Intelligent Risk Analysis in Access Control

Boštjan Kaluža, Erik Dovgan, Tea Tušar, Matjaž Gams
Department of Intelligent Systems
Jožef Stefan Institute
{bostjan.kaluza, erik.dovgan, tea.tusar, matjaz.gams}@ijs.si

Abstract

Access control is an important security activity that prevents undesired persons from entering secure buildings or perimeters. The advanced risk analysis presented in this paper enables distinguishing between acceptable and undesired entries based on several entry sensors, such as fingerprint readers, and intelligent methods that learn behavior from previous entries. We have extended the intelligent layer in two ways: first, by adding a meta-learning layer that combines the output of specific intelligent modules, and second, by constructing a Bayesian network to integrate the predictions of learning and meta-learning modules. The obtained results indicate an important increase in detecting security attacks.

1 Introduction

Safety and integrity of buildings and systems have gained importance in the modern world due to terrorist attacks, system intrusions and frauds. An important security issue is to ensure effective access control that prevents unauthorized persons to access specific areas.

Although widely used, access control has certain weaknesses in the real world. Classic security methods fail to recognize unauthorized access if, for example, an identification card is stolen, a fingerprint is faked or an employee is forced to open the door to unauthorized persons. A human supervisor or guard is not able to efficiently control various access points for several hours and can get fooled even by simple tricks. Advanced intelligent access control systems promise to increase performance at reasonable cost.

In this paper we present an intelligent access control system that analyzes the risk of each entry and offers an explanation, both in real time. It upgrades classic access control systems, e.g., biometric or other sensors for identification, with intelligent verification based on user behavior. In the first stage, the system utilizes an arbitrary number of intelligent modules, where each module analyzes user behavior from different viewpoints and performs its own risk analysis. The modules are aggregated into meta-modules in the second stage of learning. In the third stage, the system integrates the analysis of modules and meta-modules and evaluates the overall event probability.

The rest of this paper is structured as follows. Section 2 reviews the related work. The general structure of the proposed system is presented in Section 3, while Section 4 describes the individual modules and final integration in detail. Section 5 presents the experimental evaluation and results. Finally, Section 6 concludes the paper with discussion and summarizes the work done.

2 Related Work

Classic access control can be improved in several ways. In this review, three selected approaches are examined: additional biometric methods (e.g. voice and face recognition), behavior analysis, and combination of different sensors.

The first approach is based on advanced biometric sensors. Wahyudi and Syazilawati [2007] presented verification with speech analysis. They constructed voice-based models for authorized persons and performed identification with adaptive network-based fuzzy inference system. In a similar spirit, Wong and Ho [2009] and Sun and Tien [2008] focused on face recognition. Various facial features were extracted from video, saved into a database and compared with a new entry. The authors report accuracy of over 90 %.

The second approach is focused on behavior analysis of two kinds: analyzing video sequences (e.g. from a surveillance camera), and analyzing transactions and logs. Zhang *et al.* [2007] proposed a system for visual analysis of human motion from a video sequence, which recognizes unusual behavior based on walking trajectories, namely treading tracks. Two types of line shapes were studied : closed curve and spiral line. If somebody's treading track takes on one of these shapes, this person wanders around and is therefore suspicious. Lin *et al.* [2009] described a video surveillance system based on color features, distance features and a count feature, where evolutionary techniques are used to measure the observation similarity. The system tracks each person and classifies their behavior by analyzing their trajectory patterns. This is performed with a hybrid genetic algorithm, which uses a Gaussian synapse.

In contrast to the video-based methods, analyzing transactions and logs detects unwanted attempts at accessing systems mainly through a network. In [Quah and Sriganesh, 2008], an approach to online banking fraud detection based

on users' spending behaviors is presented. It makes use of a self-organization map to learn users' spending patterns, while neural networks are used for filtering unusual events and analyzing user behavior to detect fraud. Furthermore, Alexandre [1997] proposed a system based on behavior recognition of keyboard signature, which is more difficult to copy or fake than fingerprint or smart card. The presented technique implements a neural network, which is evaluated in terms of efficiency and performance.

In the third approach, the outputs of different sensors can be combined using data mining techniques. Lamborn and Williams [2006] introduced an intelligent system which consists of several heterogeneous sensors. The sensors are divided into clusters according to their GPS position using self organizing maps. Outputs from sensors are classified in each cluster and a voting algorithm is used for computing the final classification. Several data mining methods were tested for cluster classification, e.g. k-nearest neighbors, neural networks and support vector machines. A similar system was also presented in [Bontempi and Borgne, 2005].

