

On-Line Update of Situation Assessment: A Generic Approach

Vladimir Gorodetsky, Oleg Karsaev, Vladimir Samoilov

St. Petersburg Institute for Informatics and Automation, 39, 14-th Liniya, St. Petersburg, 199178, Russia

{gor, ok, samovl}@mail.iias.spb.su

Corresponding author Prof. Vladimir Gorodetsky, SPIIRAS, 39, 14-th Liniya, St. Petersburg, 199178, Russia.

Tel. +7(812)3233570, Fax +7(812)3280685, E-mail gor@mail.iias.spb.su

Abstract. Situation is an abstraction that characterizes a complex system consisting of semi-autonomous objects striving to achieve certain particular goals (sequences of goals) and operating in a coordinated mode controlled by a meta-goal of the system on the whole. Situation and objects are discriminated by their "states" taking values from finite sets of classes' labels. Situation assessment, which is the topmost task in many practically important applications, is a classification task aimed at assessing the situation current state and assigning a class label to the situation. The paper covers certain key issues of the situation assessment problem. It analyses formal frameworks proposed for specification of the situation models, and highlights a number of challenging problems associated with situation assessment, the most crucial of which is caused by the peculiarities of input data used for situation assessment, which are a superposition of asynchronous discrete streams of heterogeneous data received from multiple sources and having finite "life time". The paper is focused on the approaches and algorithms intended for on-line updating of the situation assessment, on the situation assessment systems generic architecture and on the demonstration of the proposed techniques and architecture using a computer assurance system solving anomaly detection task.

1. Introduction

1.1. What are Situation and Situation Assessment?

The subject of the paper is situation assessment problem, some critical issues associated with it and solutions proposed. From the conceptual point of view, *situation* is defined as "a structured part of reality that is discriminated by some agent"¹. Situation is an abstraction that characterizes a complex system composed of semi-autonomous objects ("*situation objects*") striving to achieve certain private goals (sequences of goals) and operating in coordinated mode directed by a meta-goal of the system on the whole. Situation objects can be physical (e.g. technical means participating in a rescue operation) or abstract (e.g. components of software in which traces of attack against computer are manifested).

Situation and objects are discriminated by their "*states*" taking values from finite sets of labels. *Situation state assessment* task (hereinafter "*situation assessment*") is a classification task designed for determining the situation class of state at a given moment. Since both the situation's and the objects' states have dynamic nature, situation assessment is a real time task.

Situation assessment is the topmost subtask in many practically important applications commonly known as "situational awareness" [11]. Examples of such applications are associated with assessment of the status of the critical state infrastructures including information systems (for example, security status or performance status); security status of computer networks; monitoring and restoration of inherently dangerous enterprises like nuclear power plants, electric power grids, etc. Important class of such applications covers assessment of threat and prediction of situation development associated with natural and man-made disasters and mitigation of their negative impact on the environment. Very specific class of the applications in question includes security-related applications caused by the necessity to predict terrorist intents and counteract against terrorists' attacks. The list of applications for which the situation assessment is one of the tasks of topmost importance could be continued.

1.2. Object Assessment and Situation Assessment: What is the difference?

In general, the difference between notions of "*object*" and "*situation*" and, consequently, distinction between *object assessment* and *situation assessment* tasks are rather vague. For example, if we consider a multi-level data processing within an application from situational awareness scope [11], an object of an intermediate data

¹ Quoted from [34], which, in its turn, refers to [8].

processing level can play the role of "situation" with regard to its adjacent lower level. That is why it is reasonable to explicate some properties of the task that allow to identify it as a situation assessment task.

Experience confirms that both the distinction between situation and object models and the distinction between object assessment and situation assessment tasks are mainly determined by *data models* used for specifications of objects and situation states, respectively. Let us analyze the aforementioned data models.

In general case a simplified *model of a situation* specifying its states can be presented formally as follows¹:

$$S = \{ \langle x_1, x_2, \dots, x_n, h, t, \{ R_j \}_{j=1}^k \rangle \},$$

where x_1, x_2, \dots, x_n are particular situation objects specified by certain data structures, h —location, t —time, and R_j is m_j - place relation between individual objects, $m_j \leq n$, $j=1,2,\dots,k$. In practice, a *real situation* is recognized through a set of facts to be used for situation assessment by means of matching these facts with the situation model [35].

Thus, a real situation state is constituted by states of situation objects, their locations at given time instants, their activities during a time interval and meaningful relations between objects. According to the common opinion, objects *activities* and *relations* between objects are essential *distinctive properties* of the situation assessment as compared with the object assessment [35]. As a rule, these properties are the most informative components of the situation model intended for its state assessment. Existence of specific relations and kinds of the situation object activities considerably affect the resulting assessment of the situation state. Let us note that objects' states are normally specified only in terms of features. While discussing the distinction between object assessment and situation assessment, it is also said in [18] that "...The main distinction of higher level data fusion² is that it is no longer emphasized on physical objects but on the *relations* amongst the objects constituting a situation. The task of relation representation and their use for situation assessment is much more computationally intensive. As a rule, the relations are poorly understood by persons (experts, knowledge engineers) involved in modeling and specification of situations and associated notions and tasks (situation state, treat, assessment of the above entities, etc.)." This opinion is supported by D.Lambert [21], who explicitly states that "Situation fusion is the process of utilizing one or more data sources over time to assemble a representation

¹ Quoted from [34] and slightly changed.

² According to the model of data and information fusion accepted in situation awareness community (so called JDL model [40]) situation assessment is a task of higher level fusion but object assessment task—not.

of *relations of interest* in an environment. A situation assessment is a stored representation of relations between objects obtained through situation fusion".

The relations between situation objects can be of diverse nature [35]: spatial, temporal, ordering, etc. Importance of particular relations as of the properties specifying situation states is highly application-dependent. *Temporal* relations normally reflect time ordering of events and other time associated relations between them that particular situation objects undergo. Examples of such events are the completion of an aircraft refueling mapped to a time interval, the arrival of a subset of situation objects to a predefined region before a given moment of time, the appearance of an alert sequence in some information sources of a computer network assurance system, etc. Other relations between objects may take into consideration their spatial configuration. In some cases dynamic spatial relations are of the greatest importance (e.g. "objects are approaching each other").

