Network Intrusion Detection using a Secure Ranking of Hidden Outliers

Marwan Hassani and Thomas Seidl

Data Management and Data Exploration Group
RWTH Aachen University, Germany
{hassani, seidl}@cs.rwth-aachen.de

Abstract. Network intrusion detection has recently attracted a lot of attention in both research and industry of computer network security. By intrusion, attackers try to perform malicious activities inside the network using harmless-looking connections. Network intrusion detection systems try to differentiate these attacks from normal connections by grouping them into families based on similarity. As new forms of intrusions different from the already detected ones are usually seen, *clustering* of network connections is widely used to deal with that. In data mining, clustering aims at dividing objects into different groups (called *clusters*) such that objects in one cluster are similar to each other and dissimilar to objects from other clusters. Some sparse objects deviate from all available clusters and are not dense enough to form a new cluster. These objects are called *outliers* and they usually do not belong to any of available clusters.
For network security, when clustering the connections in the network, many connections could be considered as outliers when compared to the clusters of normal connections but nevertheless they are not real intrusions. Considering every outlier connection as a network intrusion will result in too many false alarms. Previous solutions which handled this problem were not effective enough for detecting intrusions which are hidden in subspaces of the connection data.
We suggest an oultier ranking algorithm for ranking these outlier connections. Using a scoring function, our algorithm gives higher degree of "outlierness" for strongly-deviated outliers hidden in subspaces of the network connection data. We see another challenge when seeking for intrusions in the network. Attackers usually try slight modifications of previously-successful intrusions for producing new attacks. Our novel scoring function carefully gives higher degree of outlierness for outliers found in subspaces which contain known intrusions. Thus we should considerably reduce false alarms since only strongly-deviated outliers and outliers detected in suspected subspaces of the connections will be considered as intrusions.

Keywords: Intrusion detection, Network security, Outlier ranking, Subspace clustering

1 Introduction

Network intrusion detection is an active research topic for both research and industry of computer network security. By intrusion, attackers try to perform malicious activities inside the network using harmless-looking connections. According to [13], known network attacks can be classified into the following main categories:

- **Unauthorized access to the network resources**
  Among the examples are: password cracking, Trojan horses, network packet listening, stealing information, usage of the network resources for private purposes.
- **Unauthorized alteration of the network resources after gaining unauthorized access**
  Like falsification of identity (for example to get system administrator rights) or unauthorized transmission and creation of confidential datasets.
- **Denial of Service (DoS)**
  Attackers in this form compromise the network resources by sending huge amounts of useless information to lock out legal traffic which results in denying services in the network. This can be done by ping floods, send mail floods or TCP requests floods.

Attackers usually start by performing some changes in the legal connections inside the network. This is done by altering some attributes of the connection, aiming at finding a gap in the network. These attempts are innocent-looking connections with only limited odd values of some attributes. Detecting these attempts in the network is an important required feature of any intrusion detection system. Upon a successful intrusion attempt, attackers keep targeting this security gap before it is detected. All further connections similar to that known intrusion are prohibited by the network.

Deciding whether a connection is legal or an intrusion attempt is recently an active research topic in the area of network security. One of the most effective ways for doing that is borrowed from the area of data mining and called *clustering*. In data mining, clustering aims at categorizing objects in the data into different groups *clusters* such that objects in one cluster are similar to each other and dissimilar to objects from other clusters.

This is very convenient for the problem of network intrusion detection problem. Figure 1 illustrates the case of clustering connections with 2 attributes. Similar normal connections are clustered together and known intrusions also form special clusters. New connection which fall in any of the intrusion clusters will result in signaling an alarm with the corresponding intrusion. In this example, attackers attempted to form *intrusion 2* out of normal connections by changing some values of Dim 2.

The unsupervised nature of clustering is another important feature which makes it suitable of the problem of intrusion detection. Attackers usually keep trying new forms of attacks to intrude networks. Upon receiving these new forms

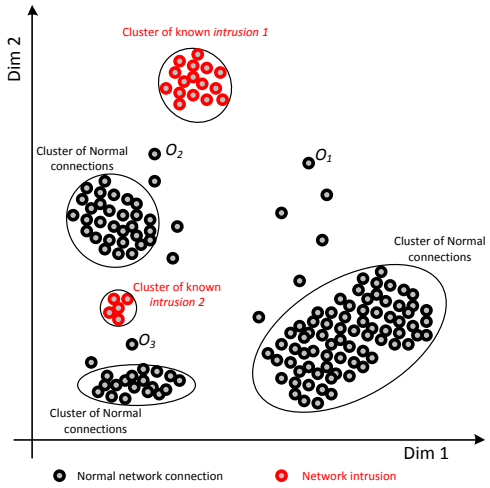

Fig. 1. Clustering of Network Connections into Normal Ones and Intrusions

of attacks, they are usually clustered in new clusters that differ from the available ones.

