
Multi-agent and Data Mining Technologies for Situation Assessment in Security-related Applications

Vladimir Gorodetsky, Oleg Karsaev and Vladimir Samoilov

SPIIRAS, 39, 14-th Liniya, St. Petersburg, 199178, Russia
{gor,ok,samovl}@mail.iias.spb.su

The paper considers one of the topmost security related problems that is situation assessment. Specific classification and data mining issues associated with this task and methods of their solution are the subjects of the paper. In particular, the paper discusses situation assessment data model specifying situation, approach to learning of situation assessment, generic architecture of multi-agent situation assessment systems and software engineering issues. Detection of abnormal use of computer network is a case study used for demonstration of the main research results.

1 Introduction

Security-related problems, which recently became of great concern for human society, constitute a new class of applications within information technology scope. Among such applications, the most important ones are those associated with security of critical state infrastructures including computer networks and information systems assurance, safeguard and restoration of critical enterprises like nuclear power plants, electrical power grids, etc. Other important class of such applications covers assessment of threat and prognosis of development of situations associated with large scale natural and man-made disasters and mitigation of their negative impact on the environment. Very specific class of security-related applications is caused by the necessity to predict terrorist intents and counteract against terrorist attacks. The list of such security related applications of topmost concerns can be continued.

From information technology point of view, security-related applications possess a number of common very specific properties making extremely difficult the development of the corresponding decision making and control systems. Among such properties, the most specific ones are multiplicity of distributed data sources, heterogeneity, incompleteness, uncertainty and temporal nature of input data to be fused for decision making, large scale and

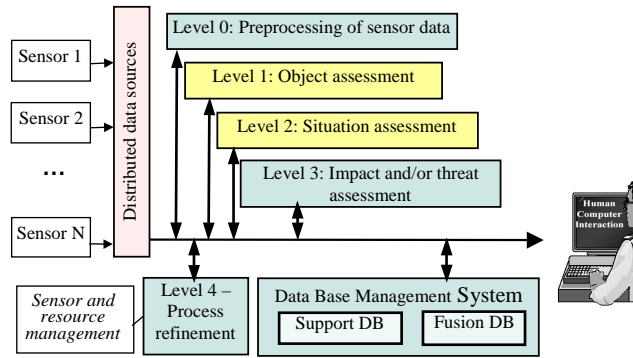


Fig. 1. JDL model of data and information fusion [Salerno-01]

distributed nature of decision making problem, etc. The above impulses new research in the area of distributed intelligent information systems ([11], [12], [1], [8], [15], etc.) whose main objective is a so-called situational awareness task that is understood as the in-depth comprehension, prediction, and management of what is going within the system and environment of interest.

The experience accumulated with regard to situational awareness problem allowed creating a general model of data processing within respective applications, so-called JDL model¹ [14]. It considers hierarchy of tasks associated with the situation awareness-related applications (Fig. 1). In the commonly accepted the situational awareness is a *situation-centric* problem, whose most significant subtasks are *Object Assessment* often referred to as Data Fusion and *Situation Assessment* referred to as Information Fusion. Both these tasks are currently the subjects of intensive research ([1], [8], [15], etc.).

Certain important aspects of the situation assessment task constitute the main subjects of this paper. On the one hand, distributed nature of situation assessment system input data necessitates the use of distributed architecture. In this respect the paper takes advantage of multi-agent architecture for systems in question. On the other hand, incomplete and temporal nature of input data makes the decision making problem rather specific, and this issue is also a subject of the paper. It is further shown that due to the aforementioned properties of input data the classification has to be produced based on multiple asynchronous data streams. Unfortunately both learning of classification and classification itself for such kind of data are poorly investigated. The paper proposes an approach that allows coping with the respective learning and classification problems.

The subsequent part of the paper is organized as follows. Section 2 introduces the basic notions associated with the situation assessment task and

¹JDL model was developed by Joint Directories Research Laboratories of the US Air Force within the framework of Information Fusion Initiative.

outlines specific features of input data model used for situation assessment. Section 3 presents briefly the developed and completely implemented methodology of situation assessment based on information fusion and outlines the multi-agent architecture of a particular security-related application that is a detection of the security status of computer networks. Section 4 describes an approach intending training of classifiers destined for on-line situation assessment update based on asynchronous inputs from multiple sources. Section 5 considers implementation issue of multi-agent situation assessment systems and demonstrate this aspect by example of such a system developed by the authors. Conclusion summarizes the research results and outlines future works.

