Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 19 (2013) 1174 – 1181

Procedia
Computer Science

The 3rd International Workshop on Sensor Networks for Intelligence Gathering and Monitoring (SNIGM)

A Hierarchical Framework Using Approximated Local Outlier Factor for Efficient Anomaly Detection

Lin Xu[a,c], Yi-Ren Yeh[b,c], Yuh-Jye Lee[b], Jing Li[a]

[a] Dept. of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China
[b] Dept. of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan
[c] Intel-NTU Connected Context Computing Center, National Taiwan University, Taipei, Taiwan

Abstract

Anomaly detection aims to identify rare events that deviate remarkably from existing data. To satisfy real-world applications, various anomaly detection technologies have been proposed. Due to the resource constraints, such as limited energy, computation ability and memory storage, most of them cannot be directly used in wireless sensor networks (WSNs). In this work, we proposed a hierarchical anomaly detection framework to overcome the challenges of anomaly detection in WSNs. We aim to detect anomalies by the accurate model and the approximated model learned at the remote server and sink nodes, respectively. Besides the framework, we also proposed an approximated local outlier factor algorithm, which can be learned at the sink nodes. The proposed algorithm is more efficient in computation and storage by comparing with the standard one. Experimental results verify the feasibility of our proposed method in terms of both accuracy and efficiency.

Keywords: Local sensitive hashing, local outlier factor, anomaly detection, hamming distance

1. Introduction

Large number of spatially distributed, small, low-cost sensor nodes with one or more sink nodes collecting readings of sensor nodes comprise a wireless sensor network (WSN). These autonomous sensors monitor physical or environmental conditions, such as heat, temperature, etc. and cooperatively send the readings to the sink nodes. The WSNs, which are widely used in personal, industry, business and military domains, provide fine-grained real-time data about the physical world. By given such abundant data, many data mining or machine learning methods have been applied to make intelligent decisions for the applications in WSNs continuously and automatically [1, 2, 3]. However, the sensors in WSN are often low cost and poor quality ones which make anomalies by various reasons, such as sensor malfunction, sensor fault, or compromised sensor. For those intelligent systems, few anomalous data might lead to a biased model, which makes improper decisions and causes unexpected costs. Thus, how to design an anomaly detection mechanism for protecting the systems in WSNs has drawn attentions from researchers [4].

doi:10.1016/j.procs.2013.06.168

The existing anomaly detection techniques can be categorized as supervised, semi-supervised and unsupervised methods based on the availability of pre-defined data [5]. Supervised and semi-supervised approaches are required labeled data as normal data or anomalies in the training phase. One should note that the pre-classified data is neither always available nor easy to be obtained in real world. In contrast, the unsupervised approaches require no labeled data for the training. The main idea of unsupervised anomaly detection methods is using a certain criteria for profiling normal behavior, such as the nearest neighbor based, spectrum based and clustering based approaches [6, 7, 8]. It is worth noting that most of them are designed for batch learning and with high computational complexity and memory consumption. All the constraints inhibit the progress on anomaly detection techniques in WSN. To satisfy the demands of anomaly detection in WSNs, we introduce a hierarchical framework to provide detection models with different computational complexities for sink nodes and the remote server. For the detection model at sink nodes, we proposed an approximated local outlier detection (LOF) [9], which is unsupervised and lightweight by comparing with the exact ones. In our approximated LOF, the hamming distance is used to represent the similarities of different instances, and the computational complexity and memory consumption can be significantly reduced. It can work well directly on the sensors and save energy consumption. Specifically, our technical contributions in this paper are as follow:

- We provide a hierarchical framework for anomaly detection in WSNs, in which data are pre-processed in the sensors before transmit to remote sever. This framework is proven to be more energy-efficient and more accurate than state-of-the-art algorithms.
- We introduce a novel LOF-based algorithm for detecting anomalies efficiently. Locality sensitive hashing is introduced to provide a compact and effective representation for the collected data.
- We present experimental results that show the capability and the performance of anomaly detection, and also verify that our algorithm is more efficient than the standard LOF algorithm. We conclude that our proposed hierarchical framework and the improved LOF algorithm are practical for the real-world applications in WSNs.

