Relational Frequent Patterns Mining for Novelty Detection from Data Streams

Michelangelo Ceci, Annalisa Appice, Corrado Loglisci, Costantina Caruso,
Fabio Fumarola, Carmine Valente, and Donato Malerba

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

Abstract. We face the problem of novelty detection from stream data, that is, the identification of new or unknown situations in an ordered sequence of objects which arrive on-line, at consecutive time points. We extend previous solutions by considering the case of objects modeled by multiple database relations. Frequent relational patterns are efficiently extracted at each time point, and a time window is used to filter out novelty patterns. An application of the proposed algorithm to the problem of detecting anomalies in network traffic is described and quantitative and qualitative results obtained by analyzing real stream of data collected from the firewall logs are reported.

1 Introduction

A *data stream* is an ordered sequence of data elements which arrive on-line, with no control on their order of arrival, such that once an element has been seen or processed, it cannot be easily retrieved or seen again unless it is explicitly stored in the memory [3]. Data streams are common to a variety of applications in the realm of telecommunications, networking, and real-time monitoring. The huge amount of data generated by these applications demands for the development of specific data mining techniques which can effectively and efficiently discover the hidden, useful knowledge embedded within data streams.

Several data stream mining algorithms have already been proposed in the literature, mainly for clustering, classification, association analysis and time series analysis [9]. Some works focus on the problem of *novelty detection*, i.e., identifying new or unknown situations which were never experienced before. In particular, Spinosa *et al.* [15] propose an incremental learning method to cluster data elements as they arrive, and identify novelties with new clusters formed over time. Ma and Perkins [11] propose to learn a regression function which reflects the normal behavior of a system and define novelties as those data elements which significantly differ from the prediction made by the regression function. Keogh *et al.* [10] take a different perspective on the problem and propose a method which discovers patterns whose frequency deviates from the expected value. A review of novelty detection methods is reported in [13].

P. Perner (Ed.): MLDM 2009, LNAI 5632, pp. 427–439, 2009.

Although all cited works present interesting results, they can only process data elements such that each of them is described by a feature vector. When data elements are complex objects represented by several database relations, these novelty detection algorithms cannot be directly applied, and some kind of data transformation has to be performed which may result in information loss. This observation motivates this work whose main contribution is that of investigating the novelty detection problem in a (multi-)relational setting [8]. In particular, we propose and evaluate a novelty detection method which processes ordered sequences of objects collected at consecutive time points and described by multiple database relations. The method first discovers relational patterns [2] which are frequent at a single time point and then it considers a time window to establish whether the pattern characterizes novelties or not.

The proposed algorithm has been evaluated on data extracted from network connection logs. Indeed, malfunctions and malicious connections can be considered as a form of anomaly in network traffic, and their automatic detection is of great help in daily work of network administrators. The direct representation of all packets of a connection demands for a relational representation which expresses properties of both connections and packets, as well as relationships between connections and packets and relationships between packets.

This relational representation was actually proposed in a previous work [5] which aimed to detect anomalies by comparing the connections ingoing a network firewall one day with the connections ingoing the same firewall another day (not necessarily consecutive). The comparison is based on relational emerging patterns [2] which capture differences between objects (the connections) belonging to different classes (the days) [6]. The main limitation of previous work is the lack of a temporal dimension in the analysis which prevents the investigation of the evolution of pattern support over time. Therefore, an additional contribution of this paper is an improved method for anomaly detection from network connection logs.

The paper is organized as follows. Some definitions relevant for the formalization of the novelty detection problem are introduced in the next section, while a method that solves the problem is described in Section 3. Section 4 introduces the dataset and reports both a quantitative and a qualitative analysis of the results obtained with the proposed method. Lastly, some conclusions are drawn.

2 Problem Definition

In the relational data mining setting, data describing complex objects are scattered over multiple tables of a relational database D. Let S be the schema of D. We assume that S includes the definition of a table T_R, named *target* table, which stores properties (or attributes) of a set R of *reference* (or *target*) objects. These are the main subject of analysis and there is a unit of analysis for each reference object. The support of discovered patterns is computed as the number of reference objects which satisfy the conditions expressed in the pattern. For instance, in the application to novelty detection from network connection logs, the reference objects are the connections, since novelty patterns refer to connections.

