SAND2013-7612C

Host-based Anomalous Behavior Detection Using Cluster-Level Markov Networks

Ahmad Mustafa, Mohiuddin Solaimani, Justin Sahs, Latifur Khan
Department of Computer Science
The University of Texas at Dallas
{amm106220, mxs121731, justin.sahs, lkhan}@utdallas.edu

Ken Chiang and Joe Ingram
Sandia National Laboratories
{kchiang, jbingra}@sandia.gov

***Abstract*—Anomaly detection is a common component in computer security. This paper proposes a host-based anomaly detection method based on k-means clustering and Markov networks. First, the training data are clustered. Then, in each cluster, a separate Markov network is built to model the underlying benign behavior. During testing, each Markov network predicts the probability for each previously-unseen instance. If the probability from multiple Markov networks is low, the point is classified as malicious. We experimentally show that our proposed approach outperforms several other approaches, while being less sensitive to label noise.**

***Index Terms*—Computer security, host-based anomaly detection, Markov random fields**

I. Introduction

Anomaly detection [1] refers to the problem of detecting patterns in data that do not conform to a normal or expected behavior. These patterns are called anomalies or outliers. An outlier [2] can be defined as a data instance which greatly deviates from the other instances, generally because it was generated by a different underlying mechanism. In network security an anomaly can be any illegitimate/malicious instance that deviates from normal/benign data.

Anomaly detection aims to find malicious data by examining the data records under the assumption that the malicious instances will somehow differ from the benign instances. The detection process can be divided into two categories: 1) host-based [3] in which the system monitors operating system events like system calls of each computer in the network and 2) network-based [4] in which the system monitors the network traffic data. Traditionally, network-based anomaly detection focuses on data received from an outsider. Therefore, it cannot detect malicious data generated by an insider or malicious data that is being sent without generating abnormal network traffic.

Our system is a host-based anomaly detection system which collects periodic snapshots of system calls from machines on the network. Then, it examines them by comparing each snapshot to a prior model which is trained using historical data. The objective is to classify the snapshots as benign or malicious.

Various approaches can be used to train and use the model. One approach assumes that there is enough training data to represent both the benign and malicious data in the model. So, the model learns the patterns of both classes of data. Typically, the training dataset may not be balanced. That is, one class may have few instances and the other may have a large number of instances. This imbalance may cause a problem in training. In our case, the malicious training data are rare and it is not possible to represent all of the "characteristics" of malicious behavior during training. In other words, some malicious behavior may not appear in the training data. Hence, it is difficult to come up with malicious data *a priori*. In this paper we follow two different approaches. The first uses only benign training data which we call *unsupervised* learning, and the second uses benign and few malicious data to train the model which we call *semi-supervised* learning.

In this paper, we present a host-based anomaly detection method, *clustered Markov networks* (CMN). CMN first applies k-means clustering to the benign training data, and then uses Markov networks to probabilistically model each cluster. During testing, each Markov network predicts the probability of each testing instance. If the average predicted probability is below a threshold, the system declares the instance malicious.

We experimentally compare our methods with existing methods, including *label propagation* (LP) [5], *Markov networks* (MN) [6], and *k-means outlier detection* (KMOD) [7]. We also investigate *clustered label propagation* (CLP) and compare it to our CMN approach. CLP starts by applying k-means clustering to training data with both benign and malicious instances. It then labels each cluster based on the cluster's central-most point. At testing time, these points are added to the testing data as labeled points and label propagation, a semi-supervised learning algorithm, is used to label the testing data. Our experiments show that CMN outperforms these other methods, because it is able to model the local characteristics of the benign data.

This paper makes the following contributions:

- We propose a novel anomaly detection algorithm (CMN) in which we combine k-means clustering with Markov networks.
- We compare our approach with other approaches, namely: Markov networks (MN) (without clustering), k-means outlier detection (KMOD), label propagation (LP) (without clustering), and clustered label propagation (CLP).
- We show that our CMN approach outperforms the other methods in terms of the detection of true malicious data without adding additional false alarms.