2 On-line Situation Assessment Update: Peculiarities of Input Data

Situation assessment is the topmost task in the security-related scope. *Situation* is a characteristic of a system constituted by semi-autonomous objects (*situation objects*) having particular goals and operating in a coordinated mode to achieve certain goal of the system on the whole. Situation object can be physical (e.g., technical means participating in a rescue operation) or abstract (e.g., components of software where traces of attack against computer are manifested). Situation and objects are characterized by their “states” taking values from finite sets of labels. *Situation assessment* task (or rather “situation state assessment” task) is a classification task aiming to determine its current state; its essence is that at each given time instant a label is mapped to situation. Situation and objects states are of dynamic nature and therefore situation assessment is a real time task.

Situation related information arrives continuously from multiple distributed sensors. As a rule, the outputs of these sensors come into situation assessment system with different frequencies and in irregular mode constituting jointly *asynchronous input data streams* that have to be processed by situation assessment system in order to online update the current situation state.

The below given example from computer network security demonstrates peculiarities of situation assessment system input; it considers the anomaly detection task. It is assumed that security status of a computer network can take values from binary set {“*Normal*”, “*Abnormal*”}. It is also assumed for simplicity that four data sources resulting from preprocessing of network traffic constitute input of the security status assessment system²; they are:

(1) *connection — related vectors of binary sequences* specifying six-component stream of IP packets headers;

(2) *statistical attributes of connections* manifested in traffic (like duration, status, total number of connection packets, etc.);

²The whole case study from intrusion detection scope that is used for validation of the situation assessment technology under development includes multiple data sources of traffic level, operating system level and application level.

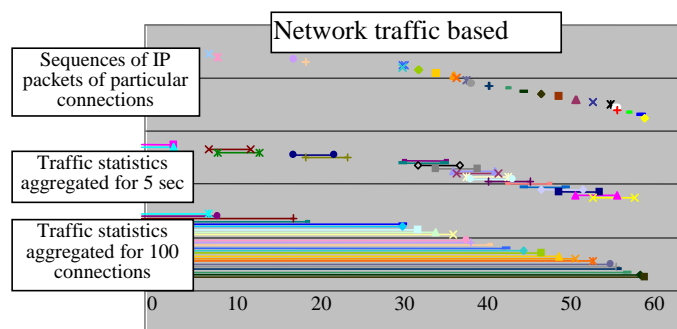


Fig. 2. Multiplicity of input data streams used for anomaly detection based on data of network traffic level

(3) *statistical attributes of traffic during the short time (5 sec) intervals* presented by four features specifying integral characteristics of input traffic like total numbers of connections and services of various types for last 5 sec; and

(4) *statistical attributes of traffic for long time intervals* composed of the same statistics as previous ones but averaged over chosen number of connections.

Fig. 1 illustrates part of these data streams graphically. The datasets of the above kinds used below for demonstration of the properties of the developed approach to mining of asynchronous data streams were resulted from *Tcpdump/WinDump* data processed by *TCPTrace* utility and also by some other ad-hoc developed programs.

Sensor data are collected continuously, and one of their peculiarities is that they are time-stamped and particular data streams input into situation assessment system with different frequencies and possess finite values of “life times”, that can considerably vary for data of different streams. Finiteness of life time results in the fact that after elapsing a definite time a part of data becomes useless for situation assessment. Therefore at a time of situation assessment update some attributes can be not assigned a value and, thus, input data vector to be used for situation assessment update has *missing values*. Fig. 2 demonstrates this fact. Indeed, let us assume that new data (“events”) arrive at the times T_1, T_2, T_3 and T_4 , and according to the necessity to update situation assessment in real time mode, at the same times T_1, T_2, T_3 and T_4 situation assessment system has to make decision about current computer network security status. Decision at the time T_1 is initiated by arrival of data denoted as Z_1 about the most recent connection completed. At that moment, *life times* of the most recently received data corresponding to the traffic statistics aggregated for 5 sec, Z_2 , and corresponding to the traffic statistics aggregated for 100 connections, Z_3 , are not yet elapsed and that is