We also assume S includes a number of additional (non-target) tables T_{T_i}, such that each T_{T_i} stores attributes of a set R_i of *task-relevant* objects. These contribute to define the units of analysis and are someway related to the reference objects, but they are not the main subject of analysis. In the application to network traffic analysis, packets play the role of task-relevant objects and each unit of analysis includes all packets of a connection.

The "structure" of units of analysis, that is, the relationships between reference and task-relevant objects, is expressed in the schema S by foreign key constraints (FK). Foreign keys make it possible to navigate the data schema and retrieve all the task-relevant objects in D which are related to a reference object.

Definition 1 (Unit of Analysis). *A unit of analysis $D(o)$ consists of the reference object $o \in T_R$ and all task-relevant objects in D that are related to o according to foreign key constraints.*

In this work, units of analysis are associated time points. More precisely, if τ is a sequence of consecutive and discrete time points and $\preceq$ is a total order relation defined on τ, we associate each unit of analysis $D(o_i)$ with a time point $t_i \in \tau$. Therefore, the input data is a series of time-stamped units of analysis, $DS = \{\langle D(o_1), t_1\rangle, \langle D(o_2), t_2\rangle, \ldots, \langle D(o_n), t_n\rangle\}$, where $t_i \preceq t_{i+1}$. It is important to observe that several units of analysis can be associated with the same time point. This allows us to compute the support of a relational pattern at a specific time point.

In order to formalize the concept of relational pattern, we define three types of predicates, namely key, structural and property predicates.

Definition 2 (Key Predicate). *The "key predicate" associated with the target table T_R in S is a unary predicate $p(t)$ such that p denotes the table T_R and the term t is a variable that represents the primary key of T_R.*

Definition 3 (Property Predicate). *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.*

Definition 4 (Structural Predicate). *A structural predicate is a binary predicate $p(t, s)$ associated with a pair of tables T_j and T_i, with T_j and T_i 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_j (T_i).*

A relational pattern is defined as follows:

Definition 5 (Relational Pattern). *A relational pattern P over the schema S is a conjunction of predicates:*

$$p_0(t_0^1), p_1(t_1^1, t_1^2), p_2(t_2^1, t_2^2), \ldots, p_m(t_m^1, t_m^2)$$

where $p_0(t_0^1)$ *is the key predicate associated with the table* T_R *and* $p_i(t_i^1, t_i^2)$, $i = 1, \ldots, m$, *is either a structural predicate or a property predicate over* S.

In this work we also use the set notation of relational patterns, i.e., the conjunction $p_0(t_0^1), p_1(t_1^1, t_1^2), p_2(t_2^1, t_2^2), \ldots, p_m(t_m^1, t_m^2)$ is represented as the set $\{p_0(t_0^1)$, $p_1(t_1^1, t_1^2)$, $p_2(t_2^1, t_2^2), \ldots,$ $p_m(t_m^1, t_m^2)\}$. The two representations are slightly different (neither sequential ordering nor multiple occurrences of atoms are relevant in the set notation), but in this work these differences are not meaningful.

The support of a relational pattern P can be computed at a specific time point t as follows:

$$supp_t(P) = \frac{|\{D(o)|\langle D(o), t\rangle \in DS, \exists\theta : P\theta \subseteq D(o)\}|}{|\{D(o)|\langle D(o), t\rangle \in DS\}|}, \tag{1}$$

where θ is a substitution of variables into constants and $P\theta$ denotes the application of the substitution θ to the pattern P. Therefore, we define a relational pattern P as *frequent* with respect to a minimum support threshold $minSupp$ if a time point $t \in \tau$ exists, such that $supp_t(P_i) \geq minSupp$.

The notion of frequent relational pattern allows us to define a novelty pattern.

Definition 6 (Novelty Pattern). *Let*

- $W(i, w) = \langle t_i, t_{i+1}, \ldots, t_{i+w}\rangle$ *be a time window, i.e., a subsequence of* w *consecutive time points in* τ $(i + w \leq |\tau|)$*;*
- P *be a relational pattern that is frequent in at least one time point* t_i *in* τ *according to a user-defined threshold minSupp, i.e.* $\exists i \in \tau, supp_{t_i}(P) \geq minSupp$*;*
- $\Theta_P : [0, 1] \rightarrow \Psi$ *be a discretization function which associates a support value of* P *in the interval* $[0, 1]$ *with a discrete values* $\psi \in \Psi$.