In summary, the described approaches use state of the art methods that successfully reduce the risk of intrusion. They use additional biometric sensors and behavior analysis as upgrades to classic access control. Our approach makes a step further in combining an arbitrary number of methods in three stages. Similar as in [Lamborn and Williams, 2006], our system constructs situational awareness from different sensors, but in contrast to their method, the outputs of intelligent modules are assembled using meta-learning, on top of which the final reasoning is performed with a Bayesian network. The system is also able to explain the evaluation to the human operator.

3 System Structure

3.1 Functional Description

In order to reduce the risk of intrusion, we have designed a modular system heavily relying on intelligent methods. The aim of the system is to ensure higher security in critical areas, for example, headquarters or political institutions, by detecting irregular accesses or unusual behavior at access points and raising an alarm.

The entry procedure is shown in Figure 1 and proceeds as follows. First, a user is identified. Next, if the identity exists, the user gets verified, which leads to releasing the doorlock in the case of positive outcome. The verification process is performed in two stages, where the first stage is classic biometric verification and the second stage is intelligent verification. Intelligent modules perform entry evaluation and suggest proper action.

The development of our intelligent access control system was based on the following five requirements. First, the system is required to monitor entries and process entries' evaluations in real time. Second, several access points may be monitored at the same time taking into account the knowledge about a user's movement between them. Third, an arbitrary number of sensors and intelligent modules can be used, depending on equipment at specific access points and data availability. Fourth, the system is expected to evaluate an entry and suggest a proper action. Finally, the system should provide an explanation of its evaluation in a user-friendly interactive control panel. In a nutshell, the aim is to create a system that will improve security of entry control and help the operator to control numerous access points effectively.

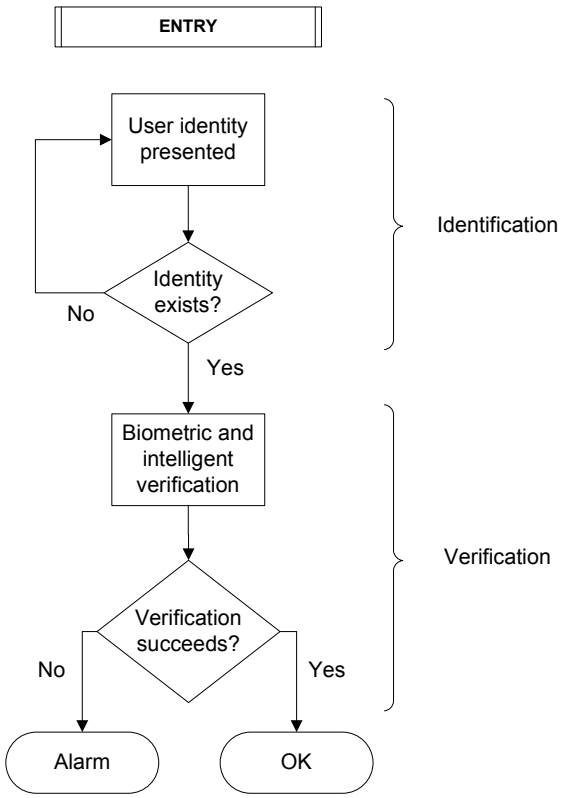

Figure 1: Entry and verification procedure.

3.2 Architecture

The main architectural tasks are collecting data from peripheral devices and sensors, processing and analyzing this data, integrating the analyses into a human-readable form, and displaying them to a user with a suggestion for an appropriate action (Figure 2).

The architecture of the system consists of six basic layers. At the first, hardware layer, data processing starts by gathering data from various sets of sensors at different access points, e.g., biometric sensors, visual sensors or door sensors. The sensors capture data from the environment and pass it to the next layer through a controller. The next layer stores raw data into a database and supports implementation of higher layers. The intelligent layer has three levels consisting of various numbers of intelligent modules and an ontology as a special module for storing and presenting acquired knowledge. Each low-level module applies an intelligent method to a specific data type, e.g. visual data, temporal relations etc. At the next level, some of the modules are gathered in meta-modules. The final output is combined using the integration of modules and meta-modules. The last layer is the application layer. It contains human-readable tools, e.g. re-

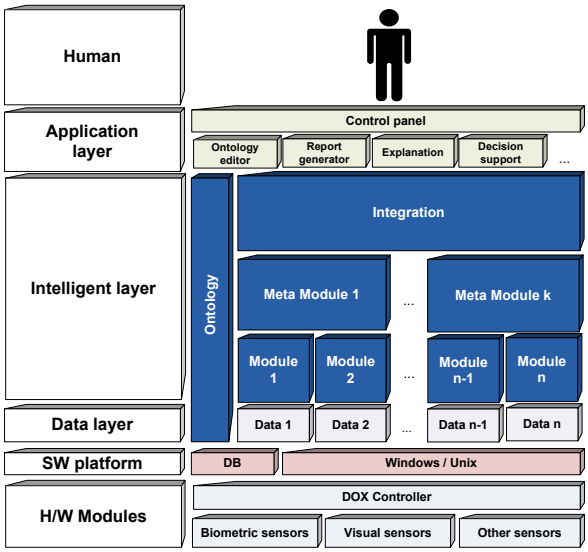

Figure 2: General architecture of the system.

port generator, decision support and explanation, which helps the operator to understand the decisions and to manage entry control points. The tools are collected in a user-friendly control panel. Our major contribution is in the intelligent layer presented in Figure 2 in dark boxes: modules, meta-modules and integration, all the time manipulating data in one central ontology.