The rest of this paper is organized as follows: Sec-

tion II presents technical background information; Section III presents our Markov network and label propagation techniques; Section IV presents our experimental results; Section V presents related work; finally, Section VI concludes the paper.

II. Background

Our work relies on a number of methods, including k-means clustering, Markov networks, and label propagation. We review these algorithms here.

A. K-means Clustering

K-means [8] is a clustering algorithm which partitions the data into k optimal clusters. In k-means, each data point is assumed to be a d-dimensional vector. Each cluster has a "centroid," which is the average vector of all points assigned to that cluster. Each point is assigned to the cluster with the nearest centroid. The algorithm procedes in an expectation-maximization style: first, k centroids are generated randomly. Then, the data are assigned to the nearest centroid, and the centroids are updated accordingly. This process repeats until some convergence criterion is met.

However, k-means can be sensitive to the initial random choice of centroids. So, we run the algorithm multiple times with different random centroids. We then report the clustering that leads to the most accurate results.

B. Markov Networks

Markov networks [6] (also known as Markov random fields) are a form of *probabilistic graphical models*, which specify probability distributions that model data. In Markov networks, the distribution is represented by an undirected graph $G = (V, E)$ where nodes represent random variables (i.e, features) and edges encode independence properties of the joint distribution. Specifically, two variables are independent if there is no path between them. Similarly, two variables are conditionally dependent if there is a path between them. Additionally, each Markov network has a set of "factors" associated with subsets of V. These factors are functions $\phi_i : val(D_i) -> \mathbb{R}$, where $D_i \subset V$, and $val(D_i)$ denotes the possible value combinations of the variables in D_i. Then, the Markov network defines a probability distribution

$$P(\vec{x}) = \frac{1}{Z} \prod_i \phi_i(\vec{x}),$$

where Z is a normalization constant,

$$Z = \sum_{\vec{x} \in val(V)} \prod_i \phi_i(\vec{x}).$$

We calculate $P(\vec{x})$ without normalization because computing Z is computationally difficult. However, this does not affect the final results when all the instances are tested using the same models.

Algorithm 1: Label Propagation

input : The data points X and their classifications Y, $y_i \in \{-1, 0, 1\}$, where 0 represents unlabeled data.

output: The predicted labels of unlabeled points.

begin
  Compute W from X;
  Compute the diagonal matrix D by $D_{ii} \Leftarrow \sum_j w_{ij}$;
  $\widehat{Y}^{(0)} \leftarrow Y$;
  repeat
    $\widehat{Y}^{(t+1)} \leftarrow D^{-1} W \widehat{Y}^{(t)}$;
    $\widehat{Y}_l^{(t+1)} \leftarrow Y_l$
  until *convergence to* $\widehat{Y}^{(\infty)}$;
  $\widehat{y}_i \leftarrow \text{sign}(\widehat{y}_i^{(\infty)})$;

C. Label Propagation

Label propagation [5], [9] is a semi-supervised learning algorithm in which labeled and unlabeled data are used to form a similarity or "affinity" matrix W, where w_{ij} represents the similarity between data points i and j. A common choice of similarity measure is the Gaussian kernel,

$$w_{ij} = e^{-\frac{\|x_i - x_j\|}{2\sigma^2}},$$

where σ is the width parameter. In general, we assume that $w_{ij} = W_X(x_i, x_j)$, where $W_X(\cdot)$ is a symmetric, positive function that may depend on the entire dataset X.

Given W, the label propagation algorithm is based on a simple idea: labeled nodes propagate their labels to their neighbors, with this propagation being weighted by the similarity measure W. A variation of the label propagation algorithm is given in Algorithm 1. This version is restricted to two classes [5].

