Discovering Emerging Patterns for Anomaly Detection in Network Connection Data

Michelangelo Ceci, Annalisa Appice, Costantina Caruso, and Donato Malerba

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari, Italy
{ceci,appice,caruso,malerba}@di.uniba.it

Abstract. Most intrusion detection approaches rely on the analysis of the packet logs recording each noticeable event happening in the network system. Network connections are then constructed on the basis of these packet logs. Searching for abnormal connections is where the application of data mining techniques for anomaly detection promise great potential benefits. Anyway, mining packet logs poses additional challenges. In fact, a connection is composed of a sequence of packets, but classical approaches to anomaly detection loose information on the possible relations (e.g., following) between the packets forming one connection. This depends on the fact that the attribute-value data representation adopted by classical anomaly detection methods does not allow either the distinction between connections and packets or the discovery of the interaction between packets in a connection. In order to face this issue, we resort to a Multi-Relational Data Mining approach which makes possible to mine data scattered in multiple relational tables (typically one for each object type). Our goal is to analyse packet logs of consecutive days and discover multivariate relational patterns whose support significantly changes from one day to another. Discovered patterns provide a human-interpretable description of the change in the network connections occurring in consecutive days. Experimental results on real traffic data collected from the firewall logs of our University Department are reported.

1 Introduction

In the last years, an increasing number of organizations are becoming vulnerable to a wide variety of cyber threats, which come from hardware failures, software flaws, tentative probing and malicious attacks. Intrusion detection (ID) is the process of analyzing the events occurring in a network system in order to detect the set of malicious actions that may compromise the integrity, confidentiality, and availability of information resources (security violations) [3]. Traditional methods for intrusion detection are classified into two broad categories: misuse detection and anomaly detection [10]. Misuse detection works by searching for the traces or patterns of well-known attacks while anomaly detection uses a model of normal user or system behavior and flags significant deviations from this model as potentially malicious.

A. An et al. (Eds.): ISMIS 2008, LNAI 4994, pp. 179–188, 2008.

In this paper we are interested in analyzing the firewall logs of a network system in several consecutive days and discovering significant deviations (or changes) in daily network traffic. Hence, we concentrate on detecting anomalies from network connection data. Although anomaly detection has been deeply investigated in the literature [7,8], all proposed methods assume that data are stored in a single table of a relational database (attribute-value representation). This representation allows efficient algorithmic solution but it does not allow to represent the packet-based structure of a single connection.

To overcome limitation posed by single table representation, we exploit findings of research conducted in Multi-Relational Data Mining [6] in order to distinguish between connections (i.e., reference objects of analysis) and packets (i.e., task-relevant objects of analysis) and to mine their interactions: a connection is constructed from one or more packets and the packets are timely related to define a sequence. Coherently with the goals posed by the anomaly discovery task, we propose investigate the opportunity of discovering descriptions of abnormal connections. Such descriptions are in form of relational patterns whose support significantly decreases from one day (target day) to another (background day). Such patterns, known as relational emerging patterns [2], may be employed to capture the "possible" deviation in the traffic network from a day to another: the larger the difference of pattern support, the more interesting the patterns to detect a deviation in network traffic. The interpretation of emerging patterns would add additional depth to the administrators defenses, and allow them to better determine what are the threats against the network they manage.

The paper is organized as follows. In the next Section, we formally present the faced problem. A method to discover relational emerging pattern is described in Section 3. Anomalies detected by mining relational emerging patterns from four successive weeks of firewall logs of a network system are described in Section 4. Lastly, some conclusions are drawn.

2 Problem Definition

Network connection data can be constructed from packet logs recorded by means of packet capturing utilities [4]. The basic premise is that when audit mechanisms are enabled, distinct evidence of anomalies in daily network connections (i.e., differences in connections recorded in consecutive days) will be manifested in the recorded audit data.

