Anomaly Detection from Network Logs Using Diffusion Maps

Tuomo Sipola, Antti Juvonen, and Joel Lehtonen*

Department of Mathematical Information Technology
University of Jyväskylä, Finland
{tuomo.sipola,antti.juvonen}@jyu.fi, joel.lehtonen@iki.fi

Abstract. The goal of this study is to detect anomalous queries from network logs using a dimensionality reduction framework. The fequencies of 2-grams in queries are extracted to a feature matrix. Dimensionality reduction is done by applying diffusion maps. The method is adaptive and thus does not need training before analysis. We tested the method with data that includes normal and intrusive traffic to a web server. This approach finds all intrusions in the dataset.

Keywords: intrusion detection, anomaly detection, n-grams, diffusion map, data mining, machine learning.

1 Introduction

The goal of this paper is to present an adaptive way to detect security attacks from network log data. All networks and systems can be vulnerable to different types of intrusions. Such attacks can exploit e.g. legitimate features, misconfigurations, programming mistakes or buffer overflows [15]. This is why *intrusion detection systems* are needed. An intrusion detection system gathers data from the network, stores this data to logfiles and analyzes it to find malicious or anomalous traffic [19]. Systems can be vulnerable to previously unknown attacks. Because usually these attacks differ from the normal network traffic, they can be found using anomaly detection [2].

In modern networks clients request and send information using queries. In HTTP traffic these queries are strings containing arguments and values. It is easy to manipulate such queries to include malicious attacks. These injection attacks try to create requests that corrupt the server or collect confidential information [18]. Therefore, it is important to analyze data collected from logfiles.

An *anomaly* is a pattern in data that is different from the well defined normal data [2]. In network data, this usually means an intrusion. There are two main approaches for detecting intrusions from network data: *misuse detection* and *anomaly detection* [19]. Misuse detection means using predefined attack signatures to detect the attacks, which is usually accurate but detecting new types of

* Now with C2 SmartLight Oy.

L. Iliadis and C. Jayne (Eds.): EANN/AIAI 2011, Part I, IFIP AICT 363, pp. 172–181, 2011.

attacks is not possible. In anomaly detection the goal is to find actions that somehow deviate from normal traffic. This way it is possible to detect previously unknown attacks. However, not all anomalous traffic is intrusive. This means there might be more false alarms. Different kinds of machine learning based methods, such as self-organizing maps and support vector machines, have been used in anomaly detection [20,23]. Information about other anomaly detection methods can be found in the literature [19]. *Unsupervised anomaly detection* techniques are most usable in this case, because no normal training data is required [2].

This study takes the approach of dimensionality reduction. Diffusion map is a manifold learning method that maps high-dimensional data to a low-dimensional diffusion space [5]. It provides tools for visualization and clustering [6]. The basic idea behind any manifold learning method is the eigen-decomposition of a similarity matrix. By unfolding the manifold it reveals the underlying structure of the data that is originally embedded in the high-dimensional space [1]. Diffusion maps have been applied to various data mining problems. These include vehicle classification by sound [21], music tonality [10], sensor fusion [12], radio network problem detection [25] and detection of injection attacks [8]. Advantages of this approach are that the dimensionality of the data is reduced and that it can be used unsupervised [2].

2 Method

2.1 Feature Extraction

First let us define an n-gram as a consecutive sequence of n characters [7]. For example, the string *ababc* contains unique 2-grams *ab*, *ba* and *bc*. The 2-gram *ab* appears twice, thus having frequency of 2. A list of tokens of text can be represented with a vector consisting of n-gram frequencies [7]. Feature vector describing this string would be $x_{ababc} = [2,\ 1,\ 1]$. The only features extracted are n-gram frequencies. Furthermore, syntactic features of the input strings might reveal the differences between normal and anomalous behavior. Computed n-grams can extract features that describe these differences.