The second peculiarity of the situation assessment task is that situation-related information arrives continuously from multiple distributed sensors. Since situation is an issue of dynamic character, the input information used for situation assessment may have different *time stamps*. As a rule, different objects possess different dynamics and, consequently, different components of input information possess different *life time*. Combining such information to produce situation assessment constitutes a theoretical problem that has not been explored well enough.

The third important peculiarity of the situation assessment task is that input data representing a situation arrive to a situation assessment system as *a set of asynchronous data streams*. This issue results in the fact that at the time of the situation assessment update different components of data structure specifying situation state are marked by different time stamps. As a consequence of this, information may grow out-of-date before the moment of the situation assessment updating comes.

Additionally, a certain part of the information specifying situation state may be either not collected or lost. For example, airborne surveillance system may fail due to meteorological factors or masking. This makes the situation assessment task especially difficult, because this factor results in missing data values.

The aforementioned peculiarities of data and information used for situation assessment will be clearly illustrated below based on an example of real-time anomaly detection system.

The paper is focused on the approaches and algorithms intended for on-line update of the situation assessment, on the situation assessment systems generic architecture and on a case study based demonstration of the proposed techniques and architecture application, an anomaly detection system being used as an example.

The rest of the paper is organized as follows. Section 2 contains a brief survey of the topic-related works investigating situation assessment problems, outlines the main research directions in this area and presents a more detailed analysis of the recent works devoted to the formal frameworks proposed for specification of the situation models and situation assessment. This section also outlines the most challenging problems in the area being explored. Section 3 describes a case study, i.e. anomaly detection task, aimed at the assessment of the computer security status, with the focus on the input data model. This case study is later used for an example-based demonstration of the main peculiarities of the input data model and of the generic structure of data processing in the situation assessment procedure. It is also used for explanation and evaluation of the basic solutions proposed in the paper (formal framework, approach and techniques used at the design stage and as well as the operation stage of the target real time situation assessment system). Generic architecture of the systems in question is also demonstrated based on this case study. The latter is the key point of section 4. Although this section considers the multi-agent architecture of an anomaly detection system, its specific functionalities and their allocation to agents, this architecture is generic for many applications of situation assessment scope. Section 5 considers the main algorithmic issues of the situation assessment system design, in particular, learning of on-line situation assessment update based on asynchronous data streams arriving from many distributed sources. Due to use of a threshold-based model for information ageing, the aforementioned task is reduced to the data mining with missing values. The paper describes the developed algorithm for direct mining of such data and presents the results of its evaluation. Conclusion summarizes the results achieved and outlines the future works.

2. Situation Assessment Related Works: Approaches and Challenges

Although understanding of the importance of the situation assessment task was reached more than fifteen years ago [41], this task until recently was not a subject of intensive research. Currently this task is an object of the information fusion community attention [19]. The main aspects of interest in this respect are listed below:

1. Formal frameworks suitable and adequate for specification and support of fusion of the incomplete data and information continuously arriving from multiple heterogeneous data sources;

2. The most advantageous architectures for situation assessment systems;
3. The ontology and other semantic-oriented specification means providing for the interaction of heterogeneous "actors" involved in the processes associated with the situation assessment (experts operating at different levels of application abstraction, computers, computer programs, simulation models, heterogeneous means and techniques for situation-related information display, etc.);
4. Different aspects of the user interface supporting interaction between the system and the experts;
5. Integration of textual and semi-structured information produced by experts with well-formalized information created using the sensor-supplied data and symbolic models;
6. Methodology and technology for engineering, implementation and deployment of the situation assessment systems and supporting software tools, etc.

This paper primarily deals with the formal framework for information fusion aimed at the situation assessment. So, provided below is a brief analysis of the recent works concerning only this aspect of situation assessment problem, and an outline of associated challenges is given. Information on more general aspects of the situation assessment problems can be found in the survey [1].

2.1. Formal frameworks

One of the most popular formal frameworks used for situation assessment is Bayesian Belief Network (BBN) and some of its enhancements. The publications advocating the use of BBN are numerous. In particular, the paper [7] states that BBN "can be thought of as a graphical program script representing casual relations among various ... concepts represented as nodes to which observed significant events are posted as evidences". The idea of the paper is to construct BBNs from sub-networks. An important advantage of this approach is that it uses BBNs distributed across multiple computers exploiting simple standard functionalities named "publish" and "subscribe" what allows for significant enhancing of the inference efficiency. One more advantage of this distributed BBN approach is that it provides the possibility to operate with the BBN-oriented specification of a situation at different levels of abstraction. The latter is an important property, because at many organizations solving situation assessment tasks users of different responsibilities are involved. The paper also claims a potential capability of distributed BBNs to cope efficiently with real time situation assessment in dynamic environment, even though it offers no proof in support of this claim.

Other paper of the same research group [6] discusses the BBN truth maintenance problem after its updating due to receiving of new evidence. When posted into a BNN node, the new evidence can lead to violation of

BNN consistency thus hindering the use of BNN for decision making. Therefore before posting incoming evidences into a BNN node it is necessary to check the consistency between the node's currently expected state and the newly occurring state. The authors proposed the truth maintenance procedures, whose purpose is to revise some default assumptions.

The [43] discusses an idea of using a distributed version of Bayesian networks for situation assessment and it is close to the above surveyed paper [7]. It suggests using a so-called "Bayesian Network Fragment" notion encapsulating a certain fragment of the entire knowledge. The positive features of this kind of BBN are (1) design simplification and (2) making possible the automated design of task-oriented BBNs by composing them from much simpler BBNs representing knowledge fragments (including reusable ones). This results in the increase of the efficiency of Bayesian inference (belief propagation) and allows for parallel execution of the procedures implementing situation assessment. The proposed BBN model is used for distributed event-driven reasoning about uncertain numbers of uncertain hierarchically structured entities based on incomplete evidences. It also provides the necessary solution modularity. The example presented in the paper shows that the proposed framework possesses the ability to recognize the organization hierarchy (in the paper – military force structure) as a part of situation assessment result.