2 Outlier Connections

Some connections in the network deviate from all available clusters and are unique to form a new cluster. In Figure 1 these connections are the sparse points outside the borders of all clusters. These objects are called *outliers* and usually do not belong to any of available clusters because they are dissimilar to their objects.

2.1 Outliers and Intrusions

The deviation of some attributes from expected patterns could indicate a successful intrusion or intrusion attempt. When considering the calendar and time attributes for example, bank transfers during working days are expected patterns, while they are suspicious when performed on weekends or holidays. Similarly, when considering spatial attributes (like IP Address attributes), using a credit card in different adjacent sites within few hours is a usual pattern, while performing simultaneous or successive transactions from two distant is an indication of intrusion.
For network security, many connections are considered as outliers when compared to the groups of normal connections but nevertheless they are not real intrusions. Considering every outlier connection as a network intrusion will result in too many false alarms. In mobile networks for example, setting a threshold to differentiate between a normal activity and a network intrusion based on the

spatial attributes for example is very tricky. A strict setting of this threshold could result in a huge number of false alarms, while a tolerant one might end with ignoring intrusions.

2.2 Outlier Ranking

Considering every outlier connection as a network intrusion will result in too many false alarms. In order to prune outliers connections to get limited number of intrusions out of them, one suggestion was given in [22]. The authors limited the amount of detected intrusions to a user given number n. This tends to be inconvenient as all found outliers (regardless how far are strong they are deviating from clusters) are reported as intrusions. On the one hand, this does not solve the false alarms problem, and n on the other hand is soon reached resulting in ignoring other (maybe strongly-deviated) outliers which could be real intrusions. Recall Figure 1, outliers O_1, O_2 and O_3 are more suspicious than other outliers since they have the most deviation from available clusters. O_2 and O_3 could be *intrusion 1* and *intrusion 2* respectively, while O_1 could be an attempt of a new intrusion. Thus it is needed to have a mechanism for measuring how "different" some outlier connections are from the usual ones, and thus how suspicious they are. Additionally, one is interested in filtering these outliers to get the one with a high "degree of outlierness". In this paper, we suggest using the ranking of outliers detected in the data.

Section 3 will discuss the meaning of full space and subspace outlier detection and the effectiveness of each for detecting intrusions. Section 4 introduces "SecRank", our novel algorithm for detecting network intrusions using a secure ranking of outliers in subspaces of data. Section 5 gives a preliminary results for proving our concept then we conclude this paper with an outlook in Section 6.

3 Ranking Outliers Hidden in Subspaces of the Data

3.1 Subspace Clustering

In the clustering task, similar objects are automatically grouped together in clusters. Similarity between objects is defined by low distances between them. Objects separated by far distances are dissimilar and thus belong to different clusters. In recent applications like network intrusion detection, objects (connections) are described using many dimensions. For example, each connection in the Network Intrusion Dataset [1] has 42 dimensions. For such kinds of data with higher dimensions, distances grow more and more alike due to an effect termed "curse of dimensionality" [7]. This effect makes traditional clustering algorithm insufficient for getting meaningful clusters. Recent research introduced subspace clustering which aims at locally detecting relevant dimensions per cluster. If a objects in a certain cluster are densely close to each other on some dimension then this dimension is relevant to that cluster, however if they are scattered over

other dimensions then those dimensions are irrelevant to that cluster. Thus, for each cluster, relevant dimensions are locally determined and irrelevant dimensions are ignored. Figure 2 gives an example of subspace clustering of objects

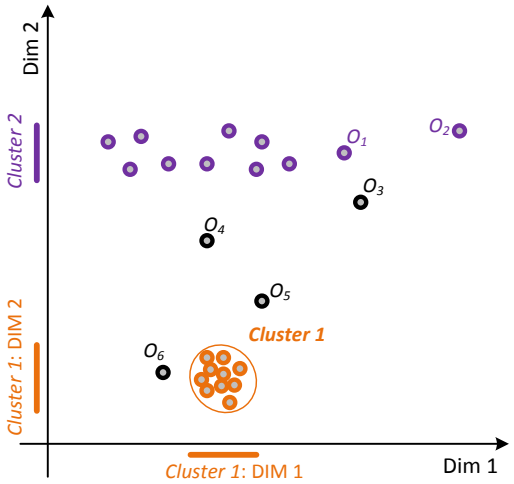

Fig. 2. Example of Subspace Clustering

with two dimensions. *Cluster 1* represents a full space cluster as both dimensions are relevant, while *Cluster 2* identifies a cluster only in the projection to dimension Dim 2. Objects of *Cluster 2* are highly scattered in Dim 1 making Dim 1 irrelevant for *Cluster 2*.