The frequencies are collected to a feature matrix X whose rows correspond to lines in logfiles and columns to features. These n-gram frequencies are key-value fields, variable-length by definition. Key strings are ignored and 2-grams are produced from each parameter value. The count of occurrences of every occurring 2-gram is summed. In practice n-gram tables produced from real-life data are very sparse, containing columns in which there are only zero occurrences. To minimize the number of columns, the processing is done in two passes. If a column contains no variation between entries, that column is not present in the final numeric matrix X. That makes it reasonable to use diffusion maps to process n-gram tables directly with no further preprocessing.

2.2 Dimensionality Reduction

The number of extracted features is so large that dimensionality reduction is performed using diffusion maps. It is a manifold learning method that embeds the

original high-dimensional space into a low-dimensional diffusion space. Anomaly detection and clustering are easier in this embedded space [6].

The recorded data describe the behavior of the system. Let this data be $X = \{x_1, x_2, \ldots, x_N\}, x_i \in \mathbb{R}^n$. Here N is the number of samples and n the dimension of the original data. In practice the data is a $N \times n$ matrix with features as columns and each sample as rows.

At first, an affinity matrix W is constructed. This calculation takes most of the computation time. The matrix describes the distances between the points. This study uses the common Gaussian kernel with Euclidean distance measure, as in equation 1 [6,16].

$$W_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{\epsilon}\right) \tag{1}$$

The affinity neighborhood is defined by ϵ. Choosing the parameter ϵ is not trivial. It should be large enough to cover the local neighborhood but small so that it does not cover too much of it [21].

The rows of the affinity matrix are normalized using the diagonal matrix D, which contains the row sums of the matrix W on its diagonal.

$$D_{ii} = \sum_{j=1}^{N} W_{ij} \tag{2}$$

P expresses normalization that represents the probability of transforming from one state to another. Now the sum of each row is 1.

$$P = D^{-1}W \tag{3}$$

Next we need to obtain the eigenvalues of this transition probability matrix. The eigenvalues of P are the same with the conjugate matrix in equation 4. The eigenvectors of P can be derived from $\tilde{P}$ as shown later.

$$\tilde{P} = D^{\frac{1}{2}} P D^{-\frac{1}{2}} \tag{4}$$

If we substitute the P in equation 4 with the one in equation 3, we get the symmetric probability matrix $\tilde{P}$ in equation 5. It is called the normalized graph Laplacian [4] and it preserves the eigenvalues [16].

$$\tilde{P} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{5}$$

This symmetric matrix is then decomposed with singular value decomposition (SVD). Because $\tilde{P}$ is a normal matrix, spectral theorem states that such a matrix is decomposed with SVD: $\tilde{P} = U \Lambda U^*$. The eigenvalues on the diagonal of $\Lambda = \mathrm{diag}([\lambda_1, \lambda_2, \ldots, \lambda_N])$ correspond to the eigenvalues of the same matrix $\tilde{P}$ because it is symmetric. Matrix $U = [u_1, u_2, \ldots, u_N]$ contains in its columns the N eigenvectors u_k of $\tilde{P}$. Furthermore, because $\tilde{P}$ is conjugate with P, these two

matrices share their eigenvalues. However, to calculate the right eigenvectors v_k of P, we use equation 6 and get them in the columns of $V = [v_1, v_2, \ldots, v_N]$ [16].

$$V = D^{-\frac{1}{2}}U \tag{6}$$

The coordinates of a data point in the embedded space using eigenvalues in Λ and eigenvectors in V are in the matrix Ψ in equation 7. The rows correspond to the samples and the columns to the new embedded coordinates [6].

$$\Psi = V\Lambda \tag{7}$$

Strictly speaking, the eigenvalues should be raised to the power of t. This scale parameter t tells how many time steps are being considered when moving from data point to another. Here we have set it $t = 1$ [6].

With suitable ϵ the decay of the spectrum is fast. Only d components are needed for the diffusion map for sufficient accuracy. It should be noted that the first eigenvector v_1 is constant and is left out. Using only the next d components the diffusion map for original data point x_i is presented in equation 8. Here $v_k(x_i)$ corresponds to the ith element of kth eigenvector [6].

$$\Psi_d : x_i \rightarrow [\lambda_2 v_2(x_i), \lambda_3 v_3(x_i), \ldots, \lambda_{d+1} v_{d+1}(x_i)] \tag{8}$$

This diffusion map embeds the known point x_i to a d-dimensional space. Dimension of the data is reduced from n to d. If desired, the diffusion map may be scaled by dividing the coordinates with λ_1.