Another formal framework used for the information fusion in situation assessment tasks is the belief fusion approach based on Dempster–Shafer model [34]. From the very beginning this framework was oriented at combining evidences, and it is natural that it is widely used in the situation assessment related applications. This framework, its advantages and shortcomings are well known. Its main advantage is that it deals with uncertain and incomplete information called here "evidences" and provides a simple way for combining such evidences when they arrive. Unfortunately, for particular applications, it is not simple to build a Dempster–Shafer model, i.e. to determine a domain-related frame of discernment and basic belief assignments. This shortcoming noticeably restricts the possibilities of Dempster–Shafer model practical use. Its another shortcoming is that the Dempster-Shafer rule for evidence combining can be defined in many ways [22] what leads to a certain degree of arbitrariness of the result, thus decreasing its worthiness. A certain extension of this framework called Dezert–Smarandache theory, DSMT, specifically in the context of the situation assessment, was proposed in [9], [10]. However, the practical applicability of this extension has not been justified so far.

One more type of formal frameworks developed specifically for decision fusion usable for the situation assessment is based on combining the decisions produced by the classifiers associated with particular data

sources. Several approaches of such type were proposed. Although the idea of the first of them, *stacked generalization* [42], is very simple, it gave birth to several particular techniques for decision combining. Its most advantageous variant, "*meta-classification*", was proposed in the early 1990s (see, e.g. , [29]). According to this approach, a procedure specifically designed for decision combining, which is called *meta-classifier*, is a conventional, for example, rule-based classifier whose input is constituted by a number of outputs of source-based classifiers.

A two-step technique is used for the meta-classifier training. At the *first step* a so-called *meta-data sample* is computed. It is done based on the source-based classifiers testing. For this purpose a special testing data sample is used. The result of base classifiers testing is represented as an array of classification labels, with possibly some values missing if the data from some sources have not been received (e.g. not measured or out of date). Some of these labels can be correct, some – incorrect. At the next step, this array is extended by the correct classification label and the resulting array, with possibly missing values is viewed as an instance of the *meta-data* to be used for meta-classifier training and testing. The *second step* of meta-classifier design consists in conventional training and testing based on the computed meta-data sampling. Different aspects of this model of decision fusion as applied to situation assessment are considered in [12], [14], [15].

The basic idea behind the *second group* of decision combining techniques exploits the experience-based assumption that the quality ("competence") of each classifier varies in different regions of the representation space. The algorithms exploiting *evaluation of base classifiers' competences* with regard to each particular record of input data are called *competence-based* meta-classifiers. This idea was first proposed in [39] and [24] and further developed in [40]. In this group of decision combining techniques, a special procedure called "*referee*" is associated with each base classifier. Its responsibility is to assess the competence of the corresponding classifier in regard to a particular instance of input data [28]. After competence assessing, the referees negotiate with each other to determine the most competent classifier whose decision is selected as the final one. To provide the referee with the ability to operate accordingly, a training procedure is exploited. The referee training uses the same training dataset as for training of the corresponding classifier itself. The distinction is that the results of testing of the base classifier are labeled as "correct" or "erroneous" in respect to each particular instance of the training dataset. As a result, the instances of the testing dataset are divided into two classes. One of them corresponds to the region where the classifier is "competent" while the other corresponds to the set of instances classified incorrectly. In regard to the latter the classifier is evaluated as

"incompetent". This partition of testing data set is then used as the referee of the corresponding classifier training. Different aspects of the last framework as applied to situation assessment are discussed in [12], [14], [15].

2.2. Challenges

Situation assessment problem practically has no research history. It has become a subject of intensive research only recently and now this problem is at its infancy, contains many unsolved tasks and incomprehensible aspects. In a sense, current research in the situation assessment scope is of "ad-hoc" and unsystematic character. R.Mahler [23] indicates that the existing problem statements of the situation assessment task mostly cover particular applications and therefore the obtained results have limited value. *To formulate the situation assessment task in general terms amendable to algorithmic solution is one of the challenges.* A few theoretical and some other efforts are typically aimed at limited specific cases that are being solved using heuristic based approaches – in such a way the current state of the art concerning general problem statement of the situation assessment task can be described. The same is stated by J.Bierman [2] who points out that it is necessary to define the high level information *fusion problem at a sufficient level of abstraction and to develop an appropriate model* answering the suitable fusion methods.

It is well known that the most informative components of the situation model are specified in terms of relations existing between the situation objects. Unfortunately, there exists a problem called "*relevance of relations*" [20]. Indeed, the total number of the relations that can be defined over a given set of objects is huge. This property practically excludes a possibility to check all of them or at least the major part of them to find out if they are relevant for the situation description ("informative" for situation assessment). Experience proves that, typically, only a small subset of such relations would be relevant. Thus, the *next challenge* [20] is to develop methods making it feasible to determine which of the numerous relations are really relevant for the task in question and, therefore, should be monitored. Additional challenge here [20] is *how a program can determine whether a particular relation is held or not.* For any domain a set of such rules must be developed.

A number of grand challenges are associated with the dynamic nature and behavioral aspects of any situation. P. Svensson [38] draws attention to the number of hard and hot problems of the above-mentioned nature. Amongst them, there exists an important unsolved and *challenging problem* [38] *of creating a network-based high-level structure intended for real-time behavior recognition and forecasting,* development of methods

allowing for discovering, learning and recognition of the vague dynamically changing relation structure combining the situation objects.

The [31] also provides a thorough analysis of the challenging problems associated with the dynamic nature of any situation and, therefore, dynamic nature of any situation assessment system input information resulting from the *continuous* and *asynchronous* mode of information gathering. This challenging aspect of the situation assessment system operation and an associated challenging problem, consisting in the *on-line situation assessments update* based on asynchronous data streams *having finite and various life times* are the main subjects of this paper.