3.3 Observing User's Behavior

Each human typically performs activities in a specific way, be it at a micro or macro scale. The behavior of the users in our system is actually monitored at three points of view. From the first viewpoint, denoted as *micro level*, one typically deals with tenths of second or seconds. For example, one user always carries his identity card in a wallet and puts the whole wallet near the wireless identity card reader, while another user carries her card in a handbag and spends some time to take it out, identify herself, and put the card back. The user's movement around the access point depends on his/her habits and mental/physical properties. This facts determine users' patterns at the micro level.

The second viewpoint, denoted as *macro level*, describes daily users' routines. The activities of interests are their arriving time to an access point, movements between different access points in an access control network, and even the dependencies between users, e.g. user A often enters after user B in a short time period. The time scale used in macro level can vary from seconds to months.

The third viewpoint, denoted as *visual level*, captures users' visual movement at an access point using a camera. It is also focused on micro level movements, but in contrast to the micro level, it obtains features from visual characteristics of a user and the movement, e.g. the user's height and door opening dynamics.

Several rules additionally control the regular entry procedure, regular working time, and access permissions.

3.4 Experimental Environment

To design and test our intelligent modules for access control, we have set up an experimental environment, as shown in Figure 3. It consists of a single access point protecting a flat in a building. The access point is equipped with a camera (on the ceiling), a card reader and a fingerprint reader (on the wall near the door), an electronic lock and an open/close sensor on the door. The input signals are collected with a multi-channel access controller, which can be connected to various peripheral devices.

Figure 3: Prototype access point configuration (camera view). The task is to detect suspicious entries of persons, e.g., under influence of drugs or under a gun threat outside the camera field.

When a user passes the access point, four times are registered:

- t_c – time of card reader acceptance
- t_f – time of fingerprint reader acceptance
- t_{do} – time of door opening,
- t_{dc} – time of door closing.

The data is collected and written into the ontology for additional processing by six intelligent modules. The first module, denoted as *expert rules*, detects prohibited and basic undesired behavior. It uses SWRL rules for querying the system ontology (see Section 4.1). The second module, *micro learning*, learns patterns of user behavior during entry at the micro level. The learning is performed with local outlier detection (LOF) method (described in more detail in Section 4.2). The three *macro learning* modules learn the access patterns at the macro level and are combined at a meta-level (see Section 4.3). The last module, *visual learning*, uses histograms of optical flow for detecting behavior at the visual level (see Section 4.4).

Each module performs its own risk analysis of an entry and returns an evaluation with explanation. The meta-module uses basic weighted voting upon decisions of single modules, while the integration module accepts classifications of modules as observations and performs reasoning with a Bayesian network.