III. Our Approach

We present two novel approaches for anomaly detection: clustered Markov networks (CMN) and clustered label propagation (CLP). We assume that we have only benign labeled data for CMN and that we have both benign and malicious data in CLP.

A. Clustered Markov Networks (CMN)

Markov networks (MN) can be used as an anomaly detection method. The networks are trained using the "benign" training data only. All the vertices (i.e, features) in the MN are connected to each other. So, the network is a completely connected graph. Each factor in MN consists of only two features. Each factor is built by counting the number of co-occurrences of each value of its first feature with each value of its second feature. Then, the network is used as a model to predict the probabilities of the instances to be examined. If the probability of the instance is high, then the instance is classified as benign. Conversely, if its probability is lower

Algorithm 2: Training Clustered Markov Networks

input : Training data D_{tr}, consisting of "benign" data; k, the number of clusters.
output: An ensemble E of Markov networks.

1 **begin**
2 $E \leftarrow \emptyset$;
3 **for** $c \in k\text{-Means}(D_{tr})$ **do**
4 $E \leftarrow E \cup \text{BuildMN}(c)$;

than some threshold h, then it is classified as malicious. The threshold h can be specified in multiple ways. In all of our networks, we take h to be the average probability of all the testing instances plus their variance.

We call the previous approach a "global" Markov network, because it learns from the whole training dataset without considering that the training data may have multiple types of benign characteristics. The results show that the false negative rate is high and the true positive rate is low, which indicates that the model is missing multiple types of malicious data. This drawback may be due to the heterogeneity of the training data, which may lead to overestimated MNs.

To solve this problem, we use the CMN approach. We assume that the normal data with similar characteristics will have minimum distances between each other. In CMN the training data are first clustered using k-means, then "local" MNs are trained on each cluster independently, as shown in Algorithm 2. Line 3 shows that k-means clustering is applied to the training data D_{tr}. In line 4, a Markov network is built using each cluster c and the resulting networks are stored in an ensemble E.

During testing, each local network independently predicts the probability of the instance. Then, the resulting probability predictions are averaged to obtain an overall prediction. If the average probability of an instance is less than some threshold, it is classified as malicious; otherwise, it is classified as benign, as shown in Algorithm 3. In line 2, each testing instance d_i is retrieved from the testing dataset D_t. Then, in lines 3 and 4, each Markov network M_j in ensemble E calculates the probability p_j of d_i. In line 5, the average probability, P_{avg}, of all p_j's is calculated and compared to the threshold h as stated from lines 6 to 9. As stated before, we calculate h by taking the average probability of all the testing instances plus their variance.

Clustering boosts the performance of the networks. It allows MNs to be built from data that have similar characteristics. This leads to MNs that are more capable of detecting malicious data. The experimental results show that CMN is able to detect a higher number of malicious instances.

B. Clustered Label Propagation (CLP)

Next, we present a variation of the label propagation (LP) algorithm (described in Section II-C) called clustered label propagation (CLP). In this method, k-means clustering is

Algorithm 3: Testing with Clustered Markov Networks

input : Testing data D_t, consisting of "benign" data; E, an ensemble of Markov networks; h, the threshold.
output: $\widehat{D_t} = \{\widehat{d_i}\}$, the predicted classifications of the data points D_t.

1 **begin**
2 **for** $d_i \in D_t$ **do**
3 **for** $M_j \in E$ **do**
4 $p_j \leftarrow \text{Test}(M_j, d_i)$;
5 $P_{avg} \leftarrow \frac{\sum_{j=1}^{|E|} p_j}{|E|}$;
6 **if** $P_{avg} > h$ **then**
7 $\widehat{d_i} \leftarrow -1$;
8 **else**
9 $\widehat{d_i} \leftarrow 1$;