Then, P *is a* novelty pattern *for the time window* $W(i, w)$ *if and only if:*

$$\Theta(supp_{t_i}(P)) = \ldots = \Theta(supp_{t_{i+w-1}}(P)) \neq \Theta(supp_{t_{i+w}}(P)). \tag{2}$$

Intuitively, a pattern P characterizes novelty in a time window $W(i, w)$ if it has approximately the same support for all time points in $W(i, w)$, except for the last one. Therefore, novelty detection depends on two user-defined parameters: the minimum support ($minSupp$) and the size (w) of the time window. The novelty detection problem can be formalized as follows:

Given:

- a sequence of consecutive and discrete time points τ;
- a series of time-stamped units of analysis $DS = \{\langle D(o_1), t_1\rangle, \langle D(o_2), t_2\rangle, \ldots, \langle D(o_n), t_n\rangle\}$, $t_i \in \tau$, $1 = 1, 2, \ldots, n$, derived from a database D with a target table T_R and m non-target tables T_{T_i};
- a minimum support threshold $minSupp$;
- a time window size w;

Find the sets $NP_{W(i,w)}$ of novelty patterns associated with the time windows $W(i, w)$, $i = 1, 2, |\tau| - w$.

An algorithmic solution to this problem is presented in the next section.

3 Novelty Pattern Discovery

The proposed solution consists of two phases. In the first phase, relational patterns are mined, while in the second phase they are filtered out in order to keep only those which represent a novelty according to Definition 6.

The relational 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 P_1 and P_2, $P_1 \geqslant P_2$ denotes that $P1$ (P_2) is more general (specific) than P_2 (P_1). Hence, the search proceeds from the most general pattern and iteratively alternates the candidate generation and candidate evaluation phases as in the levelwise method [12]. Candidate novelty patterns are searched in the space of linked relational patterns, which is structured according to the θ-subsumption generality order [14].

Definition 7 (Key Linked Predicate). *Let $P = p_0(t_0^1), p_1(t_1^1, t_1^2), \ldots, p_m(t_m^1, t_m^2)$ be a relational pattern over the database schema S. For each $i = 1, \ldots, m$, the (structural or property) predicate $p_i(t_i^1, t_i^2)$ is* key linked *in P if*

- *$p_i(t_i^1, t_i^2)$ is a predicate with $t_0^1 = t_i^1$ or $t_0^1 = t_i^2$, or*
- *there exists a structural predicate $p_j(t_j^1, t_j^2)$ in P such that $p_j(t_j^1, t_j^2)$ is key linked in P and $t_i^1 = t_j^1 \vee t_i^2 = t_j^1 \vee t_i^1 = t_j^2 \vee t_i^2 = t_j^2$.*

Definition 8 (Linked Relational Pattern). *Let S be a database schema. Then $P = p_0(t_0^1), p_1(t_1^1, t_1^2), \ldots, p_m(t_m^1, t_m^2)$ is a linked relational pattern if $\forall i = 1 \ldots m$, $p_i(t_i^1, t_i^2)$ is a predicate which is key linked in P and two structural predicates do not insist on the same foreign key.*

Definition 9 (θ-subsumption). *Let P_1 and P_2 be two linked relational patterns on a data schema S. P_1 θ-subsumes P_2 if and only if a substitution θ exists such that $P_2\theta \subseteq P_1$.*

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

Definition 10 (Generality Order Under θ-subsumption). *Let P_1 and P_2 be two linked relational patterns. P_1 is more general than P_2 under θ-subsumption, denoted as $P_1 \geqslant_\theta P_2$, if and only if P_2 θ-subsumes P_1.*

Example 1. Let us consider the linked relational patterns:
P_1: connection(C).
P_2: connection(C),packet(C,P).
P_3: connection(C),service(C,'http').
P_4: connection(C),packet(C,P), starting_time(P,8).
P_5: connection(C), packet(C,P), next(I,P,Q).
P_6: connection(C), packet(C,P), next(I,P,Q), distance(I,35).
Then it can be proved that the patterns are ordered as follows: $P_1 \geqslant_\theta P_2$, $P_1 \geqslant_\theta P_3$, $P_1 \geqslant_\theta P_4$, $P_1 \geqslant_\theta P_5$, $P_1 \geqslant_\theta P_6$, $P_2 \geqslant_\theta P_4$, $P_2 \geqslant_\theta P_5$, $P_2 \geqslant_\theta P_6$, $P_5 \geqslant_\theta P_6$.