Definition 1 (Anomaly detection in Network Connection Data). *Let us consider the connection data constructed from the packet logs* $L_1,\ldots,L_n$ *such that each pair* $\langle L_i, L_{i+1}\rangle$ *describes the network traffic recorded in two consecutive days. Anomaly detection aims at identifying significant deviations (anomalies) in connections recorded one day with respect to connections recorded the day before (or after).*

Such deviations may involve features which describe the connections (e.g., the machine that was contacted, the service that was adopted, the duration of connection) or features which describe one or more packets within each connection

(e.g., number of bytes or duration) or features which describe the interaction between two consecutive packets within the same connection (e.g., distance). Hence, the anomalies of a connection may depend on anomalous values of related features of different object type. Therefore, anomaly detection needs distinguishing between connections and packets and mining their inherent interaction. In fact, connection data are naturally stored in "separate" tables of a relational database D according to a schema S: one table for each object type (connections and packets). The relation between connections and packets is expressed by means of foreign key constraints, while the interaction between packets (e.g., the packet P is consecutive to the packet Q) is stored in a separate table of S. By this mapping of packet logs into a relational database, it is then possible to take into account attributes of related task relevant objects (i.e., packets) when investigating properties of the reference objects (i.e., connections) which are the main subject of analysis. By taking into account the multi-relational structure of data, anomalous connections are described by means of relational patterns. A formal definition of relational pattern is provided in the following.

Definition 2 (Relational pattern). *Let S be a database schema. A relational pattern P over S is a conjunction of predicates:*

$$p_0(t0_1), p_1(t1_1, t1_2), p_2(t2_1, t2_2), \ldots, p_m(tm_1, tm_2)$$

where $p_0(t0_1)$ is the key predicate associated with the target table of the task at hand (i.e., table that contains reference objects) and $\forall i = 1, \ldots, m$ $p_i(ti_1, ti_2)$ is either a structural predicate [1] or a property predicate[2] over S.

Henceforth, we will also use the set notation for relational patterns, that is, a relational pattern is considered a set of atoms.

The change in network traffic can be properly modeled by means of relational emerging patterns [2], that is, multi-variate features whose support significantly decreases from one data class (target class) to another class (background class). The class feature is associated with the reference objects stored in the target table, while explanatory features refer to either the reference objects or the task-relevant objects which are someway related to the reference objects. The structural information required to mine relational emerging patterns can be automatically obtained from the database schema by navigating foreign key constraints.

Definition 3 (Relational Emerging Patterns). *Let D_t and D_b be two instances of a database schema S such that each D_i ($i = t, b$) contains a set of*

[1] A structural predicate is a binary predicate $p(t, s)$ associated with a pair of tables T_i and T_j with T_i and T_j related by a foreign key FK in S. The name p denotes FK, while the term t (s) is a variable that represents the primary key of T_i (T_j).

[2] A property predicate is a binary predicate $p(t, s)$ associated with the attribute ATT of the table T_i. The name p denotes the attribute ATT, the term t is a variable representing the primary key of T_i and s is a constant which represents a value belonging to the range of ATT in T_i.

reference objects labeled with $Y = C_i$ and stored in the target table T of S. Given a minimum growth rate value (minGR) and a minimum support value (minsup), P is a relational emerging pattern to distinguish D_t from D_b if P is a relational pattern with $GR^{D_b \to D_t}(P) > minGR$ and $s_{D_t}(P) > minsup$,

The support $s_{D_i}(P)$ of P on database D_i is computed as follows:

$$s_{D_i}(P) = |O_P|/|O|, \tag{1}$$

where O denotes the set of reference objects stored as tuples of $D_i.T$, while O_P denotes the subset of reference objects in O which are covered by the pattern P. The growth rate of P for distinguishing D_t from D_b is the following:

$$GR^{D_b \to D_t}(P) = s_{D_t}(P)/s_{D_b}(P) \tag{2}$$

As in [5], we assume that $GR(P) = \frac{0}{0} = 0$ and $GR(P) = \frac{\geq 0}{0} = \infty$.

Hence, the problem of discovering relational emerging patterns to detect anomalies in connection data recorded on consecutive days, can be formalized as follows:
Given:

- a sequence $D_1, \ldots, D_n$ of relational databases which are the mapping of the packet logs $L_1, \ldots, L_n$ recorded for n consecutive days into relational databases with a schema S;
- n sets C_i $(i = 1, \ldots, n)$ of connections (reference objects) tagged with class l_i;
- n sets P_i $(i = 1, \ldots, n)$ of packets (task-relevant objects) such that consecutive packets within the same connection are related according to *next* relation;
- a pair of thresholds, that is, the minimum growth rate ($minGR \geq 1$) and the minimum support ($minsup \geq 1$).

Find the set of relational emerging patterns that describe a significant deviation of connections recorded one day with respect to connections recorded the day before (or after).

