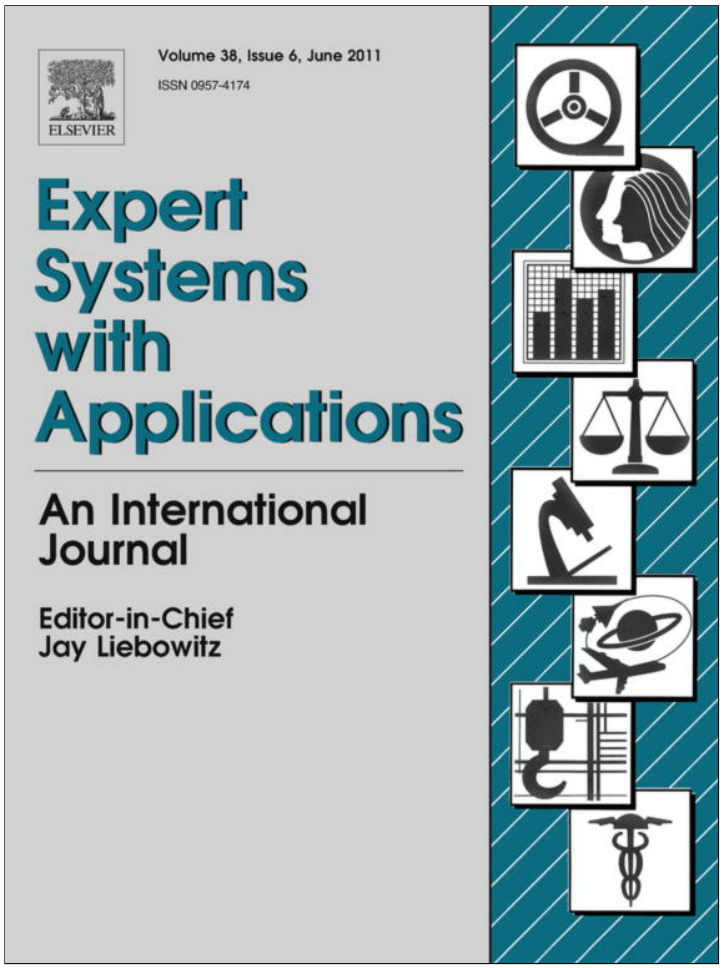

Expert Systems with Applications 38 (2011) 6696–6704

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A probabilistic risk analysis for multimodal entry control

Boštjan Kaluža [a,*], Erik Dovgan [a], Tea Tušar [a], Milind Tambe [b], Matjaž Gams [a]

[a] Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[b] Teamcore Research Group, University of Southern California, 3737 Watt Way, Los Angeles, CA 90089-0781, USA

ARTICLE INFO

Keywords:
Entry control
Verification
Risk analysis
Multi-layer learning
Machine learning
Behavior modeling
Data fusion

ABSTRACT

Entry control is an important security measure that prevents undesired persons from entering secure areas. The advanced risk analysis presented in this paper makes it possible to distinguish between acceptable and unacceptable 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 the learning and meta-learning modules. The obtained results represent an important improvement in detecting security attacks.

1. Introduction

The safety and integrity of buildings and systems have become more important due to the increased threat of terrorist attacks, system intrusions and frauds. An important security requirement is to ensure effective entry controls that prevent unauthorized persons from accessing specific areas.

The general approach is to combine a two-stage security check: the identification stage, where the user introduces his/her identity; and the verification stage, based on a password and/or one or more signals derived from physical traits, such as fingerprint, voice, iris or written signature. Although widely used, entry control has certain weaknesses in the real world. Classic security methods fail to recognize an 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. Furthermore, a human supervisor or guard is not able to effectively control a variety of access points for several hours and can be fooled even by simple tricks. However, advanced, intelligent, access-control systems offer the promise of improved performance at a reasonable cost.

