

# On-Line Update of Situation Assessment Based on Asynchronous Data Streams

Vladimir Gorodetsky, Oleg Karsaev and Vladimir Samoilov

St. Petersburg Institute for Informatics and Automation  
39, 14-th Liniya, St. Petersburg, 199178, Russia  
{gor, ok, samov}@mail.iias.spb.su

**Abstract.** The subject of the paper is multi-agent architecture of and algorithmic basis for on-line situation assessment update based on asynchronous streams of input data received from multiple sources and having finite "life time". A case study from computer network security area that is anomaly detection is used for demonstration.

## 1 Introduction

*Situation* is understood as a complex system constituted of a set of semi-autonomous objects ("*situation objects*") having certain goals and operating in a coordinated mode to achieve a common goal. A "situation object" can be either "physical" (e.g., group of aircrafts involved in a mission), or an "abstract" (e.g., components of software in which traces of attacks are manifested). Situation is characterized by "*state*" taking values from a finite set of labels. *Situation assessment* (SA) is a classification procedure mapping a label to the situation current state based on data received from multiple sources. Many important applications contain SA as a central subtask, e.g., prognosis and handling of natural and man-made emergencies, safeguard and restoration of critical enterprises like nuclear power plants and electrical power grids, prediction of terrorist intents, command and control, computer networks security, etc.

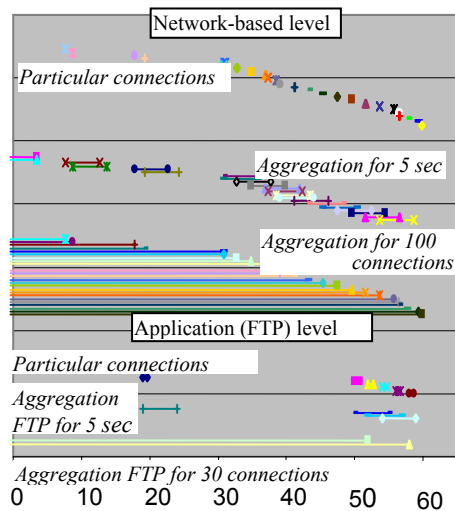
Although currently SA is recognized as the task of great concern, it is weakly researched. The paper [7] provides a thorough analysis of the challenging problems within the scope. In particular, it states that those problems are primarily caused by input data peculiarities. Among them, the following are emphasized: (1) continuous and asynchronous mode of information gathering resulting in the fact that input of SA system is composed of asynchronous data streams. (2) "perishability" of data giving rise to the necessity to update the confidence estimates; (3) incompleteness of input data caused by data unavailability, objects' masking, etc. Unfortunately, the existing research mainly ignores these peculiarities of SA input data model. For example, the recent paper [10] uses different assumptions regarding the missing and uncertain information. It does not consider temporal and on-line assessments update issues. It emphasizes that the main peculiarity of this task is to fill in "a substantial "information gap" between information that is available and information that is required" [10].

Other important issue of SA task is a selection of a strategy of SA status update. Due to dynamic nature of a situation, a practical requirement [7] is to update it "on-line", i.e. each time when a new portion of information specifying situation objects and/or their states arrives. This aspect is explained in Fig.1 and Fig.2 by example from the computer network security scope. Fig.1 presents the data sources used for anomaly detection and emphasizes variety of frequencies of inputs from different data sources. Vertical lines in Fig.2 mark the instants of time when SA has to be updated. Due to variety of frequencies, input data are characterized by different "life time": after elapsing a certain time some data become useless for SA. This practically means that at a time of SA update some previously received data can contain *missing values*.