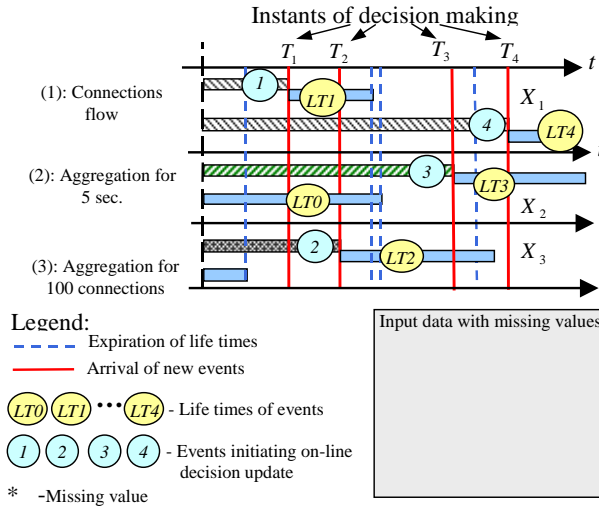


Fig. 3. Explanation of missingness nature in input of situation assessment system

why these data together with the newly arrived ones, Z_1 , constitute the fully instantiated input $Z(T_1) = \langle Z_1, Z_2, Z_3 \rangle$. Someone can make sure that the same takes place at the decision making time T_2 . At the times T_3 and T_4 the situation looks different. Indeed, at the time T_3 decision is initiated by arrival of data Z_2 corresponding to the traffic statistics aggregated for 5 sec. At that moment life time of the most recently received data Z_3 corresponding to traffic statistics aggregated for 100 connections is yet not elapsed and that is why can be used for on-line decision making, whereas life time corresponding to the most recently completed connection, Z_1 , is already elapsed (and new connection is still being in progress) and that is why the data corresponding to Z_1 are useless. Therefore the input at the time T_3 contains missing value of the data Z_1 (see Fig. 2). The similar takes place at the time T_4 , when due to elapsing of the life time of data Z_3 the input of the system assessing the computer network security status contains missing value in the last position.

As a conclusion for the above example it can be stated that asynchronous nature of the situation assessment system inputs and finite life time of these inputs result in the necessity to make decisions on the basis of data with missing values. In general case some kind of prognosis of the missing values can be used. Unfortunately in the example in question the latter is not possible at all due to the fact that, for instance, at the moment of decision making new connection can correspond to the activity of new user and in such a case there is no correlations between previous and next portions of connection-related data.

It is important to note that due to some reasons, in general case of situation assessment systems certain attributes of input data streams can be also

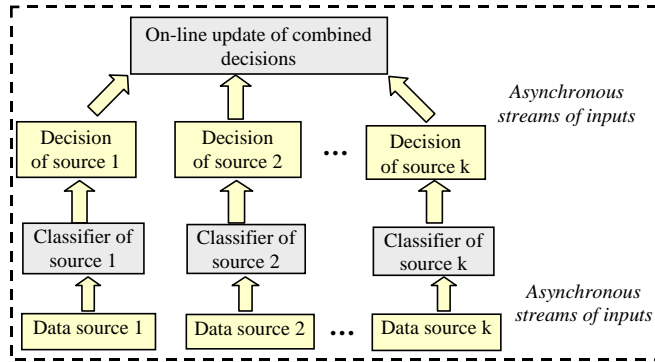


Fig. 4. Decision Fusion Methodology

missing; e.g., if airborne data are used then data can be missing due to meteorological factors, object masking, etc. Thus, missingness of data is a specific property of the input of situation assessment systems and in many cases it is impossible to impute missing values based on some statistical properties of input. The above example from computer network security scope demonstrates this fact.

Thus, a specific problem stated by situation assessment tasks is that the latter is classification task with missing values. Respectively, training and testing of situation assessment systems destined for on-line classification update is reduced to data mining and knowledge discovery from the data sets containing missing values. The respective approach and techniques as well are briefly considered in this paper.