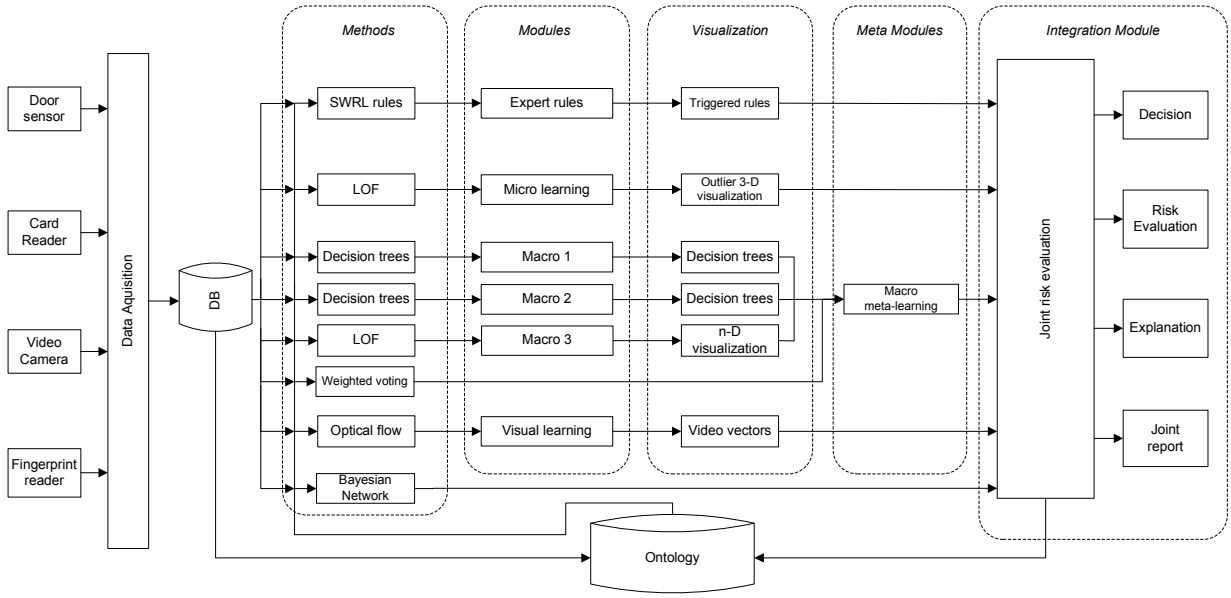

Figure 4: Information flow in the implemented platform.

According to the final probability, the entry is classified in one of the two classes: *OK*, if the entry is regular, and *alarm*, if the entry is irregular. Evaluations and explanations of each module are stored into the system ontology. The platform is presented in Figure 4.

3.5 Ontology

Various methods use the same or similar data at complex levels. Besides simple relationship between classes, complex representations are also required. For example, a sensor *belongs to* an access point. Therefore, the Web Ontology Language (OWL) was used [Horrocks *et al.*, 2003]. For the presented system data storage, the program $Protégé$ OWL was used, which, besides powerful data storage language, presents data in a user-friendly way [Protégé, 2009].

The ontology consists of a central part, including event data and its classifications, and several local parts, each of them storing the knowledge of a particular module. The central part includes information about:

- access points: position, security requirements etc;
- persons: personal details, position in a company, rooms of the building where a person has permission to enter etc.;
- sensors: type, e.g. biometric sensor, access point where the sensor is positioned;
- events: person who produced the event, access point where it was produced, sensors which sensed the event, each module's classification and the final classification, and actions that can be performed due to the evaluation.

The ontology structure enables that modules process independent data and enabling data independency between modules. Therefore, new sensors, modules or access points can be easily added to the system.

4 Modules and Algorithms

This section describes the modules and algorithms in more detail. In this particular implementation, we prefer algorithms with the ability to provide as much explanation as possible, but in general, it is possible to select any learning algorithm.

4.1 Expert Rules

The first module consists of expert rules that are defined by a security expert or a human operator. The rules do not learn from past user behavior. Each rule has its adjustable parameters enabling the operator to create a new rule by specifying rule parameter values. Rules are described in the SWRL language [W3C, 2004] for querying data stored in OWL. The test over the events is performed by the Jess rule engine [Friedman-Hill, 2009].

We have implemented two types of rules. If the entry procedure is violated, the first type of rules trigger an alarm independently of other modules. The second type of rules refer to the entry observation, e.g. *"The user accessed this area more than 5 times in the last two minutes"*. Instead of unconditionally triggering an alarm, each triggered rule R_i returns probability $p(R_i)$ that the entry is regular. If several second-type rules $R_1, \ldots, R_n$ are triggered, then $min(p(R_1), \ldots, p(R_n))$ is returned and the module composes an explanation consisting of the violated rule and its parameters.

4.2 Micro Learning

The micro learning module learns short-time behavior. Attributes are calculated as three time differences from four input times:

$$\Delta t_1 = t_f - t_c \quad (1)$$

$$\Delta t_2 = t_{do} - t_f \quad (2)$$

$$\Delta t_3 = t_{dc} - t_{do} \quad (3)$$

Each entry e_i is thus presented with a triple $e_i = (\Delta t_{i,1}, \Delta t_{i,2}, \Delta t_{i,3})$. All regular entries of a particular user form a learning set $E = \{e_1, e_2, \ldots, e_n\}$. When the user produces a new entry e_{n+k}, the module compares it with the learning set E and returns an outlier factor: if the new entry is similar to the existing entries, e_{n+k} is a regular entry with a low outlier factor, otherwise, it is an outlier with a high outlier factor.

In [Tusar and Gams, 2006] we examined various algorithms for outlier detection, selected LOF (Local Outlier Factor) [Breunig, 2001] and implemented it. The algorithm reportedly achieves reliable performance where instances are not uniformly distributed in the attribute space. The LOF for a new entry e_i is defined as

$$LOF_k(e_i) = \frac{1}{|ngb_k(e_i, E)|} * \sum_{a \in ngb_k(e_i,E)} \frac{ldns_k(a)}{ldns_k(e_i)} \quad (4)$$

where $ngb(e_i, E)$ is the set of $k \in E$ nearest neighbors of an instance e_i, and $ldns_k(a)$ is the local density of an instance a and its k nearest neighbors. Intuitively, $LOF_k(e_i) \leq 1$ when the new instance is near an existing cluster E, and $LOF_k(e_i) > 1$ when the instance is far from the cluster.