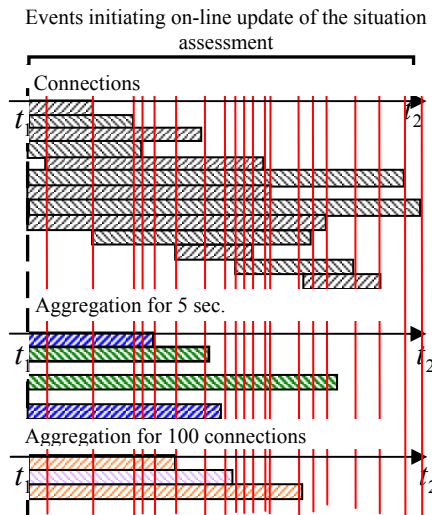
The aforementioned issues i.e. approach to design of SA mechanism based on *asynchronous streams* of input data and to *on-line update* of SA, are the subjects of the paper. In the rest of the paper, section 2 outlines a case study from computer network security scope used for demonstration of the proposed approaches. Section 3 outlines multi-agent architecture of real-time SA system destined for anomaly detection. Section 4 considers learning issue of SA mechanism design if training data are represented as asynchronous data streams with missing values. Conclusion sketches some novel results.

## 2 Anomaly Detection: A Case Study

A case study from computer network security scope aiming at anomaly detection is considered below. Like other SA applications, assessment of the computer network security status makes use of multiple data sources (Fig.1). In the case study, anomaly detection is organized in two steps. *First*, source-based classifiers label security status



**Fig.1.** Dynamic nature and multiplicity of data used for intrusion detection



**Fig.2.** Asynchronous data stream used for on-line situation assessment update

of users' activity either as *Normal* or as *Alert*. These decisions are on-line forwarded to the upper level as asynchronous data streams. The purpose of upper-level classifier is to combine these decisions and produce the final assessment of the situation status. Case study includes dataset used for training and testing of both source-based and meta-level classifiers. This dataset is composed of instances having security status "*Normal*" and those having status "*Abnormal*". The dataset of the class "*Abnormal*" comprises the instances reflecting *four types of attacks that are Probing, Remote to local (R2L); Denial of service (DOS) and User to root (U2R)*. The particular instances of each type attacks included in the case study are *SYN-scan, FTP-crack attack, SYN flood, and PipeUpAdmin* ([2], [6], [8], [9]).

Three primary data sources are considered in the case study: (1) *network-based*, i.e. network traffic; (2) *host-based* corresponding to operating system log and (3) *application-based* corresponding to applications' logs, particularly, FTP server log. In turn, four secondary data streams are generated on the basis of each aforementioned primary source. These data streams are of the same structure for each primary source. Consider them by example of secondary data of network-based level:

1. *Vectors of binary sequences specifying* stream of headers of *IP* packets within a connection. Its components are composed of certain packet header parameters.
2. *Statistical attributes of particular connections (sessions of users) manifested in input traffic*. As features the duration, status, total number of connection packets and also six additional attributes specifying other statistics of connections are used.
3. *Statistical attributes of traffic during the short time (5 sec) intervals*. This data source is presented by four features specifying integral characteristics of input traffic that are total numbers of connections and services of different types during last 5 sec.
4. *Statistical attributes of traffic for long time intervals*. These data are composed of the same statistics as previous ones but averaged over chosen number of connections.

The datasets specifying the above case study were produced from *Tcpdump/Windump* data processed by *TCPtrace* utility and also by other ad-hoc programs. <sup>1</sup>

### 3 Multi-agent Architecture of Situation Assessment System

Thus, SA is a multi-level procedure of distributed data processing producing decisions based on several asynchronous data streams (see Fig.1 and 2). This peculiarity causes a number of distinctive properties of SA systems as compared with conventional classification systems. Let us analyze them to understand how these distinctive properties affect the SA system functionalities and architecture.