Many algorithms were developed for subspace clustering of high dimensional data. FIRES (FIlter REfinement Subspace clustering) [14] is a general framework for efficient subspace clustering. CLIQUE [4], ENCLUS [9], MAFIA [21], nCluster [18] are grid-based subspace clustering algorithms that use a bottom-up, Apriori style [5] discovery of clusters. Grid-based subspace clustering algorithms are sensitive to the resolution of the grid, and they may miss clusters inadequately oriented or shaped due to the positioning of the grid. SUBCLU [12] is a grid-free approach that can detect subspace clusters with more general orientation and shape than grid-based approaches. DiSH [2] uses a bottom-up search to compute a subspace dimensionality for each data point. These subspace dimensionalities are used to derive a distance between points, which is then used in a top-down computation of clusters. In DUSC [6], a point is initially considered a core point if its density measure is F times larger than the expected value of the density measure under uniform distribution.

3.2 Outliers in Subspace Clustering

As the data dimensionality increases, distances between objects grow more and more alike due to the (curse of dimensionality). This results in having all objects

in the data looking like outliers while *real* outliers are hidden in subspaces of the data. Detecting outliers in subspaces of clusters aims at solving this problem by carefully excluding outliers from subspaces of clusters.
To illustrate the importance of outlier detection in subspaces of the data set for network security reconsider Figure 2. A full space clustering might consider O_3 as a member of *Cluster 2* and O_2 as an outlier. While a subspace clustering will decide that Dim 1 is irrelevant to *Cluster 2* and thus O_2 is a cluster member and O_3 is an outlier of *Cluster 2*. This is important for detecting intrusion attempts. Suppose that *Cluster 2* represents a known attack to the network. This means that all network connections falling inside the projection of *Cluster 2* on Dim 2 are attacks. Attackers usually try to vary the values of irrelevant attributes while using the same parameter values on relevant attributes. New attacks like O_2 will wrongly be considered as outliers in full space clustering, while detecting outliers in subspace clusters will correctly signal an alarm to O_2.

Ranking outliers in subspace of data has the same motivation as ranking them in the full space. In Figure 2 if we suppose that *Cluster 2* represents a safe connection then connection O_4 is deviating more than O_3 from *Cluster 2* and thus much more suspicious. This can only be decided through a ranking of outliers in the subspaces of the data. Ranking of outliers in subspaces of data is actively useful for deciding how suspicious is an object when ignoring irrelevant attributes.

Many algorithms were developed to detect outliers in the full space of the data like LOF [8] and ABOF [15]. For detecting outliers in subspaces of data are: the one presented in [3], OutRank [19], SOD [16] and RPLOF [17]. In SOREX [20], a nice toolkit was presented to rank the outliers detected by previous algorithms.

4 The SecRank Algorithm

Our novel (Secure Ranking) algorithm adopts the solution given in [19] for ranking outlier connections to detect network intrusions. We suggest an oultier ranking algorithm for ranking outlier connections in the network. Using a novel scoring function, our algorithm gives lower scores for strongly-deviated outliers hidden in subspaces of the network connection data.

4.1 Problem Formulation

Let $\mathcal{G}$ represents the group of all clusters in the connection dataset. Let $\mathcal{N} = \{N_1, N_2, \cdots, N_i\}$ where $\mathcal{N} \subset \mathcal{G}$ be the group of all discovered clusters with normal connections in the dataset. Similarly, let $\mathcal{A} = \{A_1, A_2, \cdots, A_j\}$ where $\mathcal{A} \subset \mathcal{G}$ be the group of all discovered clusters with intrusions (attacks) in the dataset.
In this case $\mathcal{G} = \mathcal{N} \cup \mathcal{A} \cup \mathcal{C}$ where $\mathcal{C} = \{C_1, C_2, \cdots, C_k\}$ represents the clusters with unknown category.
Let S denotes the subspace formed by the relative dimensions of a cluster $C \in \mathcal{C}$ which includes an object o. According to subspace clustering, clusters are not

redundant in any higher dimensional subspace, i.e. if C_a is a cluster in subspace S_a then C_a is not a cluster in any $S_b \supset S_a$.
The problem that SecRank is dealing with is to rank all new objects $o \notin \mathcal{N} \cup \mathcal{A}$ which will help in deciding later whether they belong to $\mathcal{N}$ or $\mathcal{A}$.