The final output of the module are the LOF value, the probability that the entry is regular, and a visual explanation. The probability is computed from the LOF value by the following procedure. Let $t_l < 1$ denote the threshold value for regular entries and let $t_u > 1$ denote the threshold value for irregular entries. Then, the probability p that the entry is regular is computed as a linear combination of the threshold values:

$$p = \begin{cases} 1.0 & \text{if } LOF \leq t_l \\ 0.0 & \text{if } LOF \geq t_u \\ \frac{t_u - LOF}{t_u - t_l} & \text{otherwise} \end{cases} \quad (5)$$

Since the module is using only three micro attributes, its visualization can be presented in a 3-dimensional space, one dimension for each attribute. Entries are thus presented as points and the LOF value of each point is presented with color: from red for outliers through yellow for unclear entries, up to the green for entries in the cluster. Figure 5 shows a cluster of entries in a learning set E (circles) and a new entry e_i (a plus).

4.3 Macro Learning and Meta-learning

The data gathered at the macro level are used in three modules. Two of them also exploit the data derived from the micro level. The macro level attributes are divided in two groups describing a current entry and relation between the current entry and previous entries, respectively. The attributes from the first group are, for example, current time and date, day of the week, date in relation to the month (i.e. second Friday in the month). The second group defines relations such is the number of previous entries in the same day (for the current user), the user who entered previously in a specific time interval, the time of entry at the same day previous week etc.

The first macro module learns only from macro attributes. Positive learning examples are the regular entries of a user, while negative learning examples are the irregular entries of the user and entries of other users. Several machine learning algorithms were tested and finally decision trees were selected, Weka's J48 implementation of C4.5 in particular [Witten and Frank, 2005]. The main benefit of decision trees is their ability to explain decision after classification occurs. The path leading from the root to the chosen leaf is colored according to the classification - green for regular entries and red for alarms. The distribution of target variable in the chosen leaf is interpreted as the probability that the entry is regular.

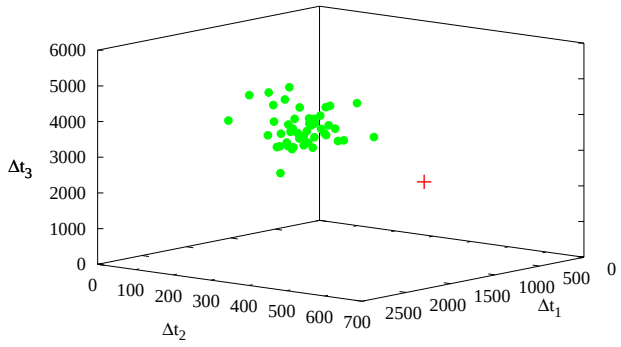

Figure 5: Regular entries of a particular person (circles) and a new entry denoted as an outlier ('+').

The second macro module applies the same algorithm as the previous module, but uses both micro and macro attributes. While the first macro module considers only behavior on macro level and discovers patterns, for example, *"User X comes to work on Mondays between 8.15 and 8.40 (93 %)"*, the second macro module refines these patterns by incorporating micro attributes.

In the third macro module, the macro and micro attributes are used for learning with the LOF algorithm. In contrast to the micro module, where visualization was intuitive, the high number of attributes requires a different representation. For this purpose, we implemented visualization with parallel coordinates. Each attribute is presented on one vertical axis ranging from the minimal to the maximal normalized value. Each entry is thus presented as a broken line intersecting coordinates at its attribute value. The line is colored according to the entry's LOF value: green for regular entries, yellow for unclear entries and red otherwise. Figure 6 shows a cluster of entries in the learning set and a new entry as a dotted line.

At the end, the macro meta-module combines the classifications of all three macro modules. In the tested prototype only weighted voting was implemented due to lack of time; however, several meta-level learning algorithms are already developed. Also, in the tested implementation, only the macro meta-learning was applied, but in principle, arbitrary subgroups of modules could be connected using meta-learners. All results and visualizations are written into the ontology.