A common practice in most reported studies is to improve the two-stage security by (1) using advanced biometric methods (Sun & Tien, 2008; Wahyudi & Syazilawati, 2007; Wong & Ho, 2009), (2) analyzing behavior (Alexandre, 1997; Depren, Topallar, Anarim, & Ciliz, 2005; Lin, Seo, Gen, & Cheng, 2009; Quah & Sriganesh, 2008; Stephen & Petropoulakis, 2005; Wilson, 2006; Zhang, Zhang, & Liu, 2007) or (3) using multiple sensors in order to combine them into a single, reliable estimation (Bontempi & Borgne, 2005; Fierrez-Aguilar, Garcia-Romero, Ortega-Garcia, & Gonzalez-Rodriguez, 2005; Lamborn & Williams, 2006). In all the studies referenced above, the methods successfully reduced the risk of intrusion, although each approach was focused on one specific viewpoint. However, recent research efforts have focused on meta-learning (Brazdil, Giraud-Carrier, Soares, & Vilalta, 2009; Vilalta & Drissi, 2002; Wang, 1997). The basic objective is to take into consideration various aspects and hypothesis about an event and the environment in order to construct a situational awareness and then, on this basis, make a reliable risk estimation.

This paper presents a hierarchical framework for an intelligent, probabilistic, risk analysis in access control offering a real-time evaluation and explanation. The framework upgrades classic access-control systems with an arbitrary number of sensors, e.g., biometric or other sensors for identification, and intelligent verification based on user behavior. In the first stage an arbitrary number of intelligent modules is utilized, where each module analyses the user behavior from different viewpoints and performs its own risk analysis. In the second stage of learning the modules are aggregated into meta-modules. The analyses of the modules and the meta-modules are integrated into the third stage and the overall event probability is evaluated. The basic assumptions of this approach are that (1) user behavior rarely changes significantly over time and (2) combined methods are much harder to bypass than a single sensor or method.

The rest of this paper is structured as follows. Related work is described in detail in Section 2. The general structure of the proposed framework is presented in Section 3, while Section 4 describes the individual modules and the final integration in detail. Section 5 presents the experimental evaluation and results with an on-line adversary test. Finally, Section 6 summarizes the work done and concludes the paper with a discussion.

* Corresponding author. Tel.: +386 1 477 3944; fax: +386 1 477 3131.
E-mail address: bostjan.kaluza@ijs.si (B. Kaluža).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.11.065

2. Related work

Research efforts dedicated to enhanced security in access control can be classified into a few large groups. In this review we examine three selected approaches: advanced biometric methods (e.g., voice and face recognition), behavior analysis, and combinations of various sensors.

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

The second approach is focused on behavior analyses 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 a 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: the closed curve and the 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 that 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. Quah and Sriganesh (2008), for example, presented an approach to online-banking fraud detection based on users' spending behaviors. Their approach makes use of a self-organization map to learn users' spending patterns, while neural networks are used for filtering any unusual events and analyzing the user behavior in order to detect fraud. In addition, Alexandre (1997) proposed a system based on the behavior recognition of a keyboard signature, which is more difficult to copy or fake than a fingerprint or a 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 that consists of several heterogeneous sensors. These sensors are divided into clusters according to their GPS location using self-organizing maps. The outputs from the sensors are classified into each cluster and a voting algorithm is used for to compute the final classification. Several data-mining methods were tested for the cluster classification, e.g., *k*-nearest neighbors, neural networks and support vector machines. A similar system was presented by Bontempi and Borgne (2005). In addition, Fierrez-Aguilar et al. (2005) introduced the idea of exploiting user-specific parameters in multimodal biometrics. They proposed an adapted learning scheme that consisted of local learning (user-dependent) and global learning (user-independent), and both results were fused with weighted voting. The authors reported that the adapted learning outperformed the results from single learning.

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 analyses as upgrades to classic access control. Our approach is a further step in combining an arbitrary number of methods in three stages. Similar to Lamborn and Williams (2006), our system constructs a situational awareness from different sensors, but in contrast to their method, the outputs of the intelligent modules are assembled using meta-learning, on top of which the final reasoning is performed with a Bayesian network. In addition, the intelligent modules utilize both user-specific parameters and global knowledge in a similar way to Fierrez-Aguilar et al. (2005), but the last integration is fused proficiently. Finally, the system is also able to explain the evaluation results to a human operator and helps him/her to understand the situation.

3. Hierarchal multimodal framework

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

3.1. Functional description

The entry procedure is shown in Fig. 1 and is as follows. First, a user is identified. Next, if his/her identity exists, the user becomes verified, which leads to the door-lock being released in the case of a positive outcome. The verification process is performed in two stages: the first stage is a classic biometric verification, and the second stage is an intelligent verification. Intelligent modules perform the entry evaluation and suggest the proper action.