4.2 The Scoring Functions

SecRank uses two scoring functions. The first is related to the connection itself inside its cluster, the second is related to its relation to neighboring clusters in the same subspace. We assume that the the higher the scoring function of an object, the lower the degree of outlierness of that object and thus the lower degree of suspecting that object.
In the first scoring function, $score_1(o)$ an object o gives a score for each cluster C it belongs to in the subspace S. This function is weighted by the size and the dimensionality of the corresponding cluster.

$$score_1(o) = \sum_{o \in (C,S)} \alpha \cdot \left(\frac{|C|}{|G_{max}|} \right) + (1 - \alpha) \cdot \left(\frac{|S|}{S_{max}} \right) \tag{1}$$

Where:

- G_{max} represents the cluster with the maximum number of members in $\mathcal{G}$.
- S_{max} represents the total number of dimensions in the full space.
- α is a weighing parameter.

Apparently, in $score_1(o)$, the higher the number of dimensions and cluster members of the cluster that o belongs to, the higher the scoring function and thus the lower the degree of outlierness of object o and thus the lower the degree of suspecting o.
The second scoring function $score_2(o)$ expresses the degree of suspecting of o using its distance to neighboring clusters with known types.

$$score_2(o) = \sum_{o \in (C,S)} \beta \cdot \left(\frac{|N_{\epsilon-neighbors}(o)| \cdot |N_S|}{|\mathcal{N}|^2} \right) - (1-\beta) \cdot \left(\frac{|A_{\epsilon-neighbors}(o)| \cdot |A_S|}{|\mathcal{A}|^2} \right) \tag{2}$$

Where:

- $N_{\epsilon-neighbors}(o)$ represents the clusters with known normal connections within the sphere with a radius ϵ surrounding o.
- N_S the total number of clusters with known normal connection in the current subspace S.
- $A_{\epsilon-neighbors}(o)$ represents the clusters with known attacks within the sphere with a radius ϵ surrounding o.
- A_S the total number of clusters with known attacks in the current subspace S.
- β is a weighing parameter.

When a low number of normal connections surrounding and a high number of clusters with attacks, o will have a lower $score_2$ function and thus a higher degree of suspicious.

To sum both internal and neighboring effects we use the $score_{total}(o)$ function for ranking outliers in the dataset as follows:

$$score_{total}(o) = \gamma \cdot score_1(o) + (1 - \gamma) \cdot score_2(o) \tag{3}$$

5 Preliminary Results

To prove the correctness of the concept suggested in our algorithm, we have used the well known Network Intrusion Dataset [1] for comparing a full space outlier ranking algorithm ABOF [15] with a subspace clustering algorithm FIRES [14]. For the evaluation we have used the SOREX [20] tool.

The Network Intrusion Dataset is a real dataset which consists of two weeks of raw TCP dump data for a local area network simulating a true Air Force environment with occasional attacks. Features collected for each connection include the duration of the connection, the number of bytes transmitted from source to destination (and vice versa), the number of failed login attempts, etc. The dataset contains four attacks and the normal connection case. The four attacks included denial of service, unauthorized access from a remote machine (e.g., guessing password), unauthorized access to root, and probing (e.g., port scanning). For our test we have picked a dataset with 100 connections where 10 out of them are attacks. Table 1 depicts the results.

Table 1. Results on the Network Intrusion Dataset (42 Attributes)

Algorithm	False Alarms	True Positives	Precision	$\%_{FalseAlarms}$
ABOF (Full Space)	8	2	0.2	0.08%
FIRES (Subspace)	**3**	**7**	**0.7**	**0.03%**

As shown in Table 1, using the subspace clustering for detecting outliers and ranking them considerably improves the precision comparing with the normal full space clustering method. Additionally using FIRES reduces the percentage of false alarms (i.e. normal connection which were wrongly ranked as attacks) from 0.08% to 0.03%. Taking into consideration the huge 42 parameters included in the ranking process, the results in the above table strongly support our assumption about the effectiveness of using suspace clustering for ranking outliers in order to detect network intrusions.