4.4 Visual Learning

The visual learning module learns patterns of a user's movement in front of an access point from video and classifies a

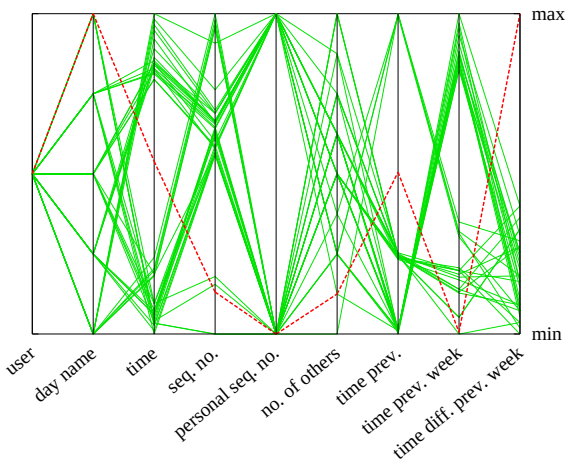

Figure 6: Multi-dimensional presentation of regular entries (thin lines) and a new entry (dotted line) classified as alarm.

new entry as regular or not. For this purpose a web camera with 1.3 Mpixel resolution and 30 fps rate was used.

When a new entry occurs, the last 30 seconds of video are analyzed in the following steps. First, the histograms of optical flow are computed and divided in six segments, representing an approximation of body parts. Next, in each segment the prevailing movement is estimated and transferred into a sequence of symbols. This sequence defines the digital signature of movement and is used for verification. Each user has a learning set of valid regular entries, which are used for comparison with new entry signatures. Finally, the module outputs the classification and probability that the entry is regular as a normalized result from comparison. More about this method can be found in [Pers *et al.*, 2007].

It should be noted that other sensor analysis such are speech or walking patterns could be added as well.

4.5 Integration

After the expert rules, micro, macro, visual and meta-learning have made their assessments, their results are integrated into a joint risk analysis of the current entry. It estimates the probability of the event E = *entry is regular* given the observations of modules. If the estimated probability does not exceed a threshold value, an alarm is triggered.

The reasoning in the prototype system is performed with a Bayesian network, structured as shown in Figure 7. Four modules have a direct impact on the event E, namely expert rules, micro learning and visual learning, and macro meta-learning module, while the macro meta-learning module depends only on the three macro modules. Probabilities in the network are computed from the test data, using the *m-estimate* for conditional probabilities and the *Laplace estimate* for a priori probabilities.

The integration proceeds in three steps. Firstly, output from each module is converted to interval $[0, 1]$ presenting a posterior probability p_{M_i} that entry is regular. Secondly, given the Bayesian network N and probabilities p_{M_i}, the estimated probability of an event E is computed from the network.

Finally, the integration module outputs the joint analysis

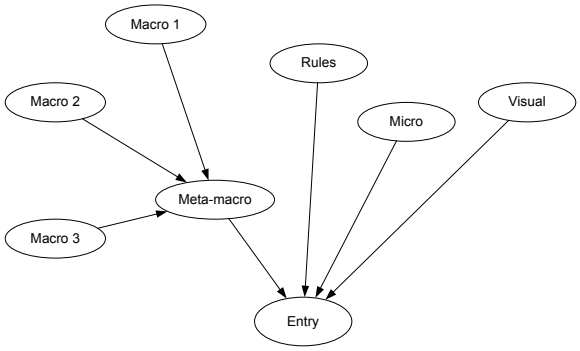

Figure 7: Bayesian network used for reasoning.

as a probability that the entry is regular and provides an explanation. According to the threshold values, the integration module triggers *alarm* or *OK* and stores the results into the ontology. In high-security areas, the cost of a false alarm is negligible compared to the cost of unrecognized intruder, therefore the system is set to minimize the latter.

5 Experimental Results

Experimental verification was performed in the prototype environment as described in Section 3.4. It consisted of two phases: learning and evaluation. In this paper we report about one learning and three evaluation experiments.

In the learning phase, four people were recorded accessing the system. Each individual completed 40 regular entries that were used as positive learning examples. Negative learning examples for one individual were the entries of the other four people. After the learning was completed, the system was ready to operate.

In the evaluation phase, we performed three experiments: two with simulated entries and one real-time experiment with security experts. The first two experiments were performed off-line with simulated tests. The focus was on *fake identity* scenario, where we recorded regular entries of four people in the role of an employee (the system already knew them) and three people in the role of an intruder (new to the system). First, each user made 31 regular entries serving as testing examples. Afterwards, we multiplied these examples by assigning different identities to the existing entry. Two datasets were constructed: in the first dataset, only the identities of the employees were swapped, while the second dataset consists of regular entries of employees and irregular entries of intruders who disguised themselves as employees. Each dataset had 496 examples with distribution of 75 % negative examples.

Both experiments were tested without the visual learning since it did not allow testing in the off-line mode. Consequently, the Bayesian network for integration was slightly changed omitting the visual learning module. The experiment was run on already learned and tuned modules from the first phase, while the probabilities in the Bayesian network were obtained with 10-fold-cross validation.