2. Related Work

Many previous studies on anomaly detection are conducted in the field of classification. Most of them base on the premise that the training data can be classified as normal or abnormal ones. The classification based methods model the labeled data to obtain a classifier, and then use the classifier to label the test data [10, 11, 12, 13]. A key drawback of classification-based methods is the requirement of the pre-labeled data, which are not always available in WSNs.

Statistic based anomaly detection approaches such as [14, 15, 16] etc. are also popular. Most of them assume that normal data instances appear in high probability region while anomalies appear in low probability region. The statistical techniques conclude a statistical model by the given data set, usually for normal behavior, and use the model to identify the unseen data instance. However, it is difficult to get an accurate reference distribution model in real world.

Clustering based methods are also widely used in anomaly detection [17]. Clustering is originally introduced in grouping similar data instances into clusters, such as K-means, DBSCAN, and GMM [18]. Normal data instances belonging to large and dense clusters is the assumption of the clustering based methods [19, 20, 21, 22]. In contrast, anomalies either belong to small or sparse clusters. However, the standard clustering based anomaly detection methods, which is originally designed for clustering, usually ignore the density of different clusters. Besides, the distribution of clusters might not be known easily in the online setting.

Another attracting approaches on anomaly detection are nearest neighbor based approaches, which are based on the assumption that normal data locate in dense region while anomalies locate in sparse region. These approaches can be grouped into two categories: using distance to k^{th} nearest neighbor and using relatively density. In the former approach, the anomaly score of a data instance is computed as the sum of its distance from nearest k nearest neighbors [23]. In [9], the authors presented a local density based anomaly

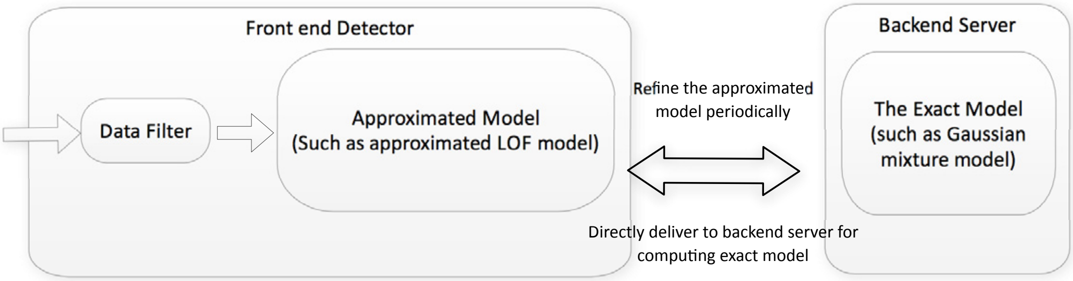

Fig. 1. Framework of Anomaly Detection System

detection approach, called local outlier factor (LOF), to compute the densities of the given data instances as their anomaly degrees. However, all these methods need to find k nearest neighbours by computing all the Euclidean distances between each two instances. The high computation and memory cost make the training phase become impractical on sensors. Instead, we proposed an approximated LOF with low computation and storage cost to satisfy the resource constraints on sensors.

3. Our Proposed Hierarchical Anomaly Detection

3.1. The Framework

The majority of the existing anomaly detection methods in WSN [7] are designed to collect raw data from sensors, and identify anomalies in the remote server. Analyzing data at the powerful remote sever, in which complicated models can be applied, provides a more accurate detection model. If there is an anomaly, the remote server then sends messages back to conduct the sensors. However, performing data transmission (i.e., receiving messages from the remote sever) frequently will cost much energy consumption for the sink nodes. It is also worth noting that learning the model at the remote server will be less adaptive to the changing environment, which widely happens in WSNs.