3 On-line Situation Assessment Update: Methodology and Multi-agent Architecture

Let us outline methodology of situation assessment that is based on the ideas of information fusion corresponding to level 2 of JDL model (see Fig. 1). This methodology determines how to allocate data and information processing functions to data source-based level and meta-level. There exist several approaches to fusion of data and information ([3]). In the used methodology at least two-level information fusion architecture is considered. In this architecture local classification mechanisms produce decisions regarding the object states based on particular data sources and then, these decisions are combined at meta-level. This methodology is advantageous in many respects, particularly, it (1) considerably decreases communication overhead; (2) is applicable to the applications where data structures of particular sources are heterogeneous, since only local decisions are forwarded to the upper level, and these

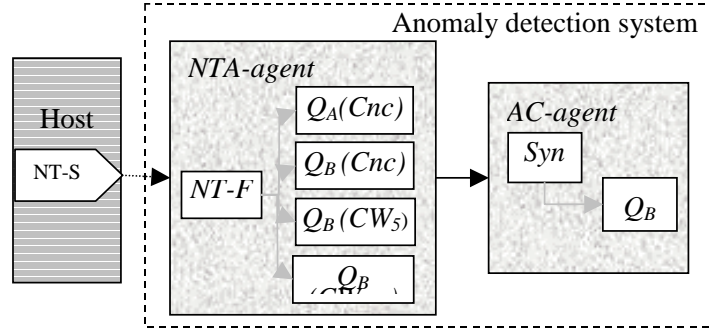


Fig. 5. Anomaly detection system architecture

decisions are represented in binary or categorical scales; (3) there exist a number of effective and efficient algorithms for combining such decisions at upper level and (4) it preserves the source data privacy. A generalized structure demonstrating such a methodology of information fusion is depicted in Fig. 4.

As concerns the above example considering anomaly detection task, at its *first level* four classifiers producing decisions based on particular data sources are used. At the *second level* dealing with asynchronous binary data streams these decisions are combined.

The respective multi-agent architecture of the anomaly detection system is presented in Fig. 5. This architecture consists of two agents of different classes:

- *Agent of Network Traffic-based Alerts, NTA-agent*, that is responsible for detection of abnormal user activity based on particular data streams of output data generated by *Network Traffic Sensor, NT-S*, and
- *Alert Correlation agent (Information Fusion), AC-agent*, that is responsible for combining the alerts generated by classifiers of *NTA-agent*.

The basic functionalities of *NTA-agent* are (1) transformation of raw data structures resulting from traffic data preprocessing (this is performed by *NT-S* component indicated in Fig. 5) into feature structures (this function is realized by *NT-F* component of *NTA-agent*), and (2) producing classifications, *Normal or Alert*, for each particular data stream of the feature structures. The last functions are realized by classifiers $Q_A(Cnc)$, $Q_B(Cnc)$, $Q_B(CW_5)$ and $Q_B(CW_{100})$. Here *Cnc* stands for *Connection* and *Ssn* stands for *Session*.

Accordingly, architecture of *NTA-agent* comprises (1) the component performing computation of the feature structures over the raw data and (2) the components performing classification (*alert generation*) based on particular data streams represented in terms of the feature structures.

Architecture of the *AC-agent* includes two components, and the Q_B classifier is its main component. It is responsible for on-line combining of decisions

produced by classifiers of the *NTA-agent*, thus, producing the on-line assessment of the host security status, *Normal, Abnormal*. In intrusion detection, this procedure is referred to as “*alert correlation*”. The *Syn* component (*Syn* is abbreviator for “*Synchronization*”) is responsible for detecting “too old” data. This component carries out “synchronization” of data in the following manner. Up to the time of receiving a new message, *AC-agent* (more exactly, its component *Syn*) keeps previous decisions of all first-level classifiers (in our case it consists of 4 attributes — labels of host security status produced by classifiers of the *NTA-agent*) with time stamps. While having new message received, *Syn* component changes values of respective attribute and deletes values of attributes that are “out of date”. The updated data vector which can contain missing values is forwarded to the Q_B classifier responsible for “alert correlation”.