2.3 Anomaly Detection

After obtaining the low-dimensional presentation of the data it is easier to cluster the samples. Because spectral methods reveal the manifold, this clustering is called spectral clustering. This method reveals the normal and anomalous samples [13]. Alternatively, k-means or any other clustering method in the low-dimensional space is also possible [17]. Another approach is the density-based method [25].

Only the first few low-dimensional coordinates are interesting. They contain most of the information about the manifold structure. We use only the dimension corresponding to second eigenvector to determine the anomality of the samples. At 0, this dimension is divided into two clusters. The cluster with more samples is considered normal behavior. Conversely, the points in the other cluster are considered anomalous [22,11,13]. The second eigenvector acts as the separating feature for the two clusters in the low-dimensional space. The second eigenvalue is the solution to the normalized cut problem, which finds small weights between clusters but strong internal ties. This spectral clustering has probabilistic interpretation: grouping happens through similarity of transition probabilities between clusters [22,14].

3 Results

3.1 Data Acquisition

The data is acquired from a large real-life web service. The logfiles contain mostly normal traffic, but they also include anomalities and actual intrusions. The logfiles are from several Apache servers and are stored in *combined log format*. Listing below provides an example of a single logline. It includes information about the user's IP-address, time and timezone, the HTTP request including used resource and parameters, Apache server response code, amount of data sent to the user, the web page that was requested and used browser software.

```
127.0.0.1 - - [01/January/2011:00:00:01 +0300]
"GET /resource?parameter1=value1&parameter2=value2 HTTP/1.1"
200 2680 "http://www.address.com/webpage.html"
"Mozilla/5.0 (SymbianOS/9.2;...)"
```

The access log of a web site contains entries from multiple, distinct URLs. Most of them point to static requests like images, CSS files, etc. We are not focused to find anomalies at those requests because it is not possible to inject code via static requests unless there are major deficiencies in the HTTP server itself. Instead, we are focused in finding anomalies from dynamic requests because those requests are handled by the Web application, which is run behind the HTTP server.

To reach this goal, the access log entries are grouped by the resource URL. That is the part between host name and parameters in the HTTP URL scheme. Those resources containing only HTTP GET requests with no parameters are ignored. Each remaining resource is converted to a separate numerical matrix. In this matrix, a row represents a single access log entry, and a column represents an extracted feature.

Feature extraction is done in two passes. In the first pass the number of features is determined, and in the second pass the resulting matrix is produced. In our study we extracted the number of occurrences of 2-grams produced from HTTP GET parameters. These frequencies are normalized with logarithm in order to scale them. This ensures that the distances between the samples are comparable.

3.2 Data Analysis

To measure the effectiveness of the method the data is labeled so that classification accuracy can be measured. However, this labeling is not used for training the diffusion map. The class labels are not input for the method.

Diffusion map reveals the structure of the data, and all the anomalies are detected. The n-gram features of the data are mapped to a lower dimensions. Figure 1 shows the resulting low-dimensional diffusion space with $\epsilon = 100$. The normal behavior lies in the dense area to the lower right corner. Anomalous points are to the left of 0.