3 Emerging Pattern Discovery

The relational emerging pattern discovery is performed by exploring level-by-level the lattice of relational patterns ordered according to a generality relation ($\geqslant$) between patterns. Formally, given two patterns $P1$ and $P2$, $P1 \geqslant P2$ denotes that $P1$ ($P2$) is more general (specific) than $P2$ ($P1$). Hence, the search proceeds from the most general pattern and iteratively alternates the candidate generation and candidate evaluation phases (levelwise method). In [2], the authors propose an enhanced version of the levelwise method [9] to discover emerging patterns from data scattered in multiple tables of a relational database. Candidate emerging patterns are searched in the space of linked relational patterns, which is structured according to the θ-subsumption generality order [11].

Definition 4 (Key linked predicate). *Let* $P = p_0(t0_1), p_1(t1_1, t1_2), \ldots, p_m(tm_1, tm_2)$ *be a relational pattern over the database schema* S*. For each* $i = 1, \ldots, m$*, the (structural or property) predicate* $p_i(ti_1, ti_2)$ *is key linked in* P *if*

- $p_i(ti_1, ti_2)$ *is a predicate with* $t0_1 = ti_1$ *or* $t0_1 = ti_2$*, or*
- *there exists a structural predicate* $p_j(tj_1, tj_2)$ *in* P *such that* $p_j(tj_1, tj_2)$ *is key linked in* P *and* $ti_1 = tj_1 \vee ti_2 = tj_1 \vee ti_1 = tj_2 \vee ti_2 = tj_2$*.*

Definition 5 (Linked relational pattern). *Let* S *be a database schema. Then* $P = p_0(t0_1), p_1(t1_1, t1_2), \ldots, p_m(tm_1, tm_2)$ *is a linked relational pattern if* $\forall i = 1 \ldots m$*,* $p_i(ti_1, ti_2)$ *is a predicate which is key linked in* P*.*

Definition 6 (θ-subsumption). *Let* $P1$ *and* $P2$ *be two linked relational patterns on a data schema* S*.* $P1$ *θ-subsumes* $P2$ *if and only if a substitution* θ *exists such that* $P2\ \theta \subseteq P1$*.*

Having introduced θ-subsumption, generality order between linked relational patterns can be formally defined.

Definition 7 (Generality order under θ-subsumption). *Let* $P1$ *and* $P2$ *be two linked relational patterns.* $P1$ *is more general than* $P2$ *under θ-subsumption, denoted as* $P1 \geqslant_\theta P2$*, if and only if* $P2$ *θ-subsumes* $P1$*.*

θ-subsumption defines a quasi-ordering, since it satisfies the reflexivity and transitivity property but not the anti-symmetric property. The quasi-ordered set spanned by $\geqslant_\theta$ can be searched according to a downward refinement operator which computes the set of refinements for a completely linked relational pattern.

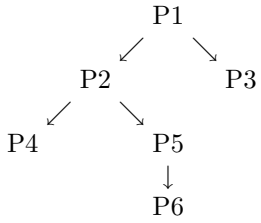
Definition 8 (Downward refinement operator under θ-subsumption). *Let* $\langle G, \geqslant_\theta \rangle$ *be the space of linked relational patterns ordered according to* $\geqslant_\theta$*. A downward refinement operator under θ-subsumption is a function* ρ *such that* $\rho(P) \subseteq \{Q \in G | P \geqslant_\theta Q\}$*.*

The downward refinement operator is a refinement operator under θ-subsumption. In fact, it can be easily proved that $P \geqslant_\theta Q$ for all $Q \in \rho(P)$. This makes possible to perform a levelwise exploration of the lattice of linked relational patterns ordered by θ-subsumption.

Example 1. Let us consider the linked relational patterns:

P1: connection(C).
P2: connection(C),packet(C,P).
P3: connection(C),service(C,'http').
P4: connection(C),packet(C,P), starting_time(P,8).
P5: connection(C), packet(C,P), next(I,P,Q).
P6: connection(C), packet(C,P), next(I,P,Q),distance(I,35).

They are structured in a portion of a lattice ordered by θ-subsumption, that is:



Emerging patterns for distinguishing D_t from D_b are then discovered by generating the pattern space one level at a time starting from the most general emerging pattern (the emerging pattern that contains only the key predicate) and then by applying a breadth-first evaluation in the lattice of linked relational patterns ordered according to $\geqslant_\theta$. Each pattern is evaluated in terms of support and grow-rate value.