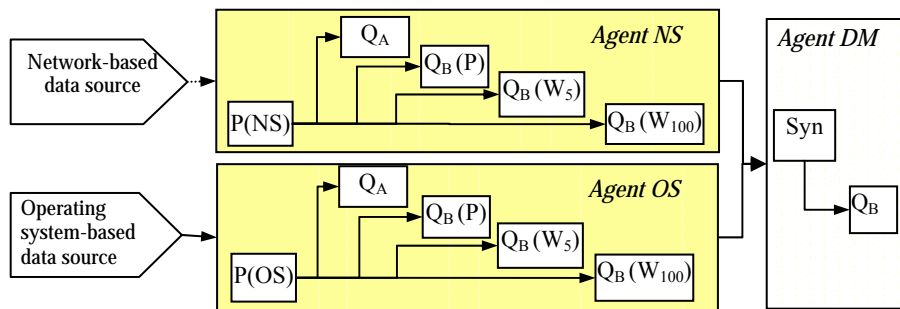
As a rule, many different data processing procedures participate in SA producing. Among them there exist such procedures whose input is formed from different data sources and, therefore, asynchronously. Due to the fact that different components of data have different "life time", at the time when a new portion of data is received by a procedure, certain data received earlier from other sources can appear to be "too old" and thus useless for making decisions. Therefore, it is necessary to have a procedure detecting such "too old" data. We call this procedure a "*data synchronization*" one.

---

<sup>1</sup> This datasets were designed and generated by Prof. I. Kotenko and his student M. Stepashkin.

All the other procedures of the SA system are ordinal for any classification systems. The latter, as a rule are divided in two groups: (1) feature space transformation (examples are computation of the truth value of rules in rule-based systems, transformation of feature space used in Support Vector Machine approach [11], computation of statistics in statistical approaches to classification, etc.) and (2) decision making.

Thus, the generalized architecture of multi-agent SA system has to support the structure of multi-level decision making in which the major agents' roles are R1: "data synchronization", R2: "feature space transformation" and R3: "decision making". Fig.3 demonstrates such architecture by example corresponding to the



**Fig.3.** Multi-agent anomaly detection system architecture

anomaly detection case study. Let us note that Fig.3 gives the fragment of the case study of the SA system. In this figure, the component corresponding to processing and combining data of the application level are omitted, because it has completely the same architecture as two other given in this figure.

The developed SA system architecture consists of agents of three classes denoted by abbreviators *Agent NS*, *Agent OS* and *Agent DM*. Two of them, *Agent NS* and *Agent OS* execute two identical roles: *R2* and *R3*, and handle the data of network-based and operating system levels respectively. *Agent DM* plays three roles that are *R1*, *R2* and *R3*, and their joint performance allows to make decisions on the basis of asynchronous streams of decisions produced by *Agent NS*, *Agent OS* and also *Agent AP* ("Application level agent"). Let us note that the latter is not shown in Fig.3.

#### 4 Learning of On-line Situation Assessment Update

As it was shown, on-line update of the situation assessment based on asynchronous data streams is reduced to a specific classification task with missing values. Respectively, learning of meta-level classifier of SA system is a task of mining of binary data with missing values.

Mining of data with missing values is a special problem of data mining, investigated at least during the last two decades. As a rule, these investigations are mainly concentrated on the methods aiming at a reasonable assignment ("imputation") of the missing values based on a statistical approach or on other ideas. Unfortunately,

this approach is not applicable to the SA task due to significantly different frequencies of arrival of data from different sources. Recently an algorithm for direct mining of data with missing values that does not use an imputation of missing values was developed by the authors [3]. This approach was applied to training of *Agent DM* (Fig.3) performing on-line SA update and exposed good properties. Let us outline the main idea of this approach.

Input of *Agent DM* is composed of binary data taking values from the set  $\{Alert, Normal\}$ , which can be coded as  $\{1, 0\}$  respectively. The idea of the aforementioned approach is conceptually simple and is as follows. If we assigned the missing values of training dataset in an arbitrary way we would be able to extract the set of *maximally general rules, MGRs* [5] using an existing technique like AQ [5], or RIPPER [1], or GK2 [4], etc. Different assignments would lead to different *MGR* sets. Let us denote an *MGR* set for an arbitrary assignment of missing values as  $\mathbb{R}_*$ .