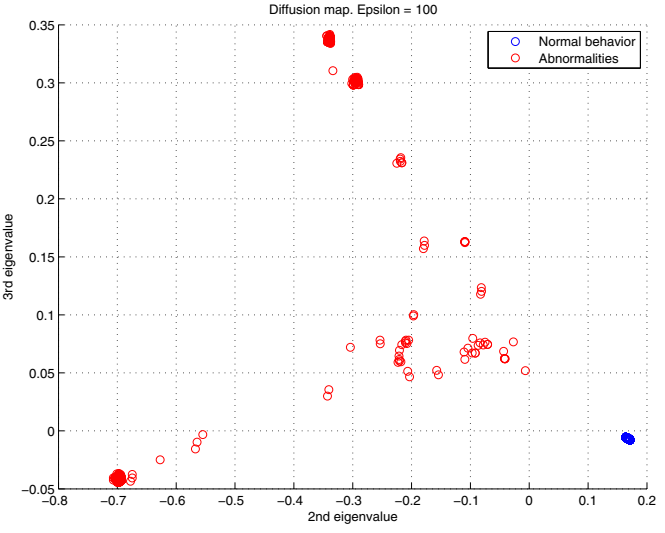

Fig. 1. Two-dimensional diffusion map of the dataset

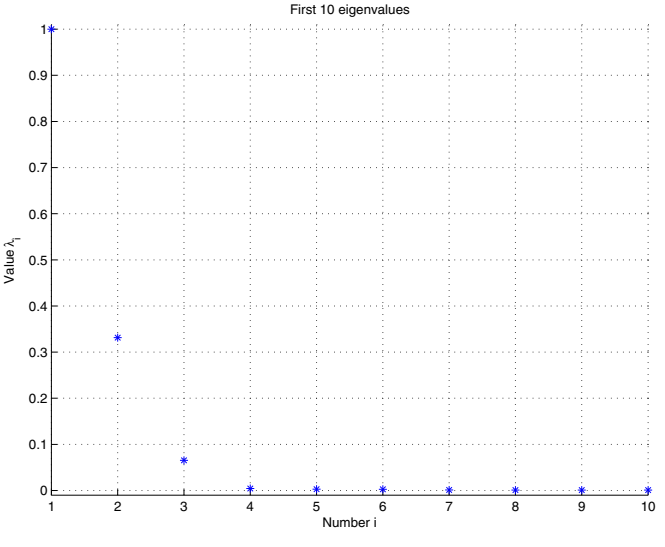

Fig. 2. Eigenvalues of transition matrix with $\epsilon = 100$

Figure 2 shows that the eigenvalues converge rapidly with $\epsilon = 100$. This means that the first few eigenvalues and eigenvectors cover most of the differences observed in the data. The first value is 1 and corresponds to the constant eigenvector that is left out in the analysis. Eigenvalues $\lambda_2 = 0.331$ and $\lambda_3 = 0.065$ cover large portions of the data when compared to the rest that have values below 0.005.

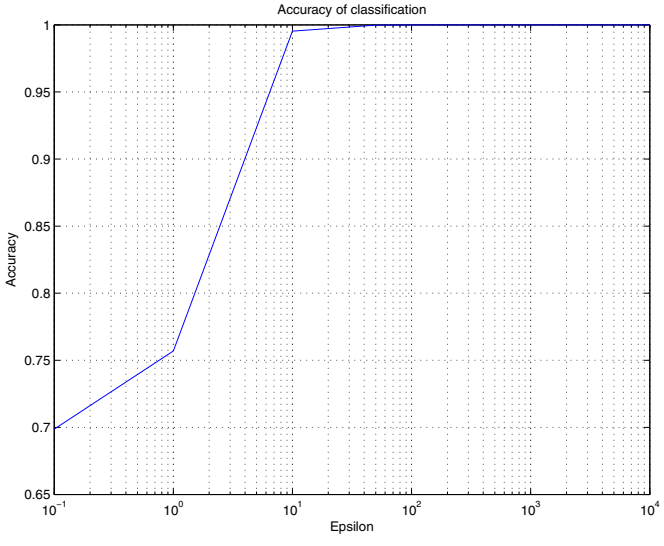

Fig. 3. Accuracy of classification changes when the parameter ϵ is changed

Classification is tested with different values of ϵ, which defines the neighborhood for diffusion map. Accuracy of classification is defined as $accuracy = (tp + tn)/(tp + fp + fn + tn)$. Figure 3 shows how the accuracy of classification changes when ϵ is changed. Higher values of ϵ result in better accuracy. Precision of classification is defined $precision = tp/(tp + fp)$. The precision stays at 1 once any anomalies are detected, which means that all the anomalies detected are real anomalies regardless of the accuracy [9, p. 361].