The development of our intelligent access-control system was based on the following five requirements. First, the system is

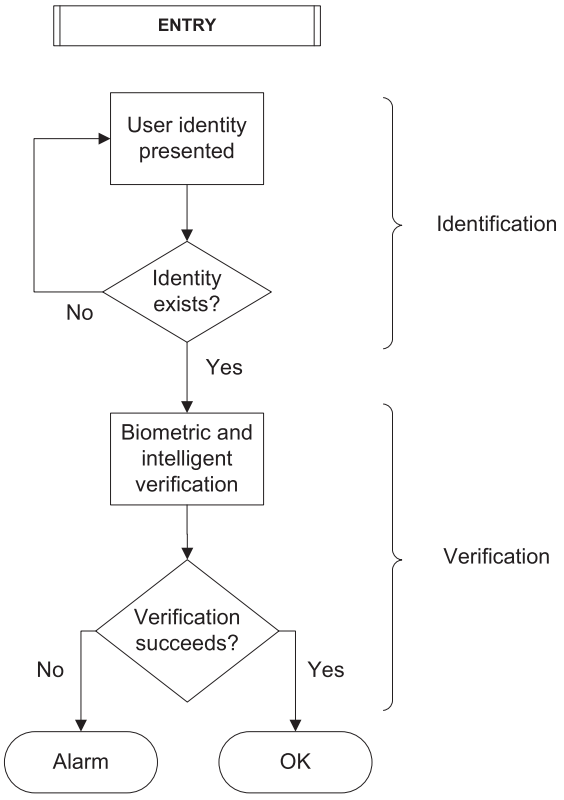

Fig. 1. Entry and verification procedure.

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

3.2. Architecture

The main architectural tasks are collecting the data from the 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 (Fig. 2).

The architecture of the system is designed in six basic layers. In the first, hardware layer, the data processing starts with gathering the data from various sets of sensors at different access points, e.g., biometric sensors, visual sensors or door sensors. The sensors capture the data from the environment and pass it onto the next layer through a controller. The next layer stores the raw data in a database and supports the 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 the acquired knowledge. Each low-level module applies an intelligent method to a specific data type, e.g., visual data, temporal relations, etc. In 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, which contains the human-readable tools, e.g., the report generator, the decision support and the explanation, which helps the operator to understand the decisions and to manage the entry-control points. The tools are collected in a user-friendly control panel. Our major contribution is in the intelligent layer presented in Fig. 2, in the gray boxes: the modules, the meta-modules and the integration, all the time manipulating the data in a single central ontology.

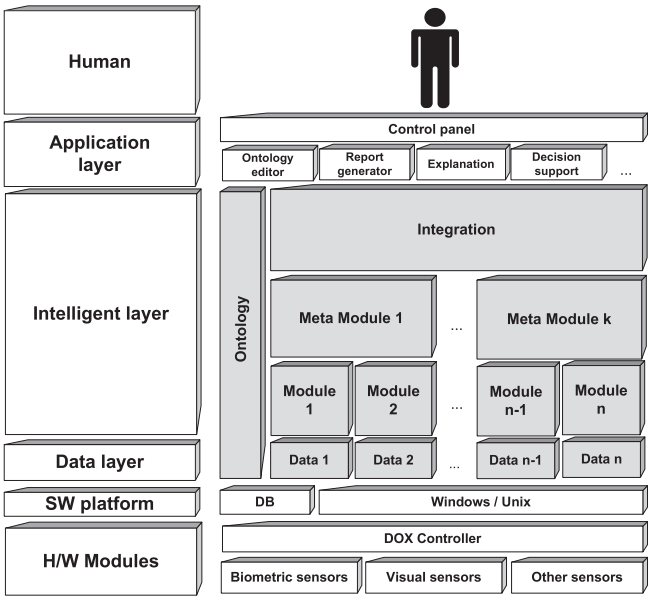

Fig. 2. General architecture of the system. Our contribution is in gray.

3.3. Observing the user's behavior

Each human tends to perform activities in a specific way, be it on the micro or macro scale. However, the behavior of the users in our system is actually monitored from three different points of view. In the first of these, denoted as the *micro level*, one typically deals with behavior that changes in tenths of a 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 requires 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 state. These facts determine the users' patterns at the micro level.

The second viewpoint, denoted as the *macro level*, describes the users' daily routines. The activities of interest are the arrival times at access points, the movements between various access points in the access-control network, and even the connections between users, e.g., user *A* often enters a short time after user *B*. The time scale used at the macro level can vary from seconds to months.