θ-subsumption defines a quasi-ordering, since it satisfies the reflexivity and transitivity property but not the anti-symmetric property. The quasi-ordered set of patterns in example 1 is structured as follows:

$$\begin{tikzcd}
 & & P_1 \arrow[dl] \arrow[dr] & \\
 & P_2 \arrow[dl] \arrow[dr] & & P_3 \\
P_4 & & P_5 \arrow[d] & \\
 & & P_6 &
\end{tikzcd}$$

It can be searched according to a downward refinement operator which computes the set of refinements for a completely linked relational pattern.

Definition 11 (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 $\rho : G \mapsto G$ such that $\rho(P) \subseteq \{Q \in G | P \geqslant_\theta Q\}$.*

In particular, the downward refinement operator ρ' used in this work is defined as follows.

Definition 12 (Downward Refinement Operator). *Let P be a linked relational pattern. Then $\rho'(P) = \{P \cup \{p(t_1, t_2)\} | p(t_1, t_2)$ is a structural or property predicate key linked in $P \cup \{p(t_1, t_2)\}\}$.*

We observe that in order to return a set of linked relational patterns, the predicate $p(t_1, t_2)$ added to a pattern P by ρ' should not insist on the same foreign key of another structural predicate in P. It can be proved that ρ' is a refinement operator under θ-subsumption, i.e., $P \geqslant_\theta Q$ for all $Q \in \rho'(P)$.

The refinement operator ρ' allows for a levelwise exploration of the quasi-ordered set of linked relational patterns. Indeed, the implemented algorithm starts from a set $\wp$ containing only the most general pattern, i.e. the pattern that contains only the key predicate, and then updates $\wp$ by repeatedly applying ρ' to all patterns in $\wp$. For each candidate pattern P, the support $supp_{t_i}(P)$ is computed at each discrete time point t_i.

In generating each level of the quasi-ordered set, the candidate pattern search space is represented as a set of enumeration trees (SE-trees)[17]. 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* ($h(g)$), i.e. the pattern enumerated at g, and the *tail* ($t(g)$) that is the ordered set consisting of all atoms which can be potentially 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 $q \in t(g)$ and appending it to $h(g)$. Therefore, $t(g_c)$ contains all atoms in $t(g)$ that follows q (see Figure 1). In the case q is a structural predicate (i.e., a new relation is introduced in the pattern), $t(g_c)$ contains both

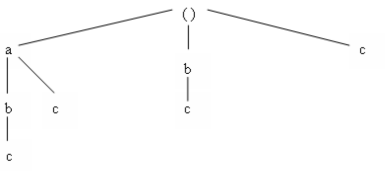

Fig. 1. The enumeration tree over the atoms $A = \{a, b, c\}$ to search the atomsets a, b, c, ab, ac, bc, abc

atoms in $t(g)$ that follows q and new atoms directly linkable to q 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 prevents the 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 ($\forall t_i \in \tau,\ \ supp_{t_i}(P) < minsup$), then P' has a support that is always lower than the user-defined threshold ($minsup$) for each $t_i \in \tau$. According to the definition of novelty pattern, P' cannot be "novel". This means that it is possible to avoid the refinement of patterns which are infrequent.

An additional pruning criterion stops the search when a maximum number of literals ($MaxNumLiterals$) have been added to a novelty pattern, where $MaxNumLiterals$ is a user-defined parameter.

Once patterns are estracted, they are further processed in order to identify novelty patterns according to Definition 6. In this work, function Θ_P is the classical equal-width discretization function [7].

4 Experiments

The method to discover (relational) novelty patterns has been applied to anomaly detection on the network connection logs which are recorded on consecutive days (each day represents a discrete time point). In this context a unit of analysis is described in terms of accepted ingoing connections (reference objects), packets (task-relevant objects) and relations "connections/packets" and "packets/packets". The reason for considering only ingoing connections is that we are ultimately interested in discovering possible attacks to network services, which are assumed to come from outside.