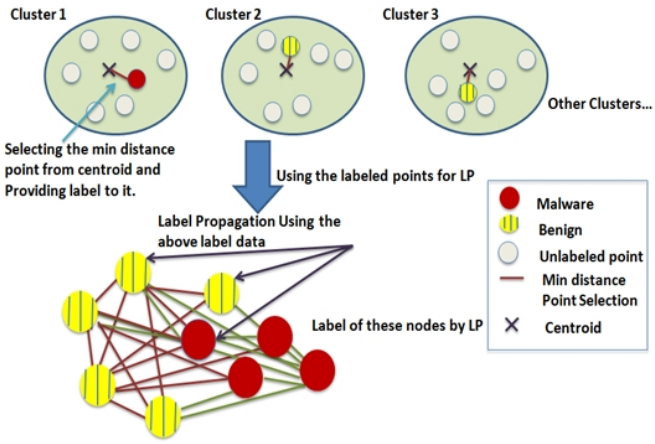

Fig. 1: Clustered Label Propagation (CLP)

combined with the label propagation algorithm. The procedure is shown in Algorithm 4.

Here, the training dataset, which contains both benign and malicious instances, is intelligently selected using clustering. Initially, there are two empty sets: labeled point set P_l and its corresponding label set Y_l. K-means clustering is applied to the training dataset D_{tr} to create k clusters c_1,c_2,...,c_k. For each cluster, the closest point p_{min} to the centroid p_c (using Euclidean distance) is selected and inserted into P_l while its corresponding label y is inserted into Y_l. Finally, label propagation is applied using P_l and Y_l to label the unlabeled points.

Fig. 1 illustrates the clustered label propagation method. The top of the figure shows the clusters for the training dataset. For each cluster, the closest point to the centroid and its label are selected. In the figure, the centroid is represented by a cross. One can see in cluster 1 that the closest point to its centroid is labeled as malicious (i.e, the shaded circle). Similarly, cluster

Algorithm 4: Training Clustered Label Propagation (CLP)

input : Training Dataset D_{tr}.
output: Labeled point set P_l and its corresponding label set Y_l.

```
begin
    P_l={ }, Y_l={ };
    C_s ← Cluster (D_tr) where C_s ={c_1,c_2,...,c_k};
    for each cluster c_i ∈ C_s do
        p_c ← Center(c_i);
        p_min ← FindMinDistPoint( p_c );
        y ← Label(p_min);
        Y_l ← Y_l ∪ {y};
        P_l ← P_l { p_min };
```

Algorithm 5: Testing with Clustered Label Propagation (CLP)

input : Testing Dataset D_t, labeled point set P_l and its corresponding label set Y_l.
output: The predicted labels for unlabeled points.

```
begin
    D_t ← D_t ∪ P_l;
    Run label propagation with Y_l;
```

2 and 3 have points nearest to the centroid which are benign (i.e, the striped circles). These labeled points are then used to run the label propagation (LP), which labels all the testing data.

The label propagation (LP) algorithm is an efficient algorithm for classification. It constructs a similarity graph over all items in the input dataset. However, LP needs some labeled data initially and finding these labeled data can be an expensive task. Clustered label propagation (CLP) has introduced the clustering feature into the label propagation algorithm. CLP requires fewer labeled data instances and these instances are more influential to the label propagation algorithm. Thus, CLP reduces the total number of initial labels needed. (Note that LP and CLP require both benign and malicious data for training and there are few instances in the malicious class.)

C. K-means Outlier Detection (KMOD)

For this method [7], only benign instances are used to train the model. During training, k-means clustering is first applied to the dataset. Then, for each test instance, the cluster with the nearest centroid is found using Euclidean distance. Finally, if the distance between the point and the centroid of the cluster is smaller than the distance between the centroid and farthest point of the cluster, it is normal. Otherwise, it is classified as anomalous.