For energy-efficient and instantaneity, we design a hierarchical anomaly detection system showing in Fig. 1. The lightweight detector runs at the sink sensor while the more accurate model is learned at the remote server. The remote server can be a server in the cloud with high computation and huge storage capacities. It receives data from the sink sensors and periodically processes these data using accurate anomaly detection models such as Gaussian mixture model which is used in our experiment. For our lightweight detector at the sink node, it is learned by limited instances information. By taking the advantage of the local density based anomaly detection methods, we proposed an approximated LOF as the lightweight detector. Thus, we fuse the clustering based and local density based methods in our framework. Experiments on different datasets verify the effectiveness of our framework for anomaly detection tasks.

3.2. Our Proposed Approximated Local Outlier Factor

As mentioned in related work, LOF is an anomaly detection based on the local density estimation. It can be summarized as three steps. Firstly, compute the reachability distance of an object p with regarded to object o. Secondly, compute the local reachability of an object *p*. Finally, compute local outlier factor of object *p*. [9] describes the algorithm in detail. In the first step, it requires high computational cost for estimating reachability distance by Euclidean distance, especially for those data in high dimensional space.

To avoid the high computation and storage cost, we project the instances into binary vectors with fixed bits using local sensitive hashing (LSH) [24]. The LSH algorithm is widely used in similarity search and nearest neighbor search [24, 25]. LSH algorithm commits to reduce the dimensionality for high dimension data. Projected into the low dimensional subspace, similar data instances fall into the same bucket in a high probability via many rounds of hashing. All the applications use Euclidean distance to find the

Algorithm 1: Modified local sensitive hashing

```
Input: dataset $A$, $hashBitCount$
Output: hash table $T$
Randomly choose one dimension from $A$ for $hashBitCount$ times to generate vector $rd$
Compute a weight vector $w$ by choose the maximum value of each dimension
Generating weight coefficient vector $t$ according to $rd$ and $w$, $t$ = unifrnd(0,1,1,$hashBitCount$) .* $w[rd]$
Generating binary vector $v = A(:, rd) \leq repmat(t, size(A, 1), 1)$
Insert the binary vectors into the hash table $T$
```

Algorithm 2: Back end anomaly detection.

```
Input: dataset $A$, cluster number $k$, $n$
Output:parameters $Param$, hash table $T$
while triggered do
    Get the instances number and dimensions of $A$
    Get the max values of each dimension of $A$, $dimmax$.
    Estimate Gaussian mixture parameters $obj$
    Set a fixed bits number $hashBitCount$ of the binary sequence obtained by hash functions
    Generate $Param$ using $obj$, $hashBitCount$, $dimension$, $n$, and $dimmax$.
    Randomly choose $n$ instances from dataset $A$.
    Process $n$ instances by converting each into a binary vector with modified LSH.
    Construct hash table $T$.
    Send $Param$ and $T$ to the front end.
end
```

nearest neighbor in the same bucket which still keeps the high computation complexity and memory consumption. In this paper we modified the original LSH algorithm to make it suitable for WSN scenarios. We introduce a weight coefficient vector to outline the differences between dimensions. Further more, we use Hamming distance instead of Euclidean distance to cut down computation complexity. Details are shown in Algorithm 1.

In our work, we project instances into binary vectors with fixed bits using LSH, and then search its k nearest neighbors. We assume that the mean of the total distances between instances in the same bucket is d and the distance between different buckets is D. Accordingly, we have

$$\mathrm{lrd}(p) = \quad 1/\Big(\frac{dn_1 + D\sum_{2\leq i\leq hashBitCount} in_i}{k}\Big) \tag{1}$$

$$\mathrm{LOF}(p) = \Big(\sum_{o\in N_k(p)} \frac{\mathrm{lrd}(o)}{\mathrm{lrd}(p)}\Big)/k = \frac{dn_{\mathrm{p1}} + D\sum_i in_{\mathrm{pi}}}{dn_{\mathrm{o1}} + D\sum_i in_{\mathrm{oi}}}/k = \frac{n_{\mathrm{p1}} + \dfrac{D}{d}\sum_i in_{\mathrm{pi}}}{n_{\mathrm{o1}} + \dfrac{D}{d}\sum_i in_{\mathrm{oi}}}/k \tag{2}$$