In the experiments reported in this section parameters are set as follows: Ψ includes only five values (i.e., Θ_P discretizes the support into five bins), $minsup = 0.1$ and $MaxNumLiterals = 5$.

4.1 Dataset Description

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 $10g$). 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 unsupervised equal-width discretization that partitions the range of values into a fixed number (i.e., 10) of bins.

The relation “connections/packets” indicates that one packet belongs to a connection, while the relation “packets/packets” represents the temporal distance between two packets within the same connection.

The considered database collects 380,733 distinct connections, 651,037 packets and 270,304 relations “packets/packets” and 651,037 relations “connections/ packets”.

4.2 Analysis of Results

Quantitative results are reported in Table 1, where the number of novelty patters for different time windows is shown. As expected, the number of discovered patterns decreases by increasing the window size ($w = 3, \ldots, 6$), since the patterns found in a time window also belong to the set of patterns extracted for smaller time windows. Interestingly, the number of patterns extracted for each time windows is rather large. This is due to the high number of similar extracted patterns. In fact, in most of cases, the system extracts the patterns that are related each other according to the θ-subsumption generality order (one is the specialization of the other). However, the number of discovered novelty patters significantly decreases for $w = 6$, where the average number of patterns extracted

Table 1. Number of discovered relational novelty patterns. Results are obtained with different $W(i, w)$; $i = 1, \ldots, 28$, while $w = 3, \ldots, 6$

Time-Points	w=3	w=4	w=5	w=6
1				
2				
3	52			
4	333	26		
5	108	78	12	
6	38	5	5	5
7	472	281	13	4
8	7	3	3	1
9	145	2	0	0
10	147	114	59	55
11	84	20	36	4
12	315	226	170	160
13	202	134	110	108
14	164	22	13	13
15	148	81	31	21
16	99	10	1	0
17	56	26	24	24
18	481	371	234	144
19	200	198	198	157
20	369	357	352	352
21	381	49	45	40
22	310	234	100	96
23	107	63	63	59
24	114	32	12	12
25	447	351	39	29
26	79	27	25	19
27	142	34	30	30
28	224	142	34	30
Total No of Novelty Patterns	5224	2886	1609	1363
Average No of Novelty Patterns	200.92	115.44	67.04	59.26

for each time point is less than 60. This makes it possible to manually analyze patterns.

A more interesting analysis can be performed by considering a graphical representation of the same results (see Figure 2), where it is possible to notice the smoothing of peaks in the number_of_novelty_patterns/time_point histogram by increasing the window size. In particular, while for $w = 3$ the cardinality of $NP_{W(i,w)}$ presents a high variance over the different time points, this is somehow mitigated by increasing values of w. This would help the user to identify and analyze critical days, when attacks may have occurred.

Figure 2 shows that there are several critical time points (days) when $w = 3$ and less when $w = 6$. In particular, days where the number of extracted novelty patters is greater than 200 are:

- 4, 7, 12, 13, 18, 19, 20, 21, 22, 25, 28 when $w = 3$,
- 7, 12, 18, 20, 22, 25 when $w = 4$,
- 18, 20 when $w = 5$ and
- 20 when $w = 6$.

According to a manual analysis performed by the network administrator, it results that on June 20th 2004 (Sunday) there were attacks which masked the requested service (or port). In particular, there were 1455 connections (the double of the http connections) characterized by "unknown" service. In contrast, there was no connection with "unknown" service in the previous day.

A qualitative evaluation confirms this analysis. In fact, the following novelty pattern is extracted by the algorithm:

$$P_1 : \mathit{connection}(C), \mathit{packet}(C, P), \mathit{service}(C, \text{“}\mathit{unknown}\text{”}).$$

since its support on June 20th is in the interval $[0.428; 0.535]$ while in the previous days its support is in the interval $[0.0; 0.107]$ (this is a novelty pattern for $W(20, 3)$, $W(20, 4)$, $W(20, 5)$, $W(20, 6)$). P_1 states that a connection C with at least one packet P and with unknown service could be considered as an anomaly.