In generating each level of lattice, the candidate pattern search space is represented as a set of enumeration trees [13]. The idea is to impose an ordering on atoms such that all patterns in the search space are enumerated. Practically, a node g of a SE-tree is represented as a group comprising: the *head* ($h(g)$) that is the pattern enumerated at g, and the *tail* ($t(g)$) that is the ordered set consisting of the atoms which can potentially be appended to g by ρ in order to form a pattern enumerated by some sub-node of g. A child g_c of g is formed by taking an atom $i \in t(g)$ and appending it to $h(g)$, $t(g_c)$ contains all atoms in $t(g)$ that follows i (see Figure 1). In the case i is structural predicate (i.e., a new relation is introduced in the pattern) $t(g_c)$ contains both atoms in $t(g)$ that follows i and new atoms directly linkable to i according to ρ not yet included in $t(g)$. Given this child expansion policy, without any pruning of nodes or pattern, the SE-tree enumerates all possible patterns and avoid generation and evaluation of candidate equivalent under θ-subsumption to some other candidate.

As pruning criterion, the monotonicity property of the generality order $\geqslant_\theta$ with respect to the support value (i.e., a superset of an infrequent pattern cannot be frequent) [1] can be exploited to avoid generation of infrequent relational patterns. Let P' be a refinement of a pattern P. If P is an infrequent pattern on D_t ($s_{D_t}(P) < minsup$), then P' has a support on D_t that is lower than the user-defined threshold ($minsup$). According to the definition of emerging pattern, P' cannot be an emerging pattern for distinguishing D_t from D_b, hence it is possible

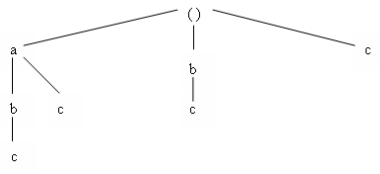


Fig. 1. The enumeration tree over the atoms $A = \{a, b, c\}$ to search the atomsets a, b, c, ab, ac, bc, abc

to avoid the refinement of patterns which are infrequent on D_t. Unluckily, the monotonicity property does not hold for the growth rate: a refinement of an emerging pattern whose growth rate is lower than the threshold $minGR$ may or may not be an emerging pattern.

Finally, as stopping criterion, the number of levels in the lattice to be explored can be limited by the user-defined parameter $MAX_L \geq 1$ which limits the maximum number of predicates within a candidate emerging pattern.

4 Experiments

Experiments concern 28 successive days of firewall logs of our University Department, from June 1st to June 28th, 2004 [4]. Each log is mapped into a relational database (Oracle $10g$). In this study, we consider only the accepted ingoing connections which are reconstructed from single packets. Relational emerging patterns have been discovered with $minsup = 0.1$, $minGR = 1$ and $MAX_L = 5$. Experiments are performed on Intel Centrino Duo - 1.66 GHz CPU RAM 1GB running Windows XP Professional.

4.1 Data Description

A connection is described by the identifier (integer); the protocol (nominal) which has only two values (udp and tcp); the starting time (integer), that is, the starting time of the connection; the destination (nominal), that is, the IP of department public servers; the service (nominal), that is, the requested service (http, ftp, smtp and many other ports); the number of packets (integer), that is, the number of packets transferred within the connection; the average packet time distance (integer), that is, the average distance between packets within the connection; the length (integer), that is, the time length of the connection; the nation code (nominal), that is, the nation the source IP belongs to; the nation time zone (integer), that is, time zone description of the source IP. The source IP is represented by four groups of tree digits and each group is stored in a separate attribute (nominal). Each packet is described by the identifier (integer) and the starting time (number) of the packet within the connection. The interaction between consecutive packets is described by the time distance. Numeric attributes are discretized through an equal-width discretization that partitions the range of values into a fixed number (i.e., 10) of bins.

4.2 Relational Emerging Patterns Evaluation

Relational emerging patterns have been discovered to capture the deviation of the daily connections from the connections recorded on the day after (or before). By comparing each pair of consecutive days, 23,383 emerging patterns are discovered in 10,776 secs. For each day, emerging patterns have been grouped with respect to the background day (the day after or before) and the growth rate value. The number of emerging patterns in each group is reported in Table 1.