3. Multi-agent Architecture of Situation Assessment System

3.1. Common Features of the Situation Assessment Systems

Disregarding the noticeable diversity of types of the situation assessment-related applications, they have a lot of common in the decision making technology forming the core of the situation assessment systems. Indeed, in any applied situation assessment system the decision making procedure is performed according to a layered structure. Most of the applications in question require preprocessing of the raw sensor data. is necessary. For instance, the input data may be represented as infra red images of different spectral ranges where filtering and detection of the objects of interest are the tasks to be performed at the *first layer*. Other example is the intrusion detection as applied to the computer network security, where the input data are formed by computer network traffic recorded in the TCP dump. These data are raw and redundant and that is why cannot be plainly used by classification procedures. Other common feature of the situation assessment tasks is the multiplicity and diversity of the data sources that causes the necessity to use source-based classifiers forming the second decision making layer. Combining the source-based decisions is the task of the next, upper layer of the situation assessment procedures. The common property of the above procedures is that due to asynchronous nature of various data sources inputs, decision combining procedure has to produce the decision based on the data having different time stamps. Thus, in general case the situation assessment is a hierarchical procedure and comprises three layers. Therefore, various applied situation assessment systems can be built based on the same generic architecture.

Since recently, Multi-Agent System (MAS) paradigm is considered to be the most promising architecture for situation assessment systems because MAS paradigm is specifically oriented to the design and implementation

of large scale distributed intelligent systems, like situation assessment ones. Indeed, as a rule, situation assessment applications are naturally distributed: data sources are spatially distributed; data processing is performed in a distributed manner, the systems and/or users interested in the results of the situation assessment system's operation are also distributed. If some data of various sources are private or classified (military data, commercial data, etc.) then such data are not available for the centralized processing. Nevertheless the data holders can make this data available for the situated agents in order to have the private data processed locally, without revealing their content. Recent publications (see, e.g., [30]) show the growth of the popularity of MAS paradigms when applied to situation assessment systems. So, it is only natural that here proposed paper also dwells upon a generic multi-agent architecture for the situation assessment systems

A well known and in many respects typical representative of the applications where the situation assessment task has to be solved is the assessment of the security status of a computer. A particular case of this task, namely the anomaly detection is considered in this paper in order (1) to illustrate by example some key peculiarities of situation assessment tasks, corresponding systems and associated problems, and (2) to explain and demonstrate the proposed framework, techniques and architecture supporting design and operation of the situation assessment systems solving tasks on-line.

3.2. Case Study and Input Data Model

An anomaly detection system analyses the computer security status based on the data being received from many sources. Each of these data source produces a discrete data stream. Thus, the input data of an anomaly detection system is a superposition of such discrete data streams. Successive portions of data of these data streams arrive asynchronously and with various average frequencies. In the case study, the data sources of the network level resulting from preprocessing of the *TCP* dump that represents the network traffic are considered. It is important to note that in general case selection and/or design of data sources for anomaly detection is a task that involves a lot of preliminary creative work followed by certain programming efforts. According to the modern view, the data sources associated notions (situation objects), their attributes and states have to be represented in the system ontology.

In our case, the *basic situation objects* represented in the anomaly detection system ontology as notions are the following: "*Network packet*", "*Connection*" (user's session) and "*Network interface*". Respectively, the attributes of each of these objects specifying its status constitute a particular data source. It is assumed that the status of each of such object can be labeled as "*Normal*" or "*Alert*". The last label corresponds to a suspicious

activity of computer users manifested in the corresponding object, which can also be called as "*Abnormal*". Here, the situation assessment problem consists in the evaluation of the computer security status based on the data sources specifying the situation objects of the above classes.

The data sources selected, the infrastructure defined for them, the information flows as well as the developed structure of data processing used for computer security status assessment are presented in Fig.1. In it, the data sources are presented by the grey rectangles. The arrows show the information flows between the components. The ellipses correspond to the decisions concerning the security status of the situation objects ("*Network packet*", "*Connection*", and "*Network interface*"). The bevels represent the classifiers assessing the security status of the situation objects based on the particular data sources. Meta-classifier is also represented as a bevel.

While summarizing the peculiar properties the *input data model* used in the case study, which, in fact, are typical for many situation assessment applications, it is important to note that:

1. Data sources present *data of different aggregation levels* resulting from preprocessing of the *TCP* dump data.

These aggregation levels are: (a) the data specifying particular network packets; these data are presented as vectors of binary sequences specifying the stream of certain parameters of *IP* packet headers; (b) the data specifying particular connections (users' sessions); as a rule, each connection involves several network packets; (c) the statistical data, specifying the network interface in terms of various attributes computed within sliding the windows of different lengths and having different shifts; the duration and shift of a sliding window are determined either by means of direct assignments of duration length and shift step (in Fig.1 the respective data sources are denoted as *TimeWindowFeatures* and *TimeWindowTrafficFeatures*) or, implicitly, by assigning the total number of connections within a window and the consequent shift in the same terms (*ConnectionWindowFeatures* and *ConnectionWindowTrafficFeatures* in Fig.1). In the considered case study, four sliding window-based data sources are used. Let us recall that all of them are used to assess the security status of the situation object "*Network interface*" and are viewed as the particular data sources specifying the same object.

2. Data sources used for computer security status assessment include the data (a) computed directly from the *TCP* dump as well as the data produced in a more complicated way, particularly, (b) as the results of the classifications produced by the packet-based and connection-based classifiers, and (c) as the results of the computations performed using both the connection-based information and the sliding window-based statistical information. These data represent interconnections given over the two situation objects (notions of

the ontology "*Connection*" and "*Network interface*") expressed in terms of the number of abnormal connections ("alerts") within the time intervals corresponding to each sliding window of the four data sources specifying '*Network Interface*".

3. the input data of the *meta-classifier* comprised the discrete *asynchronous data streams* of decisions of the base classifiers whose events arrive to the meta-classifier with different average frequency.