Another example of extracted novelty pattern is the following:

$$P_2 : \mathit{connection}(C), \mathit{packet}(C, P), \mathit{destination}(C, \text{“}XXX.XXX.XXX.127\text{”}).$$

P_2 is characterized by a support value of 0.119 on the June 18th 2004, while its support is in the interval $[5.89 \cdot 10^{-4}; 0.024]$ in the previous days (this is a novelty pattern for $W(18, 6)$ and, thus, for $W(18, 3)$, $W(18, 4)$, $W(18, 5)$). P_2 states that a connection C with at least one packet P and with destination IP address "$XXX.XXX.XXX.127$"[1] could be considered as an anomaly.

The following pattern is obtained by specializing $P2$:

$$\begin{aligned} P_3 : \; & \mathit{connection}(C), \mathit{packet}(C, P), \mathit{destination}(C, \text{“}XXX.XXX.XXX.127\text{”}), \\ & \mathit{nationcode}(C, \text{“}\mathit{IT}\text{”}). \end{aligned}$$

P_3 is characterized by a support value of 0.115 on the June 18th 2004, while its support is in the interval $[2.48 \cdot 10^{-5}; 0.023]$ in the previous days (this is a novelty pattern for $W(18, 6)$).

An example of pattern which takes into account the relational nature of data is the following:

$$\begin{aligned} P_4 : \; & \mathit{connection}(C), \mathit{packet}(C, P), \mathit{packet\_time}(P, \text{“}[34559; 43199]\text{”}), \\ & \mathit{packet\_to\_packet}(P, Q). \end{aligned}$$

P_4 is characterized by a support value of 0.091 on the June 20th 2004, while its support is in the interval $[0.003; 0.066]$ in the previous days (this is a novelty pattern for $W(20, 6)$). This pattern states that a connection C with at least two

[1] The complete IP address is not specified for privacy reasons.

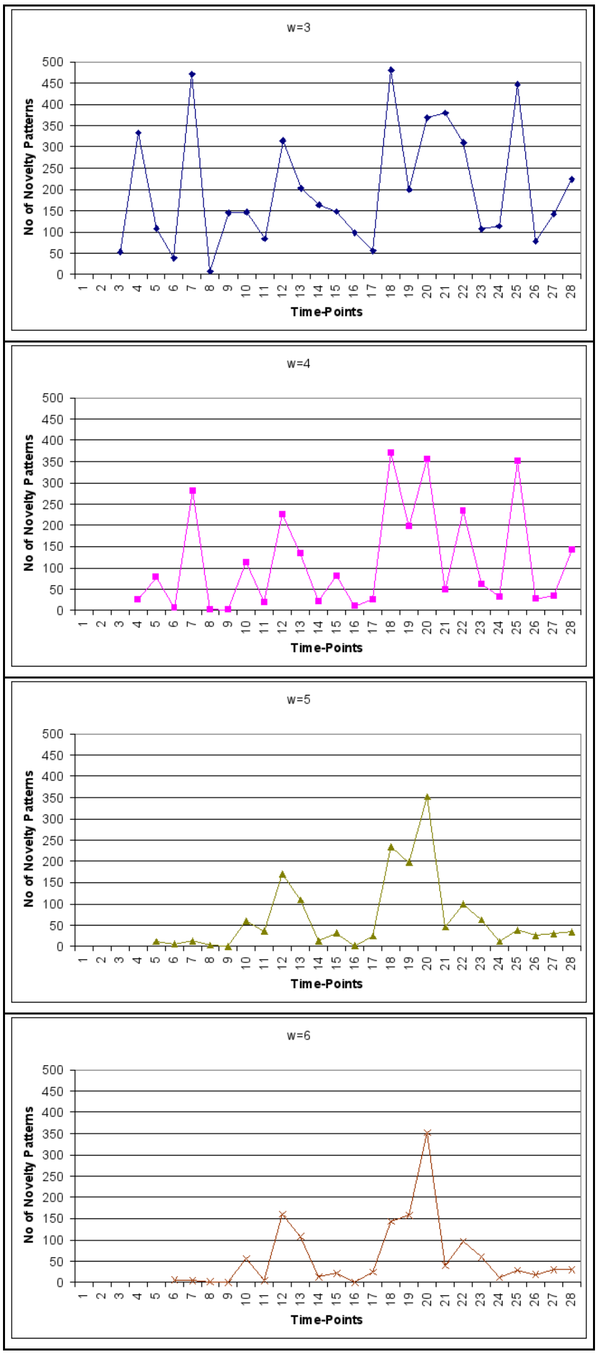

Fig. 2. Distribution of discovered relational novelty patterns. Results are obtained with different $W(i, w); i = 1, \ldots, 28 \quad w = 3, \ldots, 6$

packets P and Q, where P is sent after a relatively high time with respect to the start of the connection (between 34,559 and 43,199 ms), could be considered as an anomaly.