It was discovered that there exist two special sets of *MGR*, which can serve as *low* and *upper bounds* for all possible sets of rules corresponding to any potentially possible assignments of missing values:

$$\mathbb{R}_{low} \subseteq \mathbb{R}_* \subseteq \mathbb{R}_{upper} \quad (1)$$

where  $\mathbb{R}_{low}$  and  $\mathbb{R}_{upper}$  are the *low* and the *upper bounds* respectively for all the sets of *MGR*. Informally, it could be said that the set  $\mathbb{R}_{upper}$  corresponds to "*optimistic*" and  $\mathbb{R}_{low}$  – to "*pessimistic*" assignments of the missing values. Let us briefly explain how these bounds are built [3].

Let us denote an arbitrary *i*-th instance of the training dataset as  $t(i)$ . Let  $k$  be the index of the chosen *seed* [5],  $I_k^+$  be the set of indexes of assigned (not missing) attributes of *seed*. While searching for *MGR* corresponding to the *chosen seed*  $t(k)$ , we will ignore all the columns of training dataset, whose indexes do not belong to the set  $I_k^+$ . Let us denote the index set of missing values in an arbitrary negative example  $t(l)$  by  $I_{l,k}^-$  and the index set of missing values in a positive example  $t(r)$ ,  $r \neq k$ , by  $I_{r,k}^+$ . Let us further consider *two variants of assignment of missing values* in the negative and positive examples:

$$t_i^l = -t_i^k, \text{ if } i \in I_{l,k}^-, l \in \mathbf{NE}; \text{ and } t_i^r = t_i^k, \text{ if } i \in I_{r,k}^+, r \in \mathbf{PE}, \quad (2)$$

$$t_i^l = t_i^k, \text{ if } i \in I_{l,k}^-, l \in \mathbf{NE}; \text{ and } t_i^r = -t_i^k, \text{ if } i \in I_{r,k}^+, r \in \mathbf{PE}. \quad (3)$$

The first assignment, (2), is such that it *maximally increases the distinctions* between the *seed* and negative examples, and *maximally increases the similarities* between the *seed* and other positive examples. On the contrary, the second one (3), *maximally increases the similarities* between the *seed* and negative examples, and *maximally increases the distinctions* between the *seed* and other positive examples. Intuitively, the first assignment can be reasonably called *optimistic*, since it cannot decrease both the generalization level and coverage factor of any rule of *MGR* extracted from the complete dataset. In the second assignment, which can be

reasonably called "*pessimistic*", both the generality and coverage factors of rules extracted from complete dataset cannot be increased. The *Theorem* strictly formulates the above facts and shows how to find the upper  $\mathbb{R}_{upper}$  and low  $\mathbb{R}_{low}$  bounds of *MGR*.

*Theorem* [3]. Let us assume that seed  $t(k)$  does not contain missing values,  $\mathbb{R}_*$  be the set of all *MGRs* for an arbitrary assignment of missing values in negative and positive examples, whose indexes  $i \in I_{l,k}^-$  for  $l \in \mathbf{NE}$ , and  $i \in I_{r,k}^+$  for  $r \in \mathbf{PE}$  respectively;  $\mathbb{R}_{upper}$  be the set of all *MGRs* corresponding to the assignments (2) of positive and negative examples, and  $\mathbb{R}_{low}$  be the set of all *MGRs* corresponding to the assignments (3). Then  $\mathbb{R}_{low} \subseteq \mathbb{R}_* \subseteq \mathbb{R}_{upper}$ , where  $\subseteq$  is deducibility relationship.

This *Theorem* provides general framework for mining of data with missing values. It indicates the set of rules containing the set of *MGR* under search, but it does not show how to select rules from  $\mathbb{R}_{low}$  and  $\mathbb{R}_{upper}$  to be further used for classification.