Table 1. Number of relational emerging patterns of daily connections from the day after (or before). Emerging patterns are grouped with respect to the grow-rate value.

Day	Grow Rate Range					Day	Grow Rate Range				
	[1,1.5]	]1.5,4]	]4,8]	]8,∞]	∞		[1,1.5]	]1.5,4]	]4,8]	]8,∞]	∞
1 from 2	43	104	81	49	8	2 from 1	1	36	271	20	2
2 from 3	231	15	0	0	22	3 from 2	203	37	0	0	0
3 from 4	11	308	0	0	0	4 from 3	38	63	30	35	0
4 from 5	25	96	1	0	0	5 from 4	68	63	33	26	0
5 from 6	143	85	0	4	0	6 from 5	10	19	0	0	0
6 from 7	23	30	66	51	0	7 from 6	7	113	287	3	0
7 from 8	392	0	0	0	0	8 from 7	62	10	0	0	0
8 from 9	73	24	0	0	0	9 from 8	382	0	0	0	0
9 from 10	272	70	0	0	0	10 from 9	128	7	0	0	22
10 from 11	166	5	0	0	2	11 from 10	184	16	0	0	0
11 from 12	236	113	0	0	29	12 from 11	66	53	4	0	0
12 from 13	258	24	0	0	0	13 from 12	47	34	0	0	0
13 from 14	55	40	4	0	0	14 from 13	186	116	0	0	0
14 from 15	83	34	0	0	0	15 from 14	287	42	0	0	0
15 from 16	147	18	0	0	0	16 from 15	250	1	0	0	0
16 from 17	359	0	0	0	0	17 from 16	79	20	5	6	0
17 from 18	151	157	0	0	0	18 from 17	57	125	108	291	62
18 from 19	67	71	88	275	153	19 from 18	10	333	0	0	0
19 from 20	133	73	4	0	0	20 from 19	326	1	0	0	66
20 from 21	242	93	0	0	3	21 from 20	112	139	2	0	0
21 from 22	2	290	35	0	32	22 from 21	61	56	56	65	36
22 from 23	16	41	0	4	0	23 from 22	134	38	2	19	2
23 from 24	145	63	21	29	0	24 from 23	5	17	2	5	1
24 from 25	48	36	29	70	0	25 from 24	18	183	132	0	0
25 from 26	259	42	0	0	0	26 from 25	84	4	0	0	0
26 from 27	89	39	0	0	0	27 from 26	313	27	0	0	0
27 from 28	95	19	0	0	19	28 from 27	186	124	72	4	0

The emerging patterns whose growth rate is close to 1 ($GR <= 1.5$) capture the profile of connections ingoing the firewall which have approximately the same frequency (support) on consecutive days. Hence, emerging patterns with relatively low value of growth rate ($GR \approx 1$) capture some behavior in daily connection data, and this behavior is maintained on at least two consecutive days. Differently, the larger the growth rate, the more interesting the emerging patterns to detect change in network traffic.

According to such considerations, we can explore the distribution of emerging patterns with respect to the growth rate range and then observe that there is a high number of emerging patterns with growth rate greater than 8 ($GR >= 8$) on June 2nd. These patterns capture a significant deviation in the network traffic on June 2nd from the traffic on June 1st (3rd). In fact, a deeper analysis of these patterns reveals some intersting anomalies. For example, the pattern $P1$:

$P1$: $connection(C), service(C, unknown), packet(C, P), next(I, P, Q),$
$timeDistance(I, [0..100])$

describes the connections C reconstructed from at least two consecutive packets, denoted by P and Q, such that the time distance between P and Q is between 0 and 100 and the service of the connection is unknown. The support of $P1$ on June 2nd is 0.13 with $GR(P1) = 3.42$ from June 1st and $GR(P1) = \infty$ from June 3rd. This means that only few connections satisfying $P1$ incomes firewalls on June 1st, while no connection satisfying $P1$ incomes the firewalls on June 3th. In addition, we verify that $P1$ is unfrequent on all days observed after June 2nd, hence, $P1$ describes the profile of isolated connections (outliers) incoming on June 2nd. Furthermore, the profile of these connections is described by fully exploiting the relational nature of data: $P1$ involves some properties of connections (i.e., service is unknown) and describes the interaction between consecutive packets incoming within the same connection.