Here the structures of the data from particular sources and the respective classification algorithms are not described in detail; the various aspects of these classification algorithms training and operation are not touched upon either. Of course, we do not state that these tasks are simple, no, they are not. In the paper we deliberately omit all the aspects associated with the source-based classifiers although their training and testing require much more effort and more sophisticated algorithms than the same activities as applied to the meta-classifier. We do so because these aspects are related to the object assessment task, but not to the situation assessment and, thus, the aforementioned tasks are out of the paper scope.

We assume that each base classifier produces decision at every time instant when it receives new input information, and this decision is immediately forwarded to the meta-classifier intended for combining the decisions produced by different base classifiers. Fig.2 illustrates the asynchronous nature of the input data streams arriving to various source-based classifiers presented in Fig.1. It explicitly shows that these data streams are of different average frequencies and irregularly structured. For on-line solving the decision fusion task, anomaly detection system has to update the security status of the computer at each time instant when an event (a decision of a base classifier) arrives to its input.

The training and testing datasets needed for the design of anomaly detection system classifiers and for evaluation of its performance were generated using a specially developed and implemented model of network traffic including normal as well as abnormal traffic. The dataset of the "*Abnormal*" class comprises the instances reflecting four types of attacks, which are *Probing*, *Remote to local (R2L)*; *Denial of service (DOS)* and *User to root (U2R)*. The particular instances of these classes of attacks presented in the case study are *SYN-scan*, *FTP-crack attack*, *SYN flood*, and *PipeUpAdmin* ([4], [27], [32], [33]). Preprocessing of such *TCP* dump was based on using the *TCPtrace* utility and some other ad-hoc programs.

3.3. Generic Multi-agent Architecture: An Example

Among the many different data processing procedures involved into assessment of the security status of a computer (in general case – used for situation assessment) the most specific ones are the procedures whose input is formed from different data sources and, therefore, asynchronously. Because the data of different data streams have different "life time" and arrive to the meta-classifier at different time instants, at the instant when a decision is some data received earlier can turn out to be "too old" and, thus, useless for decision making. Therefore, it is necessary to have a procedure detecting the "age" of the data received from data sources to detect "too old" data.. We call this procedure "*data synchronization*". The necessity of such a procedure is typical for situation assessment systems operating in on-line mode. Thus, the generalized architecture of multi-agent situation assessment system must be able to support the structure of multi-level decision making in which the typical agents' roles are: *R1: "data synchronization"*, *R2: "feature space transformation"* and *R3: "decision making"*.

Let us outline the developed architecture of the anomaly detection system which in many respects is generic for other situation assessment applications. This architecture is depicted in Fig.3.

External environment of the anomaly detection system includes two sensors receiving primary (raw) data, and two storages where training and testing data can be accumulated and stored if necessary. *The sensors* are described below:

- *NL-S (Network Level Sensor)*. It receives the packets of the traffic (input and output) and preprocesses the traffic packets (as they are stored in *TCP dump*) in order to generate the primary input data for the anomaly detection system. In particular, this sensor generates *PacketBased* and *ConnectionBased* data. In the decision making structure, these data are considered as particular data sources having multiple attributes.
- *HL-S (Host Level Sensor)* carries out computing of the features of the data sources of the Operating System level manifested in *Security log, System log and Application log* (in our case–for Windows platform). This sensor generates the events "*SystemEvent*", "*SecurityEvent*" associated with the objects "*OS processes*", "*Users' sessions*" and "*OS service*" of the host level used for classification of the security status of the host and services.

It is assumed that *NL-S and HL-S sensors* perceive events of real-life data (*TCP dump, Operating System Security log, System log and Application log*) or read the data collected and stored earlier. A well known representative of the last type of data is DARPA dataset [5].

In addition, external environment includes two more components which are:

- *DB (Data Base)* that stores the training and testing data (in external storage) if necessary and
- *Buffer* that also stores the training and testing data (in *RAM*) if necessary.

The architecture of the anomaly detection system itself includes the agents of three following classes (Fig.3):

- *Agent of Network Level Alerts, NLA-agent*. It is responsible for detection of abnormal user activity based on the output data generated by the *Sensor of the Network level, NL-S*;
- *Agent of Host Level Alerts, HLA-agent*, is responsible for detection of abnormal activity of users based on the data perceived and partially preprocessed by the *Sensor of the Host level, HL-S*. It handles the data of the operating system level. The components of this agent are shown in Fig.3 schematically, because *Host Level Alerts* agent has not been completely implemented yet.

The aforementioned agents execute two identical roles: *R2* and *R3*, and handle the data of the network-based level and the host-based level respectively.

- *Alert Correlation agent, AC-agent*. This agent is responsible for combining of the alerts generated by the agents of the *NLA-agent* and *HLA-agent* classes in order to on-line updating of the computer security status assessment. This agent plays three roles, which are *R1*, *R2* and *R3*, and their joint performance allows for making decisions based on the asynchronous streams of decisions produced by *NL-A* and *HL-A* agents. (Let us recall that the role *R3* carries out the data synchronization functionality).

A multi-agent *software prototype* implementing the developed generic architecture as applied to the computer anomaly detection system, was carefully designed, implemented and deployed in a computer network. It was then used in the experiments intending evaluation and validation of the developed architecture as well as evaluation of the efficiency and accuracy of the data mining algorithm (see section 4) and respective classification mechanism specifically intended for on-line situation assessment update based on asynchronously arriving data about the current situation. Let us note that this software prototype was developed using MASDK software tool, Multi-agent System Development Kit [13] that has recently been developed by the paper authors.

4. Learning of On-line Situation Assessment Update

4.1. Model of Data Ageing

It was mentioned in subsection 1.2 that on-line update of the situation assessment based on asynchronous data streams can be reduced to a specific classification task with missing values. Respectively, learning of the

meta-level classifier of a situation assessment system is a task of mining of binary data with missing values. Let us demonstrate this fact graphically.

