k}$ is updated to:

   $$\frac{1}{t} \begin{bmatrix} (t-1)\boldsymbol{k} + \boldsymbol{b} \\ k'_t \end{bmatrix},$$

   having $k'_t = \sum_{i=1}^{t} k(\boldsymbol{x}_i, \boldsymbol{x}_t)$. After updating $\boldsymbol{K}_{\mathcal{I}}$ and $\boldsymbol{k}$, the new coefficients $\boldsymbol{\beta}_t$ are given by: $\boldsymbol{\beta}_t = \boldsymbol{K}_{\mathcal{I}}^{-1}\boldsymbol{k}$, where we apply the Woodbury matrix

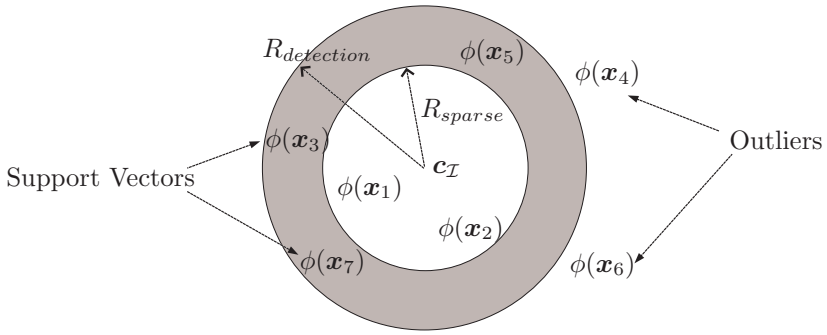

Fig. 1: A representation of the two concentric hyperspheres in the feature space.

identity to obtain the inverse of the new Gram matrix from the inverse of the old one as follows:

$$\begin{bmatrix} \boldsymbol{K}_{\mathcal{I}}^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \begin{bmatrix} -\boldsymbol{K}_{\mathcal{I}}^{-1}\boldsymbol{b} \\ \mathbf{I} \end{bmatrix} (\mathbf{I} - \boldsymbol{b}^T \boldsymbol{K}_{\mathcal{I}}^{-1}\boldsymbol{b})^{-1} \begin{bmatrix} -\boldsymbol{b}^T \boldsymbol{K}_{\mathcal{I}}^{-1} & \mathbf{I} \end{bmatrix},$$

having $\mathbf{0}$ a column vector of zeros, and $\mathbf{I}$ the identity matrix.

3. **Third case:** $\|\phi(\boldsymbol{x}_t) - \boldsymbol{c}_{\mathcal{I}}\|_{\boldsymbol{\Sigma}} \leq R_{sparse}$
In this case, $\phi(\boldsymbol{x}_t)$ is not a support vector, and it is not included into the set $\mathcal{I}$. The number of support vectors remains unchanged, as well as the $\boldsymbol{K}_{\mathcal{I}}$, and $\boldsymbol{k}$ is updated to $\frac{1}{t}((t-1)\boldsymbol{k} + \boldsymbol{b})$, where $\boldsymbol{b}$ is the column vector with entries $k(\boldsymbol{x}_i, \boldsymbol{x}_t)$ for all $i \in \mathcal{I}$. The new coefficients are given by:

$$\frac{t-1}{t}\boldsymbol{\beta} + \frac{1}{t}\boldsymbol{K}_{\mathcal{I}}^{-1}\boldsymbol{b}.$$

3 Experimental results

We tested the proposed online algorithm on two real datasets from the Mississipi State University SCADA Laboratory, the gas pipeline and the water storage tank testbeds [13]. The gas pipeline is used to move petroleum products to market, and the water storage tank is similar to the oil storage tanks found in the petrochemical industry. We have a new input sample every two seconds, and it consists of 27 attributes for the gas pipeline and 24 attributes for the water storage tank, i.e., gas pressure, water level, pump state, target gas pressure/water level, PID's parameters, time interval, length of the packets, and command functions. In addition, 28 types of attacks are injected into the network traffic of the system in order to hide its real functioning state and to disrupt the communication. These attacks are arranged into 7 groups: Naive Malicious Response Injection (NMRI), Complex Malicious Response Injection (CMRI), Malicious State Command Injection (MSCI), Malicious Parameter Command Injection (MPCI), Malicious Function Command Injection (MFCI), Denial of Service (DOS) and

Reconnaissance Attacks (RA). The Gaussian kernel used in this paper has the following expression $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2\sigma^2})$, where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two input samples, and $\|\cdot\|_2$ represents the l_2-norm in the input space. The bandwidth parameter σ is computed as proposed in [4], namely $\sigma = \frac{d_{\max}}{\sqrt{2M}}$, where $d_{\max}$ refers to the maximal distance between any two samples in the input space, and M represents the upper bound on the number of outliers among the training dataset.