Similarly, the analysis of emerging patterns discovered on June 18th reveals some new anomalies. For example, the pattern $P2$:

$P2$: $connection(C), packet(C, P), nationTimeZone(C, 1), time(C, [10h, 12h]),$
$sourceExtIP\_0(C, 193), destination(C, 151)$

has $support = 0.11$ on June 18th with $GR(P2) = \infty$ from June 19th and $GR(P2) = 187.87$ from June 17th. Furthermore, $P2$ is unfrequent ($support < 0.1$) on all observed days after (and before) June 19th (17th). Also in this case, $P2$ identifies some outlier connections incoming only on June 18th. The pattern also includes a human interpretable profile of such connections.

Differently, by analyzing emerging patterns on June 22nd we discover $P3$:

$P3$: $connection(C), packet(C, P),\ service(C, 4671), destination(C, 153),$

such that $support(P3) = 0.69$ on June 22nd with $GR(P3) = \infty$ from June 21st and $GR(P3) = 1.15$ from June 23rd. $P3$ is unfrequent on all observed days before June 21st, while $support(P3) = 0.60$ on June 23rd. This suggests the idea that $P3$ is describing a change in the traffic behavior that is persistent for at least two consecutive days.

5 Conclusions

The problem of detecting anomalies in network connection data can be formalized in the multi-relational framework. In fact, network connections are typically reconstructed from the packet logs daily recorded from firewalls of a network system. Connections and packets are naturally stored in separate tables of a relational database. This allows distinguishing between objects of different types (connections and packets), which are naturally described by a different set of properties, and representing their interactions. Relational emerging patters, that is, multivariate features involving properties of the connection or properties of the packets inside the connection or the interaction between packets within the same connection (a packet P incomes after a packet Q), are then discovered to capture significant change from one day to the day after (or before): the larger the difference of pattern support, the more interesting the patterns to detect a

deviation in the network traffic. Such patterns are employed to detect abnormal activities in the logs without too much human inputs.

As future work, we plan to use emerging patterns to define profiles useful to detect anomalies in run-time. We are interested in extending the emerging pattern discovery in order to discover patterns discriminating the network traffic of one day from the network traffic in a "sequence" of days after (or before). This new kind of emerging pattern will make possible to automatically distinguish outliers and change points [12]. An isolated change not preserved in several days may identify the presence of outlier connections, while a change whose effect is observed for several consecutive days may identify some changing pattern.

Acknowledgments

This work is supported by the Strategic Project: "Telecommunication Facilities and Wireless Sensor Networks in Emergency Management".

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (eds.) International Conference on Management of Data, pp. 207–216 (1993)
2. Appice, A., Ceci, M., Malgieri, C., Malerba, D.: Discovering relational emerging patterns. In: Basili, R., Pazienza, M. (eds.) AI*IA 2007: Artificial Intelligence and Human-Oriented Computing. LNCS (LNAI), pp. 206–217. Springer, Heidelberg (2007)
3. Bace, R.: Intrusion Detection. Macmillan Technical Publishing, Basingstoke (2000)
4. Caruso, C., Malerba, D., Papagni, D.: Learning the daily model of network traffic. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 131–141. Springer, Heidelberg (2005)
5. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: International Conference on Knowledge Discovery and Data Mining, pp. 43–52. ACM Press, New York (1999)
6. Džeroski, S., Lavrač, N.: Relational Data Mining. Springer, Heidelberg (2001)
7. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: Gupta, A., Shmueli, O., Widom, J. (eds.) VLDB, pp. 392–403. Morgan Kaufmann, San Francisco (1998)
8. Mahoney, M.V., Chan, P.K.: Learning nonstationary models of normal network traffic for detecting novel attacks. In: KDD, pp. 376–385. ACM Press, New York (2002)
9. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 1(3), 241–258 (1997)
10. Mounji, A.: Languages and Tools for Rule-Based Distributed Intrusion Detection. PhD thesis, Facultes Universitaires Notre-Dame de la Paix Namur, Belgium (1997)
11. Plotkin, G.D.: A note on inductive generalization. Machine Intelligence 5, 153–163 (1970)
12. Takeuchi, J., Yamanashi, K.: A unifying framework for identifying changing points and outliers. IEEE Transactions on Knowledge and Data Engineering 18(4) (2006)
13. Zhang, X., Dong, G., Ramamohanarao, K.: Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In: Knowledge Discovery and Data Mining, pp. 310–314 (2000)