The performance of the system and modules for the first dataset is presented in Table 1. The first two columns represent irregular entries, where the identity of the employees

was swapped, and regular entries with the correct identity of the employees. Each number denotes accuracy, e.g. the left most number represents percentage of irregular entries that were predicted as regular by expert rules. The system overall achieved 95.77 % accurate performance. The expert rules always predicted *OK*, because all entries were regular according to the entry procedure. Micro learning performed well in detecting both irregular and regular entries, while macro learning made more mistakes. The high accuracy of the micro module was expected because it is rather easy to distinguish movement of a couple of people given sufficient learning examples.

	Scenarios			
	Irregular entries		Regular entries	
Modules	*OK*	*alarm*	*OK*	*alarm*
Expert rules	1.0	0.0	1.0	0.0
Micro learning	0.06	0.94	0.93	0.07
Macro learning	0.16	0.84	0.83	0.17
Integration	0.01	0.99	0.86	0.14

Table 1: System and module performance for the off-line *fake identity* experiment with four employees only.

Measurements on the second dataset are shown in Table 2. The system was 96.57 % accurate. In contrast to the results from Table 1, where macro learning classified 16 % false positives, the number of false positives in Table 2 is only 2 %. The trend in the micro learning is just the opposite, however, the overall accuracy is comparable in both datasets. The decline in the micro learning performance was expected, since it is more difficult to classify new, unseen behavior than to distinguish between the known cases.

	Scenarios			
	Irregular entries		Regular entries	
Modules	*OK*	*alarm*	*OK*	*alarm*
Expert rules	1.0	0.0	1.0	0.0
Micro learning	0.22	0.78	0.93	0.07
Macro learning	0.02	0.98	0.82	0.17
Integration	0.0	1.0	0.86	0.14

Table 2: System and module performance for the off-line *fake identity* experiment with four employees and three intruders.

In the third, most relevant experiment we invited security experts from the Slovenian Ministry of Defense to test the system by on-line simulation of different security attacks. For the purpose of scientific experimentation, the following eight scenarios were tested and executed on-line by the experts:

1. regular entry: a user enters normally;
2. unusual time: the time of access is out of normal working hours or on a non-working day;
3. multiple entries: a user regularly accesses the secure room several times in a short period of time;
4. unusual behavior: a user is under threat or in a strange state of mind;
5. tailgating: two persons access the secure room with one identity;
6. burglary: an attacker disables hardware protection by force;
7. fake identity: an attacker accesses the secure room with stolen identity card and forged fingerprint;
8. kidnapping: an attacker forces an employee to enable access to the secure room.

Each scenario was imitated several times by different users and in a different order, as desired by the security experts. In total, there were 45 irregular entries and 15 regular entries. The results described in Table 3 are separated in two groups: regular entries (scenario 1) and irregular entries (scenarios 2-8). The numbers show the percentage of test examples that were classified as *OK*, *alarm* or *failed* by the corresponding module. The classification may fail due to disabling of sensors (burglary scenario).

	Scenarios				
	Irregular entries			Regular entries	
Modules	*OK*	*alarm*	*failed*	*OK*	*alarm*
Expert rules	0.84	0.16	0.0	1.0	0.0
Micro learning	0.0	0.89	0.11	0.93	0.07
Macro learning	0.0	0.89	0.11	0.87	0.13
Visual learning	0.08	0.88	0.04	0.73	0.27
Integration	0.0	1.0	0.0	0.87	0.13

Table 3: System and module performance for experiments with four employees and four security experts in a role of intruder.

The system achieved overall accuracy 96.75 %, identifying all irregular entries and being too suspicious in two regular entries. Once again, the expert rules classified with low accuracy (37.0 %), but when an entry was classified as an alarm, it was indeed so. They were also robust in contrast to the other modules, which failed to recognize the burglary scenario. Micro learning and macro meta-learning modules recognized irregular entries with the same accuracy, but macro meta-learning made more mistakes when classifying regular entries. It should be noted that all tests were performed within two hours, which is not well suited for macro learning. Visual learning was a bit more robust than the learning modules, but achieved lower accuracy.

6 Discussion and Conclusion

We have designed a modular intelligent system for analyzing risk at an access point. The system in general combines an arbitrary number of intelligent modules on top of an existing access control. The emphasis is on modeling regular user behavior and estimating the risk that a new entry is not regular based on meta-learning and integration.

In practical evaluation, we presented three experiments indicating encouraging results. As observed, each module has its own strong and weak points. The advanced combination and integration overcomes the particular weaknesses and combines different aspects into a reliable risk evaluation. For

example, if we had used only the best module (micro learning) in the third experiment, the achieved accuracy would be 90.0 %, while the default accuracy (rather meaningless) was 75 %. The accuracy of the integrated system was 96.75 %.