For comparison, principal component analysis (PCA) is performed on the same normalized feature matrix [9, p. 79]. Results are very similar to the diffusion map approach, because of the simple structure of the feature matrix. Furthermore, PCA reaches the same accuracy and precision as diffusion map. The low-dimensional presentation is also very similar. Figure 4 shows the first two coordinates of PCA.

We also apply support vector machines (SVM) to the same data [9, p. 337–344]. LIBSVM implementation is used [3]. We use one-class SVM with RBF kernel function. A subset of the data is used in the model selection for SVM (500 lines randomly selected). Then the rest of the data is used to test the method.

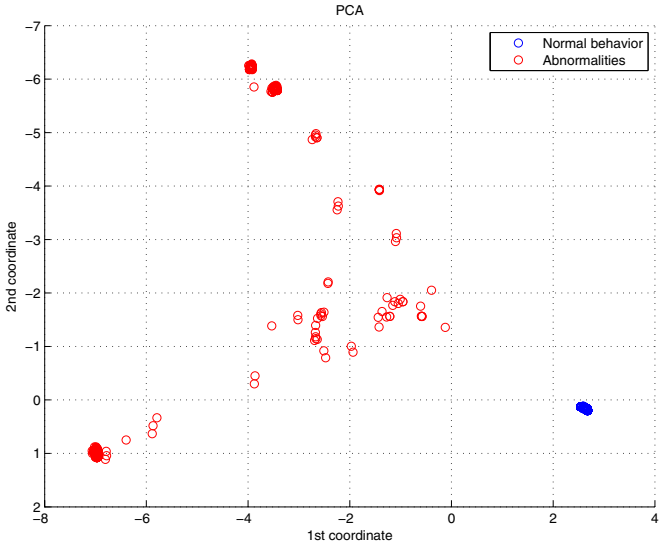

Fig. 4. PCA of the dataset, first two coordinates. The Y-axis of this figure has been reversed for better visual comparison with diffusion map.

The data labels are unknown, so the training data is not "clean" and contains some intrusions as well. It is possible to find the right parameters (ν and γ) for model selection if pre-specified true positive rate is known. The parameters which give a similar cross-validation accuracy can be selected [3]. However, this kind of information is not available. Fully automatic parameter selection for OC-SVM could be achieved by using more complicated methods, such as evolving training model method [24]. In this study the parameter selection is done manually. At best the accuracy is 0.999 and precision 0.998.

4 Conclusion

The goal of this study is to find security attacks from network data. This goal is met since all the known attacks are found. The proposed anomaly detection scheme could be used for query log analysis in real situations. In practice the boundary between normal and anomalous might not be as clear as in this example. However, the relative strangeness of the sample could indicate how severe an alert is.

The diffusion map framework adapts to the log data. It assumes that the data lies on a manifold, and finds a coordinate system that describes the global structure of the data. These coordinates could be used for further analysis of characteristics of anomalous activities.

Because all the methods perform extremely well, the data in question is rather sparse and the discriminating features are quite evident from the feature matrix.

This is the merit of n-gram feature extraction which creates a feature space that separates the normal behavior in a good manner. The features describe the data clearly, and they are easy to process afterwards.

One advantage of the diffusion map methodology is that it has only one meta-parameter, ϵ. It can be estimated with simple interval search. If for some reason the threshold sensitivity needs to be changed, ϵ gives the flexibility to adapt to the global structure. For comparison, the SVM we used has two parameters, ν and γ. Searching the best parameters for the application gets more difficult as the number of parameters increases.

The presented anomaly detection method performs well on real data. As an unsupervised algorithm this approach is well suited to finding previously unknown intrusions. This method could be applied to offline clustering as well as extended to a real-time intrusion detection system.

Acknowledgements. The authors thank Professors Amir Averbuch, Timo Hämäläinen and Tapani Ristaniemi for their continued support. Thanks are extended to Juho Knuuttila and Kristian Siljander for useful discussions and ideas.