However, the practice proved that the target rule set can be selected from  $\mathbb{R}_{upper}$  through an algorithm based on testing procedure. Let us explain this point, while assuming that the alternative classes are denoted as  $Q$  and  $\bar{Q}$ . Conceptually, the core of this algorithm consists of the following steps applied to each *seed*:

1. Assign the missing values of training dataset "optimistically" and mine the rule set  $\mathbb{R}_{upper}$  for classes  $Q$  and  $\bar{Q}$ ,  $R_{upper}(Q)$  and  $R_{upper}(\bar{Q})$ .
3. Assess the quality of the extracted rules of the sets  $R_{upper}(Q)$ ,  $R_{upper}(\bar{Q})$ , based on testing dataset and using certain evaluation criteria (coverage, false positives, etc.).
4. Based on the values of the above evaluation criteria, select the best rules from the sets for use in classification mechanism.
5. Design classification mechanism and assess its performance quality.

The other procedures are the same as for data without missing values (e.g., see [5]).

An extended experiment simulating training and testing of on-line anomaly detection system built based on the developed case study allows to optimistically evaluate the proposed approach to direct mining of rules from the datasets with missing values. Indeed, the anomaly detection system trained according to the aforementioned algorithm shows on testing dataset the estimated probability of the correct classification close to 0,99. The experimental results also allowed extending the above optimism with regard to other applied system destined for on-line SA update based on asynchronous data streams arriving from multiple data sources.

## 5 Conclusion

The paper discusses the basic ideas of an approach to learning of on-line situation assessment update. Specifically, it analyses the peculiarities of this class of

applications resulting in exposure of several aspects in which it differs from other classification tasks, in particular, the necessity to use multi-level structure of decision making and to make decisions on the basis of asynchronous data streams. The novel results presented in the paper are as follows:

(1) New sound approach to direct mining of data with missing values based on computation of upper and low bounds of the sets of *maximally general rules* that can be extracted from arbitrary assigned training data with missing values.

(2) Two level multi-agent architecture for situation assessment systems making decisions based on asynchronous data streams arriving from multiple sources.

The main paper results were used in design and implementation of a software prototype of multi-agent anomaly detection system operating on the basis of multiple data sources. The future research will aim at further validation of the paper results via design, and implementation of multi-agent software prototypes for other application.

## Acknowledgement

We wish to thank European Office of Aerospace Research and Development of the USAF, Office of Naval research Global (USA) and Russian Foundation for Basic Research (grant # 04-01-00494) for support.

## References

1. Cohen, W.: Fast efficient rule induction. Machine Learning: The 12<sup>th</sup> International Conference, CA, Morgan Kaufmann (1995)
2. Cole, E.: Hackers Beware. New Riders Publishing (2002)
3. Gorodetsky, V., Karsaev, O.: Mining of Data with Missing Values: A Lattice-based Approach. International Workshop on the Foundation of Data Mining and Discovery, Japan (2002) 151–156
4. Gorodetsky, V., Karsaev, O.: Algorithm of Rule Extraction from Learning Data. Proceedings of the 8-th International Conference "Expert Systems & Artificial Intelligence" (EXPERTSYS-96) (1996) 133-138
5. Michalski, R.: A Theory and Methodology of Inductive Learning. *Machine Learning*, vol.1, Eds. J.G.Carbonel, R.S.Michalski and T.M.Mitchel, Tigoda, Palo Alto (1983) 83-134,.
6. Northcut, t S., McLachlan, D., Novak, J.: Network Intrusion Detection: An Analyst's Handbook. New Riders Publishing (2000)
7. Salerno, J., Hinman, M., Boulware, D.: Building a Framework for Situation Assessment. 7th International Conference on Information Fusion. Stockholm, Sweden (2004) (To appear)
8. Scambray, J., McClure, S.: Hacking Exposed Windows 2000: Network Security Secrets. McGraw-Hill (2001)
9. Scambray, J., McClure, S., Kurtz, G.: Hacking Exposed. McGraw-Hill (2000)
10. Than, C.Ly, Greenhill, S, Venkatesh, S., Pearce, A.: Multiple Hypotheses Situation Assessment. Proceedings of The 6th International Conference on Information Fusion. Australia (2004) 972-978
11. Vapnik, V.: Statistical Learning Theory. J.Willey and Sons, New York (1998)