IV. Experimental Results

We have evaluated the discussed methods on a dataset of system calls, where each instance is a snapshot of system calls collected from a running executable during its first two minutes of execution. Each instance is a binary vector representing the presence or absence of a system call during execution; there are a total of 284 possible system calls. Each instance is also labeled as either benign or malicious. The benign samples were collected from live feeds of all files that crossed a corporate network border. In order to mitigate the risk of having potentially malicious samples in this feed, the samples were filtered through anti-virus scanners before labeling the data stream as "good". For malware, we used a daily feed from Arbor Networks, a security company that collects malicious software from the many network sensors that they own. The dataset contains 24,000 instances, of which 23,000 are benign and 1,000 are malicious.

	Training		Testing	
Method	# Benign	# Malicious	# Benign	# Malicious
MN	20,700	0	2,300	1,000
CMN	20,700	0	2,300	1,000
KMOD	20,700	0	2,300	1,000
LP	19,837	863	3,163	137
CLP	19,837	863	3,163	137

TABLE I: Dataset Sizes

Weka [10] was used to implement k-means in the CMN and KMOD experiments. The CLP and LP experiments were implemented in Python using the Scikit-Learn [11] implementation of label propagation and k-means.

A. Overall Performance

The Markov networks (MN), clustered Markov networks (CMN), and k-means outlier detection (KMOD) approaches were trained using 20,700 benign instances and tested on the remaining 1,000 malicious and 2,300 benign instances. However, as the label propagation (LP) and clustered label propagation (CLP) require both benign and malicious training data, the same training data cannot be used. The LP and CLP approaches were therefore trained with 19,837 benign and 863 malicious randomly-selected instances. These numbers are summarized in TABLE I.

For CLP, we have set the number of clusters k to 2,070. In CLP, we take only one point (the nearest point to the centroid) with its label from each cluster. If there are a smaller number of clusters, then there are fewer labeled points for the label propagation algorithm. So, the algorithm does not spread the labels to the unlabeled points appropriately with these fewer labeled points. For the CMN and KMOD experiments, we varied the number of clusters k from 2 to 10 and have reported the results in Fig. 3.

Our results show that CMN outperforms the other methods in terms of maximizing the true positive rate and F_2 measure while minimizing the false negative and false positive rates. The best results of all approaches are shown in TABLE II. The value of F_2 in CMN is 0.828 and it is higher than MN, KMOD, LP, and CLP which all have F_2 values of 0.483, 0.668, 0.113, and 0.0 respectively. This confirms our claim that using clustering and then building a Markov network on

Method	F_2	TPR	FPR	FNR	TNR
MN	0.483	0.571	0.582	0.429	0.417
CMN	**0.828**	**0.845**	**0.112**	**0.155**	**0.887**
KMOD	0.668	0.837	0.622	0.163	0.378
LP	0.113	0.188	0.179	0.811	0.82
CLP	0.0	0.0	0.172	1.0	0.827

TABLE II: Experimental Results

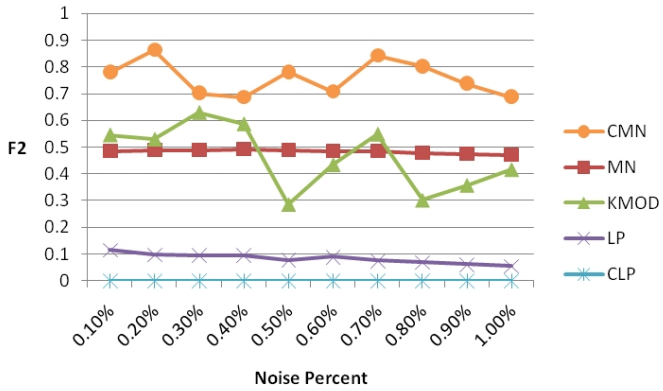

Fig. 2: Effect of Label Noise on F_2 Measure

each cluster increases the number of discovered true positives without adding false positives.