Sensor data are being gathered continuously, they are time-stamped; particular data streams arrive into the situation assessment system with different average frequencies and possess finite values of "life times". The values of "life times" can considerably vary for the data belonging to different streams. Finiteness of the life times results in the fact that after a certain period of time a part of the data becomes useless for situation assessment. Therefore, at the moment of situation assessment update some attributes may have not been assigned values and, thus, input data vector to be used for the situation assessment update contains *missing values*. Fig.4 illustrates this fact graphically for a simplified case when the input data of the meta-classifier are collected from three data sources. Indeed, let us assume that new data ("events") arrive at the time instants T_1 , T_2 , T_3 and T_4 . According to the accepted strategy, computer security status updates have to be performed at the same time instants, T_1 , T_2 , T_3 and T_4 . Decision making at the time instant T_1 is initiated by arrival of the data denoted Z_1 ; this event corresponds to the completion of the most recent connection. At that moment, *life times* of the most recently received data corresponding to the traffic statistics denoted in Fig.3 *TimeWindowFeatures*, Z_2 , and corresponding to the traffic statistics presented in Fig.3 as *ConnectionWindowFeatures*, Z_3 have not elapsed yet and that is why these data in combination with the newly arrived data, Z_1 , constitute the completely instantiated input vector $Z(T_1) = \langle Z_1, Z_2, Z_3 \rangle$. It can be seen that the same situation takes place at the time instant T_2 .

A different situation occurs at the time instants T_3 and T_4 . Indeed, at the time instant T_3 decision making is initiated by arrival of data Z_2 , i.e. the decision is made based on *TimeWindowFeatures* data source. By that moment life time of the most recently received data Z_3 corresponding to the decision made by the classifier handling with the *ConnectionWindowFeatures* data source has not elapsed yet and, therefore, it can be used for decision making at that time, whereas the life time of the most recent decision made by *Connection*-based classifier, Z_1 , has already elapsed (and a new connection is still being in progress). Hence, the value of Z_1 is useless. Therefore, the input data of the meta-classifier at the time instant T_3 contain a missing value of data Z_1 (see Fig.4). A similar situation takes place at the time instant T_4 , when due the life time of the decision Z_3

having elapsed, the input of the system assessing the computer security status contains a missing value in the last position.

Thus, the asynchronous nature of the situation assessment system inputs and finite life time of these inputs result in the necessity to make decisions using data with missing values.

Let us remark that in general case other models of data ageing can be used, while here we consider just one of the possibilities.

4.2. An Algorithm for Direct Mining of Data with Missing Values

Mining of data with missing values is a special problem of data mining that has been investigated since 1980-th. As a rule, these investigations are primarily focused 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 situation assessment task because of the significantly different frequencies of arrival of data from various 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 [16]. This approach was applied for training of the *AC-agent* (Fig.3) performing on-line situation assessment update and exposed good properties. Let us outline the basic idea of this approach.

The input of the *AC-agent* is composed of the binary data assigning values from the set $\{Alert, Normal\}$ that can be coded as $\{1, 0\}$ respectively. The idea of the approach being discussed is conceptually simple and is presented below. If we assigned the missing values of the training dataset in an arbitrary way we would be able to extract a set of *maximally general rules*, *MGRs* [25] using an existing technique like AQ [25], or RIPPER [3], or GK2 [17], etc. Different assignments would lead to different *MGR* sets. Let us denote as R_* an *MGR* set for an arbitrary assignment of missing values.

It has been found out that there exist two special sets of *MGR* that can serve as *low* and *upper bounds* for all possible sets of rules corresponding to any potentially possible assignment of missing values:

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

where R_{low} and R_{upper} are the *low* and the *upper* bounds respectively for all the sets of *MGR* and " \subseteq " is the deducibility relation. Informally, it could be said that the set R_{upper} corresponds to the "*optimistic*" and R_{low} –

to the "pessimistic" assignments of the missing values. we will briefly explain below how these bounds are built [16].

Let us denote an arbitrary i -th instance of the training dataset as $t(i)$. Let k be an index of the chosen *seed* [25], I_k^+ be the set of indexes of the assigned (not missing) attributes of the above *seed*. While searching for an *MGR* corresponding to the *chosen seed* $t(k)$, let us ignore all the columns of training dataset, whose indexes do not belong to the set I_k^+ . Let us denote the index set of the missing values in an arbitrary negative example $t(l)$ as $I_{l,k}^-$ and the index set of the missing values in a positive example $t(r)$, $r \neq k$, as $I_{r,k}^+$. Let us consider now two variants of missing values assignment in the negative and positive examples:

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

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

The assignment (2) is so organized 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 assignment (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 justifiably referred to as the *optimistic*, since it cannot decrease both the generalization level and coverage factor of any rule of the *MGR* extracted from the completely assigned dataset. In the second assignment, which can be reasonably called "pessimistic", both the generality and coverage factors of rules extracted from completely assigned dataset cannot be increased. The *Theorem* below strictly formulates the above-provided facts and shows how to find the upper R_{upper} and low R_{low} bounds an *MGR*.

Theorem [16]. *Let us assume that seed* $t(k)$ *does not contain missing values, let* 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;* R_{upper} *be the set of all* *MGRs* *corresponding to the assignments (2) of positive and negative examples, and* R_{low} *be the set of all* *MGRs* *corresponding to the assignments (3). Then* $R_{low} \subseteq R_* \subseteq R_{upper}$, *where* \subseteq *is the deducibility relation.*

This *Theorem* provides a general framework for mining data with missing values. It offers the set of rules containing the set of *MGR* being searched, but it does not show how to select the rules from R_{low} and R_{upper} to be used then for classification. However, the practice proved that the target rule set can be selected from R_{upper} through an algorithm based on a testing procedure. Let us explain this point, while assuming that the alternative classes (in our case study—"Alert" and "Normal") are denoted as Q and \bar{Q} . Conceptually, the core of this algorithm consists of the following steps applied to each *seed* used for design of the classification mechanism:

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

The other procedures are the same as for data without missing values (e.g., see [25]).