The proposed online approach is compared with two other approaches, the online quarter-sphere SVM [8] and the online coherence-based one-class [11]. The first important criterion for online intrusion detection algorithms is the time for testing new samples. The proposed approach needs 0.0019 second for each new sample, which is faster than the quarter-sphere SVM with 0.0027 second and the coherence approach with 0.0022 second. On the other hand, the quarter-sphere SVM spends less time (0.16 second) to update the classifier than the proposed approach (0.23 second) and the coherence approach (0.21 second). Another important criterion for online intrusion detection algorithms is the error detection accuracy. We tested these algorithms on nearly 100 000 samples related to the aforementioned attacks, and the detection rates are given in Table 1. The results show that the proposed online approach gives better detection rates and outperforms the other approaches for all the studied attacks. In some cases, we have important gaps between the detection rates of the approaches, which can be explained by the strong properties of the Mahalanobis distance, and the advantages of the modified one-class formulation. Finally, the quarter-sphere SVM has a false alarm rate equal to 11% in average, the coherence-based has 4%, while the proposed online approach misclassified only 1% of the normal samples. These results are very interesting for online intrusion detection in real-world applications, where the proposed approach needs less than 0.002 second to detect the intrusion, it has high detection rates and low false alarm rates.

4 Conclusion

In this paper, we proposed an online one-class classification approach based on the Mahalanobis distance in the feature space. In this approach, we defined the one-class problem by two concentric hyperspheres enclosing the support vectors, and we updated the classifier after each iteration. The properties of the Mahalanobis distance and the modified one-class formulation made our algorithm robust to outliers. We tested our algorithm on real datasets containing several types of attacks, and we compared the results with other approaches. The results showed that the proposed approach is the fastest in detecting the outliers, and it has the highest detection rates and the lowest false alarm rates. For future works, the implementation of this approach should be optimized to decrease the time consumption of the update step. This approach can be extended to include other sparsification rules, and multiclass classification can be investigated to identify the detected attack's type.

Table 1: Detection rates on the real datasets.

	Gas pipeline			Water storage		
	quarter SVM	online coherence	**proposed approach**	quarter SVM	online coherence	**proposed approach**
NMRI	92.04	86.17	**99.28**	92.71	87.91	**98.44**
CMRI	98.45	92.41	**99.83**	70.12	74.31	**80.81**
MSCI	71.13	63.53	**81.35**	96.23	86.72	**98.36**
MPCI	98.05	92.38	**99.11**	99.11	90.32	**99.64**
MFCI	76.27	68.61	**83.34**	98.26	85.62	**99.83**
DOS	81.23	84.77	**95.56**	71.76	73.67	**82.11**
RA	**99.80**	91.76	**99.80**	94.17	88.37	**99.71**

References

[1] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

[2] S. S. Khan and M. G. Madden. A survey of recent trends in one class classification. In *Proceedings of the 20th Irish conference on Artificial intelligence and cognitive science*, AICS'09, pages 188–197, 2010.

[3] P. Nader, P. Honeine, and P. Beauseroy. Mahalanobis-based one-class classification. In *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Reims, France, 21–24 September 2014.

[4] P. Nader, P. Honeine, and P. Beauseroy. l_p-norms in one-class classification for intrusion detection in scada systems. *Industrial Informatics, IEEE Transactions on*, 2014 (in press).

[5] Y. Li and P. M. Long. The Relaxed Online Maximum Margin Algorithm. *Mach. Learn.*, 46(1-3):361–387, January 2002.

[6] M. Karasuyama and I. Takeuchi. Multiple incremental decremental learning of support vector machines. *Neural Networks, IEEE Transactions on*, 21(7):1048–1059, July 2010.

[7] F. Desobry, M. Davy, and C. Doncarli. An online kernel change detection algorithm. *Signal Processing, IEEE Transactions on*, 53(8):2961–2974, Aug 2005.

[8] Y. Zhang, N. Meratnia, and P. Havinga. Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks. In *Advanced Information Networking and Applications Workshops. WAINA. International Conference on*, pages 990–995, May 2009.

[9] M. Davy, F. Desobry, A. Gretton, and C. Doncarli. An online support vector machine for abnormal events detection. *Signal Process.*, 86(8):2009–2025, August 2006.

[10] V. Gomez-Verdejo, J. Arenas-Garcia, M. Lazaro-Gredilla, and A Navia-Vazquez. Adaptive one-class support vector machine. *Signal Processing, IEEE Transactions on*, 59(6):2975–2981, June 2011.

[11] Zineb Noumir, Paul Honeine, and Cédric Richard. Online one-class machines based on the coherence criterion. In *Proc. 20th European Conference on Signal Processing*, Bucharest, Romania, 27–31 August 2012.

[12] P. C. Mahalanobis. On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, pages 49–55, April 1936.

[13] T. Morris, A. Srivastava, B. Reaves, W. Gao, K. Pavurapu, and R. Reddi. A control system testbed to validate critical infrastructure protection concepts. *International Journal of Critical Infrastructure Protection*, 4(2):88 – 103, 2011.