We replace $\frac{D}{d}$ by gap. Supposing the hamming distance between two instances in the same bucket is 1, we set gap as 2 or higher to simply identify the difference between the same bucket and different buckets. In all the experiments described below, we set gap as 5. Finally, we sort the LOF to find the real anomalies.

In our framework, the sink sensors will maintain N instances projecting to D bits binary vector where N and D are constants. The parameters of Gaussian mixture model(GMM) generated by the whole dataset at the remote server will be kept either (see Algorithm 2) . When new instances are arriving, the sink sensors use the GMM parameters to identify whether the new instances fit the standard model, which we call filtering process. If the probability of an instance fitting the standard model is lower than the threshold, the sink sensor will take a deep step into computing the local outlier factor of the instances using N instances.

Algorithm 3: Front end anomaly detection.

```
Input: $Param$, $T$, new dataset $P$, threshold of anomaly rate $Cthresh$, threshold of LOF value $Lthresh$,
    threshold of the probability an instance belonging to a distribution model $Pthresh$
Output: $LOF$ value
while $count < Cthresh$ do
    Computing the probabilities $prob$ $P$ belongs to the standard distribution model described by
    $Param.obj$
    if $prob < Pthresh$ then
        Convert $P$ into binary matrix $v$
        Insert $v$ into hash table $T$
        Compute the $LOF$ value of $v$
        if $LOF > Lthress$ then
            $count$++
        end
    end
end
Call remote server for accurate distribution model parameters
Clean up $count$ and $T$
```

When the anomaly rate of N instances is larger than a pre-defined threshold, the sink sensors will abort the N instances information and call remote server for much more accurate distribution model parameters and another new N instances (see Algorithm 3).

In the end of this subsection, we will discuss the computation and storage complexity of our proposed LOF algorithm. The computational complexity of the standard LOF algorithm is $O(n^2)$, and kd-tree based structure make the algorithm performances terrible in high dimension situation. Our novel LOF algorithm uses LSH to reduce the dimensions and makes the retrieve complexity to $O(\log N)$. For the storage complexity, the standard LOF algorithm needs to store all the features of different observations. Assuming n observations with d features each and each data consisting of b bits, the storage complexity is $O(ndb)$. While in novel LOF algorithm, the storage complexity is $O(n \log n)$ by projecting each observation into $\log n$ bits. It is obvious that the novel LOF algorithm performs better than the standard one, especially for the high dimensional data. As the analysis above, our proposed hierarchical anomaly detection model is computation and storage saving, which makes it a desirable lightweight anomaly detector in WSN.

4. Experiments

In this section, we design experiments to compare typical outlier detection algorithms introduced in [7], which can be used as our accurate model in the remote server. Besides, we also evaluate the performance of our approximated model learned at sink nodes. Finally, we report the results by using our hierarchical framework. To evaluate these algorithms with a variety of datasets (i.e., varying in size, dimensionality etc.), we totally use 6 real-world binary classification datasets from UCI Machine Learning Repository [26]. Details of each dataset are presented in Table 1. We consider the data from the majority class as normal data and randomly select 10% data instances from the minority class as outlier samples. All the experiments are repeated with 5 random trials. For the evaluation of performance, we use the Area Under the Curve (AUC) [27] as measurement, which is typically used in evaluating anomaly detection methods.