In each system, there is a fine line between being too sensitive or not sensitive enough to small changes in behavior. Although some of the methods, e.g. the Bayesian network are quite robust, practical application needs some fine-tuning of system parameters. One of the first major benchmarks painfully reminded us of the difference between laboratory and field test. Namely, one of early versions of the system was able to successfully distinguish between normal users, but security experts found a way to trick the intelligent modules. Only after incorporating some modifications, the system was able to cope with human expertise as presented in Table 3.

One of the drawbacks of the system is that it requires a learning procedure: the system can be used only after a certain amount of regular accesses have been made. Furthermore, if a person changes behavior, e.g. due to an injury, the learning must start anew. Further work on the system includes a mechanism for continuous learning and adaptation to the user through time.

The complex methods implemented seem to be an overload for a simple commercial application. In the current form the system is specialized for high-security areas. Namely, the joint verification methods turned out to be very hard to bypass. One method can be fooled relatively easily, while deceiving different methods inside the normal time interval is a much harder task.

In summary, intelligent risk analysis at an access point presents an improvement in terms of risk analysis and has a potential to demonstrate that in reality.

Acknowledgment

The project has received founding partly from Slovenian Ministry of Defense (MORS) and partly from Slovenian Research Agency (ARRS). The authors also thank the members of the department for helping us with research, in particular Jana Krivec, Robert Blatnik and Aleš Tavčar, and to the security experts and supervisors.

References

[Alexandre, 1997] Thomas J. Alexandre. Biometrics on smart cards: an approach to keyboard behavioral signature. *Future Generation Computer Systems*, 13(1):19–26, 1997.

[Bontempi and Borgne, 2005] G. Bontempi and Le Borgne. An adaptive modular approach to the mining of sensor network data. In *Proceedings of the Workshop on Data Mining in Sensor Networks, SIAM SDM*, Newport Beach, CA, April 2005.

[Breunig, 2001] M. Breunig. *Quality Driven Database Mining*. PhD thesis, University of Munich, 2001.

[Friedman-Hill, 2009] Ernest Friedman-Hill. Jess, the rule engine for the java platform. *http://www.jessrules.com*, 2009.

[Horrocks *et al.*, 2003] Ian Horrocks, Peter F. Patel-schneider, and Frank Van Harmelen. From shiq and rdf to owl: The making of a web ontology language. *Journal of Web Semantics*, 1:2003, 2003.

[Lamborn and Williams, 2006] Peter Lamborn and Pamela J. Williams. Data fusion on a distributed heterogeneous sensor network. In *Proc. SPIE, Vol. 6242*, pages 1–8, Orlando, FL, USA, April 2006.

[Lin *et al.*, 2009] L. Lin, Y. Seo, M. Gen, and R. Cheng. Unusual human behavior recognition using evolutionary technique. *Computers and Industrial Engineering*, 56(3):1137 – 1153, 2009.

[Pers *et al.*, 2007] Janez Pers, Matej Kristan, Matej Perse, and Stanislav Kovacic. Motion based human identification using histograms of optical flow. In *CVWW 2007: proceedings of the 12th Computer Vision Winter Workshop*, pages 19–26, Graz, Austria, February 2007.

[Protégé, 2009] Protégé. Open source ontology editor and knowledge-base framework. *http://protege.stanford.edu*, 2009.

[Quah and Sriganesh, 2008] Jon T. S. Quah and M. Sriganesh. Real-time credit card fraud detection using computational intelligence. *Expert Systems with Applications*, 35(4):1721–1732, 2008.

[Sun and Tien, 2008] T. H. Sun and F. C. Tien. Using backpropagation neural network for face recognition with 2d + 3d hybrid information. *Expert Systems with Applications*, 35(1-2):361–372, 2008.

[Tusar and Gams, 2006] T. Tusar and M. Gams. Outlier detection in an access control system (in slovene). In *Proceedings of the 9th International multiconference Information Society - IS 2006*, pages 136–139, Joef Stefan Institute, Ljubljana, Slovenia, October 2006.

[W3C, 2004] W3C. Swrl: A semantic web rule language combining owl and ruleml. *http://www.w3.org/Submission/SWRL*, 2004.

[Wahyudi and Syazilawati, 2007] W. A. Wahyudi and M. Syazilawati. Intelligent voice-based door access control system using adaptive-network-based fuzzy inference systems (anfis) for building security. *Journal of Computer Science*, 3(5):274–280, 2007.

[Witten and Frank, 2005] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann, June 2005.

[Wong and Ho, 2009] J. Wong and S. Y. Ho. A local experts organization model with application to face emotion recognition. *Expert Systems with Applications*, 36(1):804–819, 2009.

[Zhang *et al.*, 2007] Y. Zhang, X. J. Zhang, and Z. J. Liu. Irregular behavior recognition based on two types of treading tracks under particular scenes. In *Proceedings of the Second International Conference KSEM 2007*, pages 508–513, Melbourne, Australia, November 2007.