An extended experiment aiming at testing of the on-line anomaly detection system whose classification mechanism was built based on the developed technique allows to optimistically evaluate the proposed approach to direct mining of rules from the datasets with missing values. Indeed, the anomaly detection system that has been trained according to the above discussed algorithm shows, when using the testing dataset, the estimated probability of correct classification close to 0,99. The experimental results also allowed to extend the above optimism with regard to other applications designed for on-line situation assessment update based on the asynchronous data streams arriving from multiple heterogeneous data sources.

5. Conclusion

The subjects of the paper are an analysis of the current key problems of the situation assessment task and solution of some unsolved problems associated with the above task. Situation assessment is an important component in many topmost applications of intelligent information technology. The aforementioned analysis

shows that any situation assessment task is a task of dynamic, real-time nature, and that this task, if stated so, has no satisfactory solution yet.

The major difficulty associated with solution of the task in question consists in specifics of the input data model to be used for situation assessment. As a rule, these data are received from multiple sources and arrive into the situation assessment system in a form of asynchronous discrete streams of data, besides, the average frequencies of input data arrival can vary noticeably. Therefore, data of various input streams have different "life time" and if the data age exceeds their life time, such data are useless for the situation assessment.

The first contribution of the paper is the proposed generic multi-agent architecture for situation assessment systems operating on the basis of asynchronously arriving multiple data streams. This architecture is generic for many applications of situation assessment scope differing only in the number of data sources, input data structures and numbers of instances of standard agent classes, software agents, composing the target multi-agent situation assessment system.

The situation assessment system architecture assumes a two-level situation assessment procedure. At the first level an assessment of the states of the situation objects jointly determining such an abstract entity as situation is performed. It is important to note here that object state assessment task is a conventional classification task, meaning that all data needed for a particular object assessment arrive at the same time instant; this task is beyond the paper scope and is not considered. At the second level the task of situation assessment based on particular object assessments and relations between objects is solved. It corresponds to the decision combining (decision fusion) task. However, due to the peculiarities of the input data of typical situation assessment system, the existing procedures for decision combining cannot be directly used what inspires the development of a new approach.

The second contributions of the paper is the proposed threshold model of data ageing reducing the situation assessment task with the input comprising several asynchronous data streams to the classification task with missing values. New approach to direct mining of such data resulting in a rule-based classification mechanism presented in the paper is one more contribution. Three aforementioned components, i.e. the data ageing model, the proposed approach to direct mining of data with missing values and classification mechanism form together the basis for efficient solution of the situation assessment task on-line, assuming updating the situation assessment at any time when new information arrives in the system from any source.

The multi-agent architecture for on-line update of the situation assessment proposed in the paper as well as the developed method for the on-line update of the situation assessment based on asynchronous discrete streams of input data were completely verified through a developed multi-agent software prototype implementing case study in the computer anomaly detection. Let us note that the design, implementation and deployment of the multi-agent anomaly detection system were carried out using MASDK, Multi-Agent System Development Kit [13]. It is proved in the paper that this application possesses the main characteristics that are typical for most situation assessment applications. Due to this fact, the results received for the anomaly detection system and conclusions made based on this case study can possibly be extended to other situation assessment applications.

Future works associated with the on-line update of the situation assessment task will investigate other model of information ageing as applied to asynchronous data streams input. Another promising and important direction of the research in the situation's study is the situation prediction.

Acknowledgement

We wish to thank The European Office of Aerospace Research and Development of the USAF (Project 1993P), and The Russian Foundation for Basic Research (grant # 04-01-00494) for support of this work. We wish to thank Office of Naval research Global (USA) who financially supports the traveling that has made possible the public presentation of the results of this paper.

We would also like to thank Prof. I. Kotenko and his Ph.D. student M.Stepashkin, who generated the training and testing datasets used for evaluation of the proposed approach to mining and classification of data streams.

References

- [1] M.Ben-Basset and A. Freedy, Knowledge Requirements and Management in Expert Decision Support Systems for (Military) Situation Assessment, IEEE Transaction on System, Man, Machines and Cybernetics, 12 (1984), 479-490.
- [2] J.Biermann, Challenges in High Level Information Fusion, Proceedings of the 7th International Conference on Information Fusion, Stockholm, Sweden, 2004, pp. 528–529.
- [3] W.Cohen, Fast efficient rule induction, Machine Learning, The 12th International Conference, CA, Morgan Kaufmann, 1995.
- [4] E.Cole, Hackers Beware, New Riders Publishing, 2002.

- [5] http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html.
- [6] S.Das and D.Lawless, Trustworthy Situation Assessment via Belief Networks, Proceedings of the International Conference Fusion-02, Annapolis, MD, 2002, pp. 543-549.
- [7] S.Das, R.Grey, and P.Gonsalves, Situation Assessment via Bayesian Belief Networks, Proceedings of the International Conference Fusion-02, Annapolis, MD, 2002, pp. 664-671.
- [8] K.Delvin, Logic and Information, Cambridge University Press, 1991.
- [9] J.Dezert and F.Smarandache, On the generation of Hyper-Powersets for the DSMT, Proceedings of the International Conference Fusion-03, Cairns, Australia, 2003, pp. 1118-1125.
- [10] J.Dezert, Foundation for a New Theory of Plausible and Paradoxical Reasoning, ONERA Technical Report RT 1/06769/DTIM, January, 2003.
- [11] M.R.Endsley, D.G.Garland (Eds.), Situation Awareness Analysis and Measurement, Mahwah, NJ: Lawrence Erlbaum, 2000.
- [12] V.Gorodetsky, O.Karsaev, I.Kotenko and V.Samoilov, Multi-Agent Information Fusion: Methodology, Architecture and Software Tool for Learning of Object and Situation Assessment, Proceedings of the 7th International Conference on Information Fusion, Stockholm, Sweden, 2004, pp. 346–353.
- [13] V.Gorodetsky, O.Karsaev, V. Samoilov, V.Konushy, E.Mankov, A.Malyshev, Multi Agent System Development Kit: MAS software tool implementing GAIA Methodology. Z.Shi and Q.He (eds.) Proceedings off the International Conference on Intelligent Information Processing (IIP2004), Beijing, Springer, 2004, pp. 69-78.
- [14] V.Gorodetsky, O.Karsaev, and V.Samoilov, Distributed Learning of Information Fusion: A Multi-agent Approach, Proceedings of the International Conference Fusion 03, Cairns, Australia, 2003, pp. 318-325.
- [15] V.Gorodetsky, O.Karsaev, and V.Samoilov, Multi-agent Technology for Distributed Data Mining and Classification, Proceedings of the IEEE Conference Intelligent Agent Technology (IAT03), Halifax, Canada, 2003, pp. 438–441.
- [16] V.Gorodetsky, O.Karsaev, Mining of Data with Missing Values: A Lattice-based Approach, International Workshop on the Foundation of Data Mining and Discovery, Japan, 2002, pp. 151–156.
- [17]. V.Gorodetsky, O.Karsaev, Algorithm of Rule Extraction from Learning Data, Proceedings of the 8-th International Conference "Expert Systems & Artificial Intelligence" (EXPERSYS-96), 1996, pp. 133-138.