The third viewpoint, denoted as the *visual level*, captures the users' visual movement at an access point using a camera. It is also focused on micro-level movements, i.e., behavior that changes over a short time interval, but in addition to micro-level features, it obtains features from the visual characteristics of the user and his/her movement, e.g., the user's height and the door-opening dynamics.

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

3.4. Experimental environment

To design and test our intelligent modules for access control, we set up the experimental environment shown in Fig. 3, which consists of a single access point protecting an office 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. Such controllers can be

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

dynamically combined in order to ensure the centralized data management of sensors covering a complete access-control network. However, for our purpose, one controller was sufficient.

When a user passes the access point, four different 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 the *expert rules*, detects prohibited and basic undesired behavior. It uses SWRL rules to query the system ontology (see Section 4.1). The second module, *micro learning*, learns the patterns of user behavior during the entry at the micro level. The learning is performed with a local outlier-detection method (LOF) (described in more detail in Section 4.2). The three *macro-learning* modules learn the access patterns at the macro level and are then combined at the meta-level (see Section 4.3). The last module, *visual learning*, uses histograms of optical flow to detect the behavior at the visual level (see Section 4.4).

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

Based on the final probability, the entry is classified into one of the classes: *OK*, if the entry is regular, and *alarm*, if the entry is irregular. The evaluations and explanations of each module are stored in the system ontology. The platform is presented in Fig. 4.

3.5. Ontology

The modules and methods use the same or similar data while processing, and therefore a comprehensible presentation is required. Besides the basic relationships between pieces of information, e.g., the sensor's value, complex representations are also required, for example, a sensor *belongs to* an access point.

We have developed an ontology using the Web Ontology Language (OWL) (Horrocks, Patel-schneider, & Harmelen, 2003) and the ontology editor *Protégé* (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 that 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 that sensed the event, each module's classification and the final classification, and actions that can be performed due to the evaluation.

The ontology structure ensures a knowledge of the system and its setting in a flexible presentation. This means that new sensors, modules and 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 of an 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. These rules do not learn from past user behavior. Each rule has its own adjustable parameters, enabling the operator to create a new rule by specifying the rule-parameter values. The rules are described in the SWRL language