The simplified architecture of anomaly detection system described above presents in general features a *generic architecture* of many situation assessment systems intended to on-line update the situation status. The differences between particular cases of such systems can mainly concern the number of data sources used, the number of agents, the number of decision making levels. Nevertheless, in the most cases such a multi-agent system has to comprise more than one level of data processing and decision making, component providing “synchronization” as well as component responsible for first-level decision combining.

4 Data Mining with Missing Values for Situation Assessment

Data mining with missing values is a special problem being investigated for a long time. Here most of researchers mainly dwell upon the methods based on a reasonable assignment (“imputation”) of the missing values exploiting mostly statistical ideas, but such approaches are not applicable to SA due to substantial variety of dynamics of input data streams. Unfortunately, an approach based on using the imputation idea is not relevant to many situation assessment applications. This is the reason why the direct mining of data with missing values has to be used for this application.

An approach to direct mining of data with missing values that does not assume an imputation was proposed in [6]. The idea exploited in it is conceptually simple: if we arbitrary assigned the missing values of training dataset, we would be able to extract the set of *maximally general rules, MGR* [10] using the existing techniques like AQ [9], RIPPER [2], GK2 [7], etc. It is important to note that different assignments would lead to different MGR sets. It was shown in [6] that among different assignments of missing values of training dataset two specific variants exist that lead to the sets of MGR serving as *low R_{low}* and *upper R_{upper}* bounds for any set of MGR R_* corresponding to an arbitrary assignment:

$$R_{low} \subseteq R_* \subseteq R_{upper} \quad (1)$$

where \subseteq is the deducibility relation.

Let us outline how this assignments can be found and how the bounds R_{low} and R_{upper} can be computed. Let $t(i)$ be an arbitrary i -th instance of training dataset and k be the index of the chosen *seed* [10], I_k^+ be the indexes set of *seed* attributes with assigned values. While searching for *MGR* for *seed* $t(k)$, columns of training dataset whose indexes are out of the set I_k^+ are ignored. Let us denote the index set of missing values in a negative example $t(l)$ by $I_{l,k}^-$ and the same set in a positive example $t(r)$, $r \neq k$, by $I_{r,k}^+$. Let us consider *two variants of missing values assignment in the sets* of negative **NE** and positive **PE** examples:

1.	$t_i^l = \neg t_i^k, i \in I_{l,k}^-, l \in NE; t_i^r = t_i^k, i \in I_{r,k}^+, r \in PE,$	(2)
2.	$t_i^l = t_i^k, i \in I_{l,k}^-, l \in NE; t_i^r = \neg t_i^k, i \in I_{r,k}^+, r \in PE,$	(3)

The assignment (2) maximally increases both distinctions between the *seed* and negative examples, and similarities between the *seed* and other positive examples. On the contrary, the assignment (3) maximally increases both similarities between the *seed* and negative examples, and distinctions between the seed and other positive examples. The assignment (2) can be called *optimistic*: it cannot decrease both the generality and coverage factor of any rule of MGR extracted from any arbitrary assigned source dataset. In the assignment (3), that can be called "*pessimistic*", both the generality and coverage factors of rules extracted from any arbitrary assigned source dataset cannot be increased.

The above statements provide a general framework for direct mining of data with missing values. It is obvious that the rule set R_{low} belongs to the set of MGR under search. Other rules have to be selected from the rule set R_{upper} . Let us explain how this can be done. Let alternative classes of situations be denoted as Q and \bar{Q} . The selection algorithm used in this research consists of the following steps applied to each seed:

1. Assign the missing values of the training dataset "optimistically" as in (2) and mine the rule set R_{upper} for classes Q and \bar{Q} : $R_{upper}(Q)$ and $R_{upper}(\bar{Q})$.

3. Assess the quality of the extracted rule sets $R_{upper}(Q)$, $R_{upper}(\bar{Q})$ using certain evaluation criteria based on testing dataset and select the best rules from these sets for use in reasoning mechanism.

4. Design classification mechanism and assess its performance quality.

5. If the above quality is not satisfactory go to 4 to repeat rule selection.

The other procedures are the same as used in ordinal cases [10].