4.1. Experiments on typical outlier detection algorithms

To determine an appropriate detection algorithm at the remote server, we test traditional anomaly detection algorithms, which include *LOF*, *kmeans*, *GMM*, *PCA* by *Biomed*, *Ionosphere*, *Pima*, *Housing* datasets. As shown in Table 2, GMM (Gaussian Mixture Model) performances most steadily among the algorithms

Table 1. Description of Datasets

Dataset	Balance-scale	Biomed	Housing	Ionosphere	Pima	Vowel
Size	625	194	506	351	768	528
Features	4	5	13	34	8	10

Table 2. AUC of Typical Algorithms

Dataset	LOF	kmeans	GMM	PCA
Biomed	0.8596	0.7966	0.9619	0.9606
Ionosphere	0.9607	0.8696	0.9848	0.9763
Pima	0.6886	0.6330	0.7392	0.7270
Housing	0.5024	0.5939	0.7549	0.7331

on different datasets. As a result, we use GMM for computing the accurate reference distribution model at the remote server. Other algorithms, which are not mentioned in the candidate list, can be also used instead if they performance better.

4.2. Parameter Settings for Our Approximated LOF

In our approximated LOF, we need to determine several parameters to approximate the LOF model. To understand how these parameters affect the performance of our method, we conduct experiments on *pima* with different parameter settings.

Considering the limited storage capacity on sensors, we have to choose N instances from the whole data to be stored on sensors as references. Larger N brings higher accuracy but also higher storage space and computation consumption. To achieve an appropriate N, different sizes ranging from 10% to 50% of the instances are tried, as shown in Fig. 2. From the results, we can observe that 30 % of the whole data can perform well while the selecting of N can be a trade-off by the user's preference.

We also need to determine the parameter HashBitCount, which is the bits of binary vectors v as mentioned above. Larger HashBitCount helps identify instances more precisely while smaller one gets little result on clustering. But higher HashBitCount takes more memory consumption. By testing on different settings of HashBitCount, we observe that $\log(n)$ as the number of HashBitCount can be an appropriate one, where n is the number of instances (Fig. 3 shows the result on *Pima*).

4.3. Experiments on Our Approximated LOF and Hierarchical Model

In this section, we first compare our approximated LOF with the standard LOF by five datasets. Besides, we also apply the approximated algorithm to the proposed hierarchical model. The AUC scores and ROC curves are shown in Table 3, Fig. 4 and Fig. 5.

As shown in Table 3, the approximated LOF can achieve competitive AUC scores by comparing with the standard LOF . In certain cases, our proposed LOF algorithm even outperforms the standard one due to the dataset composition. It is worth noting that data close to a dense cluster might be identified easily by the standard LOF. In contrast, our proposed method can enlarge the outliernesses of the sparse instances (i.e., using D in Section 3.1), and give a more discriminative ability for distinguishing normal and anomalous data. Fig. 4 and Fig. 5 show the detailed ROC curves for the hierarchical model, approximated LOF, and the standard LOF with Pima and Ionosphere datasets.

In our hierarchical model, we execute the filtering process by GMM before computing the LOF scores to decrease the computation consumption. In the filtering process, the threshold of probability described in Algorithm 3 needs to be predetermined. As shown in Table 4, we use 0.0296, 0.0949, 0.2283, 0.4224 and 1 as probability threshold according to the distribution of *Biomed*. The filtering rate drops down with a higher threshold, namely more and more instances should be detected by the approximated LOF. In this case, the computation loads will be increased for the detection procedure. That is, it is a trade-off for the performance and computational cost. In our experiments (i.e., the results in Table 3), we first estimate the probability