- [18] M.L. Hinman. Some Computational Approaches for Situation Assessment and Impact Assessment, Proceedings of the International Conference Fusion-02, Annapolis, MD, 2002, pp. 687-693.
- [19] <http://www.inforfusion.org/>.
- [20] M.Kokar, Situation Awareness: Issues and Challenges, Proceedings of the 7th International Conference on Information Fusion, Stockholm, Sweden, 2004, pp. 535–536.
- [21] G.A.Lambert, Grand Challenges of Information Fusion, Proceedings of the International Conference Fusion-03, Cairns, Australia, 2003, pp. 213-220.
- [22] E.Lefevre, O.Colot, and P.Vannoorenberghe, Belief Function Combination and Conflict Management, Information Fusion Journal, Elsevier, 2002.
- [23] R.Mahler, The Levels 2, 3, 4 Fusion Challenge: Fundamental Statistics, Proceedings of the 7th International Conference on Information Fusion, Stockholm, Sweden, 2004, pp. 537–539.
- [24] C.J.Merz, Combining classifiers using correspondence analysis, Advances in Neural Information Processing, 1997.
- [25] R.Michalski, A Theory and Methodology of Inductive Learning, Machine Learning, vol.1, J.G.Carbonel, R.S.Michalski and T.M.Mitchel (Eds.), Tigoda, Palo Alto, 1983, pp. 83-134.
- [26] M.Morelli, Application of Dempster–Shafer Theory of Evidence to the Correlation Problem, Proceedings of the International Conference Fusion-02, Annapolis, MD, 2002, pp. 759-762.
- [27] S.Northcutt, D.McLachlan, J.Novak, Network Intrusion Detection: An Analyst's Handbook, New Riders Publishing, 2000.
- [28] J.Ortega, M.Coppel, and S.Argamon, Arbitrating Among Competing Classifiers Using Learned Referees, Knowledge and Information Systems, 4, 2001, pp. 470-490.
- [29] A.Prodromidis, P.Chan, S.Stolfo, Meta-Learning in Distributed Data Mining Systems: Issues and Approaches, Advances in Distributed Data Mining, AAAI Press, Kargupta and Chan (eds.), <http://www.cs.columbia.edu/~sal/hpapers/DDMBOOK.ps.gz>, 1999.
- [30] G.Rogova, C.Lollett, and P.Scott, Utility-based Sequential Decision Making in Evidential Cooperative Multi-agent Systems, Proceedings of the International Conference Fusion-03, Cairns, Australia, 2003, pp. 823-830.
- [31] J.Salerno, M.Hinman, D.Boulware, Building a Framework for Situation Assessment, Proceedings of the 7th International Conference on Information Fusion, Stockholm, Sweden, 2004, 219–226.

- [32] J.Scambray, S.McClure, Hacking Exposed Windows 2000, Network Security Secrets, McGraw-Hill, 2001.
- [33] J.Scambray, S.McClure, G. Kurtz, Hacking Exposed, McGraw-Hill, 2000.
- [34] G.Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.
- [35] A.Steinberg, Unification across JDL Data Fusion Levels 1 and 2, Proceedings of the 7th International Conference on Information Fusion, Stockholm, Sweden, 2004, pp. 533-534.
- [36] J.Sudano, A generalized Belief Fusion Algorithm. On the generation of Hyper-Powersets for the DS_mT, Proceedings of the International Conference Fusion-03, Cairns, Australia, 2003, pp. 1126-1132.
- [37] J.Sudano, Inverse Pignistic Probability Transform, Proceedings of the International Conference Fusion-02, Annapolis, MD, 2002, pp. 763–768.
- [38] P.Svensson, Information Fusion as Enabling Technology for Network-based Defense, Proceedings of the 7th International Conference on Information Fusion, Stockholm, Sweden, 2004, pp. 524–525.
- [39] K.Ting, The characterization of predictive accuracy and decision combination, Proceedings of 13th International Conference on Machine Learning, Morgan Kaufman, 1996, pp. 498-506.
- [40] L.Todorovski and S.Dzeroski, Combining classifiers with meta-decision trees, D.A.Zighen, J.Komarowski and J.Zitkov (Eds.), Proceedings of 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-00), France, Springer Verlag, 2000, pp. 54-64.
- [41] A.White, Data Fusion Lexicon. Joint Directors of Laboratories, Technical Panel for C³, Data Fusion Sub-Panel, Naval Ocean Systems Center, San Diego, 1987.
- [42] D.Wolpert, Stacked generalization, Neural Network, 5(2), 1992, pp. 241-260.
- [43] E.Wright, S.Mahoney, K.Laskey, M.Takikawa, and T.Levitt, Multi-entry Bayesian Networks for Situation Assessment, Proceedings of the International Conference Fusion-02, Annapolis, MD, 2002, pp. 804-811.

Figure Captions

Fig.1. Data model and structure of decision making used for on-line update of the situation assessment

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

Fig.3. Anomaly detection system architecture and environment

Fig.4. Explanation of the missingness nature of the situation assessment system input