6 Conclusion and Future Work

We suggested in this paper "SecRank" an oultier ranking algorithm for ranking outlier connections in the network connections data. We aimed at improving the

network security by ranking these connections according to how suspecting they are. Using a scoring function, our algorithm gives higher degree of outliernes for strongly-deviated outliers hidden in subspaces of the network connection data. Attackers usually try slight modifications of previously-successful intrusions for producing new attacks. SecRank uses a novel scoring function that carefully gives higher degree of outlierness for outliers found in subspaces which contain known intrusions. Preliminary results show that we considerably reduce false alarms since only strongly-deviated outliers and outliers detected in suspected subspaces of the connections will be considered as intrusions.

In the future we would like to perform much more extensive evaluation of our method using other real and synthetic datasets. Additionally, we aim at targeting the efficiency issue of the suggested solution. An efficient secure ranking of outliers is an important feature for algorithms applied on resource-limited devices like mobile sensors. Therefore, we would like to extend our previous sensor data clustering algorithm EDISKCO [11] and sensor nodes clustering algorithm ECLUN [10] to include a secure exchange of data and control messages between sensor nodes using the approach suggested in this paper.

Acknowledgments

This research was funded by the cluster of excellence on Ultra-high speed Mobile Information and Communication (UMIC) of the DFG (German Research Foundation grant EXC 89).

References

1. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
2. Achtert, E., Böhm, C., Kriegel, H.-P., Kröger, P., Müller-Gorman, I., Zimek. A.: Detection and visualization of subspace clusters hierarchies. In: DASFAA (2007)
3. Aggarwal, C., Wolf, J., Yu, P., Procopiuc, C., Park, J.: Fast algorithms for projected clustering. In: SIGMOD (1999)
4. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: SIGMOD (1998)
5. Agrawal, R., Srikan, R.: Fast algorithms for mining association rules. In: VLDB (1994)
6. Assent, I., Krieger, R., Müller, E., Seidl, T.: DUSC: Dimensionality unbiased subspace clustering. In: ICDM (2007)
7. Beyer, K., Goldstein J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: ICDT (1999)
8. Breunig, M., Kriegel, H.-P., Ng, R., Sander, J.: LOF: identifying density-based local outliers. In: SIGMOD (2000)
9. Cheng, C. H., Fu, A. W., Zhang, Y.: Entropy-based subspace clustering for mining numerical data. In: KDD (1999)
10. Hassani, M., Müller, E., Spaus, P., Faqolli, A., Palpanas, T., Seidl, T.: Self-Organizing Energy Aware Clustering of Nodes in Sensor Networks using Relevant Attributes. In: KDD (2010)

11. Hassani, M., Müller, E., Seidl, T.: EDISKCO: Energy Efficient Distributed In-Sensor-Network K-center Clustering with Outliers. In: KDD (2009)
12. Kailing, K., Kriegel, H.-P., Kröger, P.: Density-connected subspace clustering for high-dimensional data. In: SDM (2004)
13. Kazienko, P., Dorosz, P.: Intrusion Detection Systems (IDS) Part I - (network intrusions; attack symptoms; IDS tasks; and IDS architecture). Available http://www.windowsecurity.com/search.asp?s=Kazienko
14. Kriegel, H.-P., Kröger, P., Renz, M., Wurst, S.: A generic framework for efficient subspace clustering of high-dimensional data. In: ICDM (2005)
15. Kriegel, H.-P., Schubert, M., Zimek, A.: Angle-based outlier detection in high-dimensional data. In: KDD (2008)
16. Kriegel, H.-P., Schubert, M., Zimek, A., Kröger, P.: Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data. In: PAKDD (2009)
17. Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: KDD (2006)
18. Liu, G., Li, J., Sim, K., Wong, L.: Distance based subspace clustering with flexible dimension partitioning. In: ICDE (2007)
19. Müller, E., Assent, I., Steinhausen, U., Seidl, T.: OutRank: ranking outliers in high dimensional data. In: ICDE (2008)
20. Müller, E., Schiffer, M., Gerwert, P., Hannen, M., Jansen, T., Seidl, T.: SOREX: Subspace Outlier Ranking Exploration Toolkit. In: ECML PKDD (2010)
21. Nagesh, H., Goil, S., Choudhary, A.: Adaptive grids for clustering massive data sets. In: SDM (2001)
22. Singh, G., Masseglia, F., Fiot, C., Marascu, A., Poncelet, P.: Data mining for intrusion detection: from outliers to true intrusions. In: PAKDD (2009)