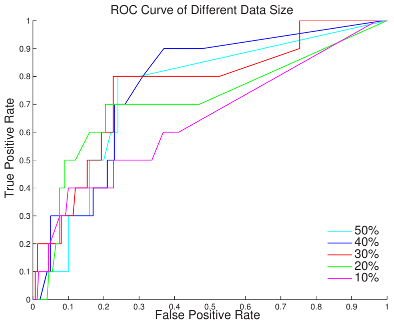

Fig. 2. ROC curves by different data size for *Pima* dataset

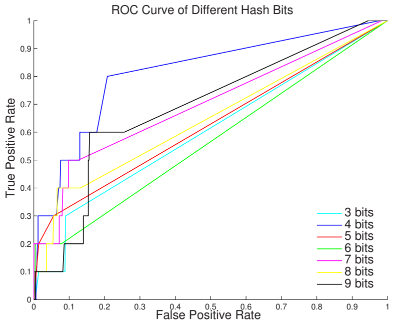

Fig. 3. ROC curves by different hash bits for *Pima* dataset

Table 3. AUCs of the standard LOF, our approximated LOF, our propped hierarchical model

Algorithm	Balance-scale	Ionosphere	Vowel	Housing	Pima
Approximated LOF	0.6841(0.1043)	0.8581(0.0473)	0.6086(0.1977)	0.6179(0.2234)	0.7572(0.0941)
Hierarchical model	0.6977(0.1198)	0.8621(0.0461)	0.6135(0.0937)	0.7059(0.1547)	0.6892(0.0843)
LOF	0.6951	0.9607	0.8781	0.5655	0.6886

of belonging to cluster centers for each instance and then calculate the mean of these probabilities as our threshold.

5. Conclusion

Finding anomalies is important for many applications. Most of the previous studies on anomaly detection cannot be directly applied to the applications with resource constraints, such as WSN applications. In this paper, we design a hierarchical framework, which includes the accurate model and the approximated model, to reduce the transmission cost by comparing with traditional centralized methods. Furthermore, we also proposed an approximated LOF utilizing the locality sensitive hashing to reduce the computational costs and memory requirements. Therefore, compared with other anomaly detection mechanisms, our proposed framework can achieve satisfactory results while significantly reducing computational costs and memory requirements. Thus, our hierarchical anomaly detection is preferable for WSN applications.

Acknowledge

This work was also supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC100-2911-I-002-001, NSC 101-2218-E-011-006, and NTU102R7501.

References

[1] X. Ma, D. Yang, S. Tang, Q. Luo, D. Zhang, S. Li, Online mining in sensor networks, in: Network and Parallel Computing, Vol. 3222, 2004, pp. 544–550.

[2] M. Sa, A. K. Rath, A simple agent based model for detecting abnormal event patterns in distributed wireless sensor networks, in: Proceedings of the 2011 International Conference on Communication, Computing & Security, ICCCS '11, ACM, New York, NY, USA, 2011, pp. 67–70.

[3] A. S. Raghuvanshi, R. Tripathi, Machine learning approach for anomaly detection in wireless sensor data, International Journal of Advances in Engineering and Technology 1 (2011) 47–61.

[4] Y. Zhang, N. Meratnia, P. Havinga, Outlier detection techniques for wireless sensor networks: A survey, Communications Surveys Tutorials, IEEE 12 (2) (2010) 159 –170.

[5] P. Tan, M. Steinback, V. Kumar, Introduction to Data Mining, Addison Wesley, May 12, 2005.

[6] E. M. Knorr, R. T. Ng, Algorithms for mining distance-based outliers in large datasets, 1998, pp. 392–403.

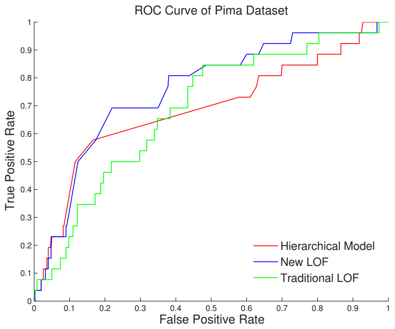

Fig. 4. ROC curves of traditional, novel LOF and the hierarchical model on *Pima* dataset