B. Sensitivity to Mislabeled Data

We have also investigated the effect of having label noise in the training data. Here, label noise refers to malicious instances that have been labeled as benign. We have varied the percentage of noise from 0.01% to 1% and reported the results in Fig. 2. The x-axis is the percentage of noisy data embedded in the training data and the y-axis is the F_2 values for each method. The figure shows that CMN outperforms MN, KMOD, LP, and CLP. For example, when 1% of the training data is mislabeled, F_2 value of CMN is 0.7 while it is 0.471, 0.416, 0.055, and 0.0 for MN, KMOD, LP, and CLP respectively. This clearly shows that CMN is less sensitive to noise.

C. Sensitivity to Number of Clusters

Additionally, we have varied the number of clusters k in CMN and KMOD from 2 to 10 to estimate the effect of cluster size on performance. The results are reported in Fig. 3. The x-axis represents k and the y-axis represents F_2. The value of F_2 has increased from 0.5 using MN to 0.8 on average using CMN. MN is represented in Fig. 3 by a straight line, because it does not use the clustering method. The F_2 measure for KMOD increases as k increases. However, the false positive rate (FPR) also increases. For example, using 10 clusters, the F_2 values for KMOD and CMN are 0.828 and 0.668 respectively, while the FPR values are 0.076 and 0.622.

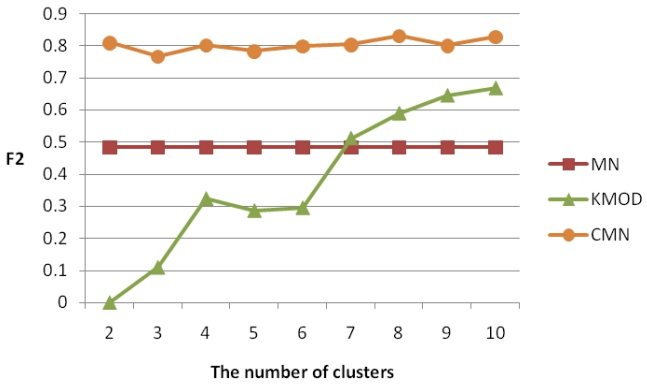

Fig. 3: Effect of Cluster Size on F_2 Measure

V. Related Work

Anomaly detection methods typically fit into one of a number of broad categories:

- *Distance-based* [12]: classification is done based on the number of points in the neighborhood of the examined point. The neighborhood of a point can be defined as the points that are closer than some distance d.
- *Density-based* [13]: classification is done based on the density of the instances neighborhood. So if the neighborhood of the instance is dense then it is considered normal; otherwise it is abnormal.
- *Statistical-based* [14]: classification is based on a probabilistic model; if the test instance is given sufficiently low probability by the model, it is considered an anomaly.
- *Clustering-based* [15]: Classification is based on clustering method. Clusters with small size are considered as outliers.

Knorr *et al.* [16] suggest a method that combines the distance and density approaches, wherein a test instance is considered an anomaly if the fraction of training instances that lie within a given radius is below some threshold. However, this approach requires setting the radius and the threshold, which may be hard to estimate *a priori*. In our approach, we do not set a threshold, instead it is calculated from the data.

Breunig *et al.* [17] present a density-based approach called *local outlier factor* (LOF) in which the data instance is assigned an outlier score which is equal to the ratio of average local density of the k-nearest neighbors of the instance and the local density of the data instance itself. However, the anomalous instances that are not far enough from other instances are missed and the use of k-nearest neighbors can be very expensive depending on the distance function.

The statistical approach [14] has the following advantages. First, it is mathematically justified by using well-established statistical methods to detect outliers. Second, if the model represents the normal data, then the results are accurate and

the methods are very efficient. Third, there is no need to keep the training data after building the model. On the other hand, a disadvantage is how to build the best model that represents the normal data from the training data.

He *et al.* [15] propose a clustering-based method in which each test instance is assigned an outlier score called the *cluster-based local outlier factor*. The score is measured by the size of the cluster that the instance belongs to and the distance between the instance and the next closest cluster. A disadvantage is that the clustering methods are designed to cluster the instances in groups, so they are not optimized for anomaly detection.