References

1. Bengio, Y., Delalleau, O., Roux, N.L., Paiement, J.F., Vincent, P., Ouimet, M.: Spectral Dimensionality Reduction. In: Feature Extraction. Studies in Fuzziness and Soft Computing, pp. 519–550. Springer, Heidelberg (2006)
2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. 41(3), 1–58 (2009)
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
4. Chung, F.R.K.: Spectral Graph Theory, p. 2. AMS Press, Providence (1997)
5. Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proceedings of the National Academy of Sciences of the United States of America 102, 7426 (2005)
6. Coifman, R.R., Lafon, S.: Diffusion maps. Applied and Computational Harmonic Analysis 21(1), 5–30 (2006)
7. Damashek, M.: Gauging similarity with n-grams: Language-independent categorization of text. Science 267(5199), 843 (1995)
8. David, G.: Anomaly Detection and Classification via Diffusion Processes in Hyper-Networks. Ph.D. thesis, Tel-Aviv University (2009)
9. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann, San Francisco (2006)
10. İzmirli, Ö.: Tonal-atonal classification of music audio using diffusion maps. In: 10th International Society for Music Information Retrieval Conference (ISMIR 2009) (2009)
11. Kannan, R., Vempala, S., Vetta, A.: On clusterings: Good, bad and spectral. J. ACM 51, 497–515 (2004)
12. Keller, Y., Coifman, R., Lafon, S., Zucker, S.: Audio-visual group recognition using diffusion maps. IEEE Transactions on Signal Processing 58(1), 403–413 (2010)

13. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17, 395–416 (2007)
14. Meila, M., Shi, J.: Learning segmentation by random walks. In: NIPS, pp. 873–879 (2000)
15. Mukkamala, S., Sung, A.: A comparative study of techniques for intrusion detection (2003)
16. Nadler, B., Lafon, S., Coifman, R., Kevrekidis, I.G.: Diffusion maps – a probabilistic interpretation for spectral embedding and clustering algorithms. In: Barth, T.J., Griebel, M., Keyes, D.E., Nieminen, R.M., Roose, D., Schlick, T., Gorban, A.N., Kégl, B., Wunsch, D.C., Zinovyev, A.Y. (eds.) Principal Manifolds for Data Visualization and Dimension Reduction. Lecture Notes in Computational Science and Engineering, vol. 58, pp. 238–260. Springer, Heidelberg (2008)
17. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems, vol. 14, pp. 849–856. MIT Press, Cambridge (2001)
18. Nguyen-Tuong, A., Guarnieri, S., Greene, D., Shirley, J., Evans, D.: Automatically hardening web applications using precise tainting. In: Sasaki, R., Qing, S., Okamoto, E., Yoshiura, H. (eds.) Security and Privacy in the Age of Ubiquitous Computing. IFIP AICT, vol. 181, pp. 295–307. Springer, Boston (2005)
19. Patcha, A., Park, J.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks 51(12), 3448–3470 (2007)
20. Ramadas, M., Ostermann, S., Tjaden, B.: Detecting anomalous network traffic with self-organizing maps. In: Vigna, G., Krügel, C., Jonsson, E. (eds.) RAID 2003. LNCS, vol. 2820, pp. 36–54. Springer, Heidelberg (2003)
21. Schclar, A., Averbuch, A., Rabin, N., Zheludev, V., Hochman, K.: A diffusion framework for detection of moving vehicles. Digital Signal Processing 20(1), 111–122 (2010)
22. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
23. Tran, Q., Duan, H., Li, X.: One-class support vector machine for anomaly network traffic detection. China Education and Research Network (CERNET) (2004)
24. Tran, Q.A., Zhang, Q., Li, X.: Evolving training model method for one-class svm. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2388–2393 (2003)
25. Turkka, J., Ristaniemi, T., David, G., Averbuch, A.: Anomaly detection framework for tracing problems in radio networks. In: Proc. to ICN 2011 (2011)