An experiment simulating training of anomaly detection system case study allows to optimistically evaluate the developed approach to online situation

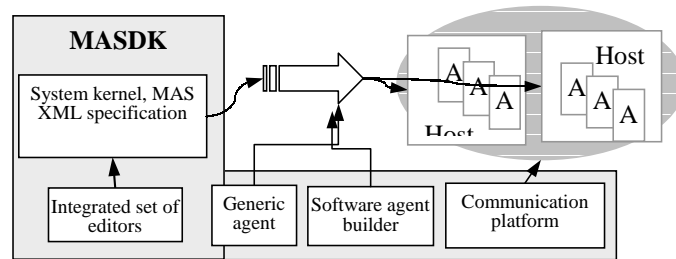


Fig. 6. MASDK software tool components and their interaction

assessment update. Indeed, as applied to the anomaly detection task which uses data of the network traffic level, data of operating system log and data of application level, this approach showed the estimated probability of correct classification about 0.99 if tested on testing and training samples with about 20% of missing values.

5 Implementation Issues

An important issue of multi-agent situation assessment systems is a technology of its analysis, design, implementation and deployment. In this research the MASDK 3.0 software tool supporting all the stages of multi-agent technology is used [5]. This software tool implementing Gaia methodology [16] consists of the following components (Fig. 6): (1) *system kernel*, which is a data structure for XML-based representation and storing of target applied MAS specification; (2) *integrated* multitude of user friendly editors supporting user's activity destined for specification of applied MAS; (3) library of C++ classes implementing what is usually called Generic agent integrating reusable component of agents; (4) communication platform to be installed in particular computers of a network; and (5) *generator of software agent instances*, which performs generation of source C++ code and executable code of software agent instances and also software needed for MAS deployment over the already installed communication platform.

Specification of applied MAS in the system kernel is carried out using the editors structured in three levels. The editors of the *first level* used for description of applied MAS at the analysis stage are as follows: (1) application ontology editor, (2) editor describing roles, names of agent classes, and high-level schemes of roles' interactions, (3) editor describing roles' interaction protocols. Editors of the *second level* supporting specification of agent classes at design stage are as follows: (1) editor specifying model of meta-level behavior of agent (while analyzing input messages and interacting with user and environment); (2) editor specifying particular agent functions and behavior scenarios in terms of state machines; (3) editor specifying software agent

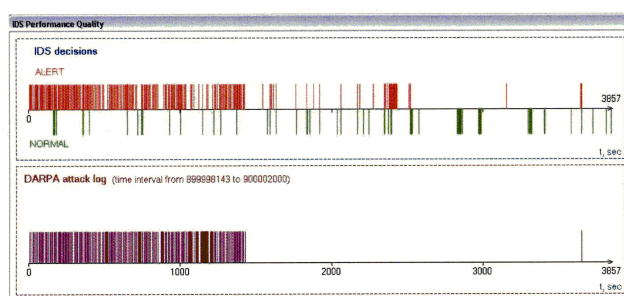


Fig. 7. Visualization component of the implemented anomaly detection system

private ontology, and (4) initial state of agent class mental model. Editors of the *third level* support specification of the MAS components needed for its deployment.

Applied MAS specification produced by designers making use of the above editors is stored as XML file in the system kernel. Generation of source (in C++) and executable codes is performed in automatic mode.

The case study on anomaly detection system described in section 2 and used through all the paper for demonstration of the proposed solutions was implemented through MASDK 3.0 software tool. All the classifiers were trained for Distributed data Mining tool also developed by the authors [4]. One of its components implements the algorithm for direct mining of data with missing values described in previous section. It was used for training of the classifier of meta-level, denoted in Fig. 5 as Q_B .

This system was trained and tested based on well known DARPA data. Fig.6 demonstrates graphically the performance of this multi-agent system for anomaly detection for certain time period lasting about one hour. In the bottom part the time intervals where intrusion takes place are presented in black color, whereas the intervals without intrusions are given in white color. The top part of Fig. 6 presents the performance results of the developed anomaly detection system. In it same colors are used for the same decisions produced by the anomaly detection system. The decisions corresponding to false alarms and missing of signals are presented here below the time axis, whereas decisions corresponding to the correct anomaly detections are given above this axis in black color. The correct detection of normal users' activity are given in white color above the time axis.