5 Conclusions

In this paper, we face the problem of discovering novelties from data streams and we propose an algorithm whose peculiarity is that it works on data represented in the form of complex objetcs possibly stored in several tables of a relational database. The algorithm uses a time window in order to establish whether the pattern expresses a novelty or not. Discovered novelty patterns are expressed in a first-order logic formalism.

The algorithm is applied to real network traffic data in order to solve a problem of anomaly detection and then support the control activity of a network administrator. Both quantitative (i.e. number of extracted novelty patterns) and qualitative (i.e., novelty patterns themselves) results proved the effectiveness of the proposed approach in detecting possible malicious attacks. By increasing the size of the time window, the number of discovered novelty patterns decreases and, thus, it is possible to simplify the manual analysis of extracted patterns by the expert (network administrator).

As future work, we intend to cluster similar patterns according to syntactic or semantic distance measures [16] in order to further simplify the analysis of extracted novelty patterns by the expert, who can focus his/her attention only on few groups. Moreover, we plan to develop an incremental novelty pattern discovery algorithm in order to face scalability issues.

Acknowledgments

This work is supported by the Strategic Project PS121: "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.T. (eds.) AI*IA 2007. LNCS (LNAI), vol. 4733, pp. 206–217. Springer, Heidelberg (2007)
3. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: PODS 2002: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 1–16. ACM, New York (2002)
4. Caruso, C., Malerba, D., Papagni, D.: Learning the daily model of network traffic. In: Hacid, M.-S., Murray, N.V., Ras, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS, vol. 3488, pp. 131–141. Springer, Heidelberg (2005)

5. Ceci, M., Appice, A., Caruso, C., Malerba, D.: Discovering emerging patterns for anomaly detection in network connection data. In: An, A., Matwin, S., Ras, Z.W., Slezak, D. (eds.) ISMIS 2008. LNCS, vol. 4994, pp. 179–188. Springer, Heidelberg (2008)
6. 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)
7. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Prieditis, A., Russell, S. (eds.) Proceedings of the Twelfth International Conference on Machine Learning, pp. 194–202 (1995)
8. Džeroski, S., Lavrač, N.: Relational Data Mining. Springer, Heidelberg (2001)
9. Gaber, M.M., Zaslavsky, A., Krishnaswamy, S.: Mining data streams: a review. SIGMOD Rec. 34(2), 18–26 (2005)
10. Keogh, E., Lonardi, S., Chiu, B.Y.-C.: Finding surprising patterns in a time series database in linear time and space. In: KDD 2002: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 550–556. ACM, New York (2002)
11. Ma, J., Perkins, S.: Online novelty detection on temporal sequences. In: KDD 2003: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 613–618. ACM, New York (2003)
12. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 1(3), 241–258 (1997)
13. Markou, M., Singh, S.: Novelty detection: a review—part 1: statistical approaches. Signal Process. 83(12), 2481–2497 (2003)
14. Plotkin, G.D.: A note on inductive generalization. Machine Intelligence 5, 153–163 (1970)
15. Spinosa, E.J., de Carvalho, A.P.d.L.F., Gama, J.: Cluster-based novel concept detection in data streams applied to intrusion detection in computer networks. In: SAC 2008: Proceedings of the 2008 ACM symposium on Applied computing, pp. 976–980. ACM, New York (2008)
16. Tsumoto, S., Hirano, S.: Visualization of similarities and dissimilarities in rules using multidimensional scaling. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS, vol. 3488, pp. 38–46. Springer, Heidelberg (2005)
17. Zhang, X., Dong, G., Kotagiri, R.: Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In: Knowledge Discovery and Data Mining, pp. 310–314 (2000)