We present a host-based anomaly detection algorithm, CMN, which uses only normal data to train the model. In CMN, the clustering-based approach is combined with the statistical approach. The training data is clustered and a separate Markov network is built from each cluster. All of the networks are then used as an ensemble to classify the instances such that an outlier is an instance that does not "fit" into any of the models. In this way, we produce models that together have better ability to detect malicious data when compared to using Markov networks without clustering. Clustering allows the models to consider the local characteristics of the data and not be confused by the possible heterogeneity of the entire dataset. Empirical study shows that it is more effective for anomaly detection when compared to other similar approaches.

VI. Conclusion and Future Work

In this paper, we have presented a novel anomaly detection approach: clustered Markov networks (CMN) in which we first use k-means clustering. We then build an ensemble of Markov networks (one per cluster) which independently predicts the probabilities of the test instances, and if the average probability of the test instance passes a threshold, it is classified as normal.

We experimentally compared our proposed approaches to several other approaches on a real dataset on system calls, and show that CMN outperforms the other methods. We also show that CMN is less sensitive to noise as compared to other approaches.

In the future, we want to improve our approaches by analyzing the quality of both the clusters and the markov networks. Then we can use this analysis to discard or combine some clusters or Markov networks.

Acknowledgments

Funding for this work came from the Laboratory Directed Research and Development program at Sandia National Laboratories.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[2] D. Hawkins, *Identification of Outliers*. Chapman and Hall, 1980.

[3] S. Freeman, A. Bivens, J. Branch, and B. Szymanski, "Host-based intrusion detection using user signatures," in *Proceedings of the Research Conference. RPI, Troy, NY*, 2002.

[4] L. Khan, M. Awad, and B. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," *The VLDB Journal*, vol. 16, no. 4, pp. 507–521, Oct. 2007. [Online]. Available: http://dx.doi.org/10.1007/s00778-006-0002-5

[5] Y. Bengio, O. Dellalleau, and N. L. Roux, *Semi-supervised learning*, 2006, ch. Label propagation and quadratic criterion.

[6] D. Koller, N. Friedman, L. Getoor, and B. Taskar, *Graphical Models in a Nutshell*. MIT Press, 2007.

[7] C. Aggarwal, *Outlier Analysis*. Springer-Verlag New York Incorporated, 2013. [Online]. Available: http://books.google.com/books?id=900CkgEACAAJ

[8] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. University of California Press, 1967, pp. 281–297.

[9] X. Zhu and Z. Ghahramani, "Learning from Labeled and Unlabeled Data with Label Propagation," Tech. Rep., 2002.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and D. E., "Scikit-learn: Machine Learning in Python ," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] T. Hu and S. Y. Sung, "Detecting pattern-based outliers," *Pattern Recogn. Lett.*, vol. 24, no. 16, pp. 3059–3068, Dec. 2003. [Online]. Available: http://dx.doi.org/10.1016/S0167-8655(03)00165-X

[13] W. Jin, A. K. H. Tung, and J. Han, "Mining top-n local outliers in large databases," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 293–298. [Online]. Available: http://doi.acm.org/10.1145/502512.502554

[14] M. I. Petrovskiy, "Outlier detection algorithms in data mining systems," *Program. Comput. Softw.*, vol. 29, no. 4, pp. 228–237, Jul. 2003. [Online]. Available: http://dx.doi.org/10.1023/A:1024974810270

[15] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recogn. Lett.*, vol. 24, no. 9-10, pp. 1641–1650, Jun. 2003. [Online]. Available: http://dx.doi.org/10.1016/S0167-8655(03)00003-5

[16] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," *The VLDB Journal*, vol. 8, no. 3-4, pp. 237–253, Feb. 2000. [Online]. Available: http://dx.doi.org/10.1007/s007780050006

[17] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 93–104. [Online]. Available: http://doi.acm.org/10.1145/342009.335388