It can be seen that although only traffic-based data source was used, and training dataset contains rather high percentage of missing values (about 20%) the results are not too bad, although they are far from "ideal". It also should be noted that the purpose of the above experiments at the current stage of research was not to evaluate the algorithm developed for direct mining of data with missing values but validate the architecture as well as the developed

design and implementation technology destined for engineering of situation assessment systems supporting on-line update of situation assessment.

6 Conclusion

The paper is devoted to certain key issues of the situation assessment problem. It considers the situation assessment task statement that accounts for the fact that input of any situation assessment system is composed of asynchronous data streams possessing various life times, and the input can contain missing values. Other important peculiarity of the situation assessment task statement that is very significant for practice is that situation assessment has to be updated on-line, i.e. that such systems operate in real-time mode.

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 security-related applications.

Acknowledgement

We wish to thank European Office of Aerospace Research and Development of the USAF (Project 1993P) and Russian Foundation for Basic Research (grant # 04-01-00494) for support of this research.

References

1. Ben-Bassat, M., Freedy, A.: Knowledge Requirements and Management in Expert Decision Support Systems for (Military) Situation Assessment. IEEE Transactions on Systems, Man and Cybernetics, vol.12. (2002) pp. 479–490
2. Cohen, W.: Fast efficient rule induction. Machine Learning: 12th International Conference, CA, Morgan Kaufmann (1995)
3. Goodman, I., Mahler, R., and Nguen, H.: Mathematics of Data Fusion. Kluwer Academic Publishers, (1997)

4. Gorodetsky, V., Karsaeyv, O., and Samoilov, V.: Software Tool for Agent-Based Distributed Data Mining. Proceedings of the IEEE Conference "Knowledge Intensive Multi-agent Systems" (KIMAS 03), Boston, USA (2003)
5. Gorodetski, V., Karsaev, O., Kotenko, and I., Khabalov, A.: Software Development Kit for Multi-agent Systems Design and Implementation. In B.Dunin-Keplicz, E.Navareski (Eds.), From Theory to Practice in Multi-agent Systems. Lecture Notes in Artificial Intelligence, Vol. 2296, (2002) 121–130
6. Gorodetsky, V., Karsaev, O.: Mining of Data with Missing Values: A Lattice-based Approach. In Proceedings of International Workshop on the Foundation of Data Mining and Discovery, Japan, (2002) 151–156
7. 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
8. Greenhill, S., Venkatesh, S., Pearce, A., Ly, T.C.: Representations and Processes in Decision Modeling. DSTO Aeronautical and Maritime Research Laboratory, Australia, DSTO-GD-0318 (2002)
9. Michalski, R.: A Theory and Methodology of Inductive Learning. Machine Learning, vol.1, Carbonel, J.G., Michalski, R.S. and Mitchel, T.M. (Eds.). Tigoda, Palo Alto (1983) 83–134
10. Michalski, R. and Kaufman, A.: Data Mining and Knowledge Discovery: A Review of Issues and Multistrategy Approach. Machine learning and Data Mining: Methods and Applications, John Wiley and Sons, (1997)
11. Proceeding of the Fifth International Conference on Information Fusion (IF-2002). Annapolis, MD, July 7–11, (2002)
12. Proceeding of the Six International Conference on Information Fusion (IF-2003). Melbourne, Australia, July 13–17 (2003)
13. Salerno, J., Hinman, M., Boulware, D.: Building a Framework for Situation Assessment. Proceedings of The 7th International Conference on Information Fusion. Sweden (2004) (To appear)
14. Salerno, J.: Information Fusion: A High-level Architecture Overview. In CD Proceedings of the Fusion-2002, Annapolis, MD (2002) 680–686.
15. Than, C. L., Greenhill, S., Venkatesh, S., Pearce, A.: Multiple Hypotheses Situation Assessment. Proceedings of The 6th International Conference on Information Fusion. Australia, (2003) 972–978
16. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. Journal of Autonomous Agents and Multi-Agent Systems, 3, vol.3. (2000) 285–312