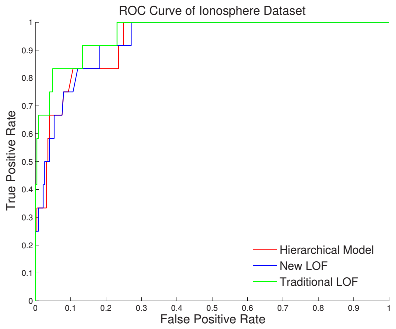

Fig. 5. ROC curves of traditional, novel LOF and the hierarchical model on *Ionosphere* dataset

Table 4. AUC of Different Threshold on Biomed

threshold	0.0296	0.0949	0.2283	0.4224	1.0000
filtering rate	81.20%	60.90%	41.35%	21.05%	0
AUC	0.6516	0.6657	0.6929	0.7073	0.7600
standard deviation	0.0047	0.0130	0.0348	0.0320	0.0231

[7] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 15:1–15:58.

[8] N. Aner, M. Goldstein, Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer, in: S. Fischer, I. Mierswa (Eds.), Proceedings of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012), Shaker Verlag GmbH, 2012, pp. 1–12.

[9] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, Lof: Identifying density-based local outliers, in: W. Chen, J. F. Naughton, P. A. Bernstein (Eds.), Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA, ACM, 2000, pp. 93–104.

[10] L. D. Baker, T. Hofmann, A. K. Mccallum, Y. Yang, A hierarchical probabilistic model for novelty detection in text (1999).

[11] D. Barbara, N. Wu, S. Jajodia, Detecting novel network intrusions using bayes estimators, in: Proc. SIAM Intl. Conf. Data Mining, 2001.

[12] G. Ratsch, S. Mika, B. Scholkopf, K.-R. Muller, Constructing boosting algorithms from svms: an application to one-class classification, Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (9) (2002) 1184 – 1199.

[13] Q. Song, W. Hu, W. Xie, Robust support vector machine with bullet hole image classification, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 32 (4) (2002) 440 –448.

[14] W. R. Young, Outliers in statistical data, Technometrics 22 (4) (1980) 631–631.

[15] M. Lauer, A mixture approach to novelty detection using training data with outliers, in: Proceedings of the 12th European Conference on Machine Learning, EMCL '01, 2001, pp. 300–311.

[16] M. Mahoney, P. Chan, Learning rules for anomaly detection of hostile network traffic, in: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, 2003, pp. 601 – 604.

[17] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.

[18] R. Xu, I. Wunsch, D., Survey of clustering algorithms, Neural Networks, IEEE Transactions on 16 (3) (2005) 645 –678.

[19] S. Basu, M. Bilenko, R. J. Mooney, A probabilistic framework for semi-supervised clustering, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04, ACM, New York, NY, USA, 2004, pp. 59–68.

[20] S. Guha, R. Rastogi, K. Shim, Rock: A robust clustering algorithm for categorical attributes, in: In Proc. of the 15th Int. Conf. on Data Engineering, 2000.

[21] Z. He, S. Deng, X. Xu, Outlier detection integrating semantic knowledge, in: Proceedings of the Third International Conference on Advances in Web-Age Information Management, WAIM '02, Springer-Verlag, London, UK, UK, 2002, pp. 126–131.

[22] Z. He, X. Xu, S. Deng, Discovering cluster based local outliers, Pattern Recognition Letters 2003 (2003) 9–10.

[23] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo, A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data, in: Applications of Data Mining in Computer Security, Kluwer, 2002.

[24] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: Proceedings of the 25th International Conference on Very Large Data Bases, VLDB '99, Morgan Kaufmann Publishers Inc., 1999, pp. 518–529.

[25] P. Indyk, R. Motwani, Approximate nearest neighbors: Towards removing the curse of dimensionality, 1998, pp. 604–613.

[26] D. N. A. Asuncion, UCI machine learning repository, `http://www.ics.uci.edu/~mlearn/MLRepository.html` (2007).

[27] J. Myerson, L. Green, M. Warusawitharana, Area under the curve as a measure of discounting (2001).