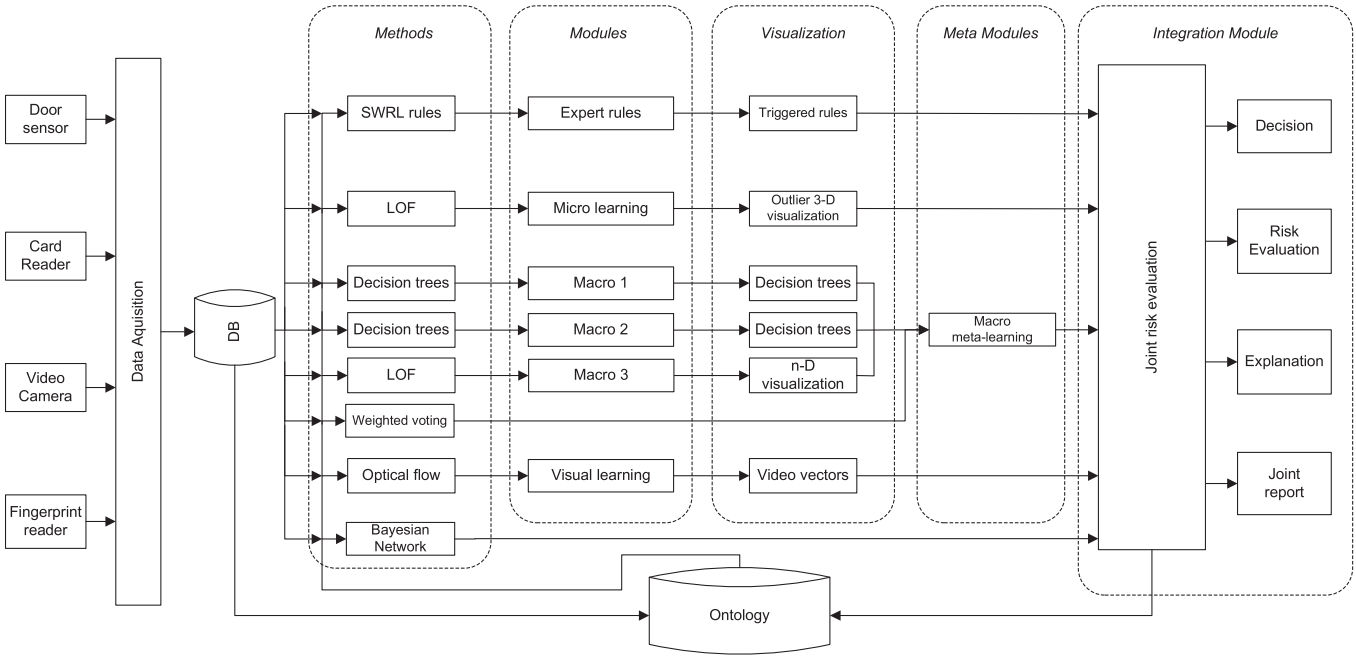

Fig. 4. Information flow in the implemented platform.

(W3C, 2004) for querying data stored in the OWL. A 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 rule triggers an alarm independently of the other modules. The second type of rules refers to the entry observation, e.g., *"The user accessed this area more than five times in the past two minutes"*. Instead of unconditionally triggering an alarm, each triggered rule R_i returns a 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 rules and their parameters. Otherwise, if none of the rules is violated, the entry is, according to the rules, regular, and therefore the returned probability p equals 1.

An example of the second-type SWRL rule is shown in Fig. 5. The rule queries events that occurred between 6 PM and 7 AM and marks these events as alarms, since events are not allowed at night.

4.2. Micro learning

The micro-learning module learns short-term behavior. The 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 represented by a triple $e_i = (\Delta t_{i,1}, \Delta t_{i,2}, \Delta t_{i,3})$. All the 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 our previous work (Tušar & Gams, 2006) we examined various algorithms for outlier detection, selected the 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

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

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, $\mathrm{LOF}_k(e_i) \leqslant 1$ when the new instance is near an existing cluster E, and $\mathrm{LOF}_k(e_i) > 1$ when the instance is far from the cluster.

The final outputs 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 using 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 $p(e)$ that the entry e is regular is computed as a linear combination of the threshold values:

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

Since the module uses only three micro attributes, its visualization can be presented in a 3-dimensional space, with one dimension for each attribute. The entries are thus presented as points, and the LOF value of each point is represented by a color: from red for outliers, through yellow for unclear entries, and onto green for entries in the cluster. Fig. 6 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 into two groups describing a current entry and the relation between the current entry and previous entries. The attributes from the first group are, for example, the current time and date, the day of the week, the date in relation to the month (i.e., the second Friday in the month). The second group defines relations such as 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 on the same day in the previous week, etc. It is important to note that macro learning would be more powerful if we had monitored more than one access point.

The first macro module learns only from macro attributes. The positive learning examples are the regular entries of a user, while the negative learning examples are the irregular entries of the user and the entries of other users. Several machine-learning algorithms

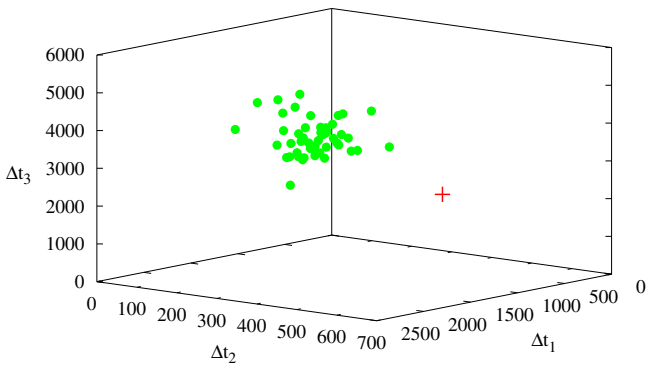

Fig. 6. Regular entries of a particular person (circles) and a new entry denoted as an outlier ('+').

```
event(?event_object) & swrl_end_of_testing(?event_object, ?event_swrl) &
swrlb:equal(?event_swrl, false) & card_time(?event_object, ?time_of_event) &
swrlb:greaterThan(?time_of_event,  "18:00:00") &
swrlb:lessThan(?time_of_event, "7:00:00")
THEN
swrl_rules_result(?event_object, "0.0") &
swrl_rules_explanation(?event_object, ?event_swrl_explanation) &
swrlb:stringConcat(?event_swrl_explanation,
                   "Alarm: event time is between 18:00 and 7:00")
```

Fig. 5. An expert-rule example written in SWRL.

were tested and, finally, decision trees were selected – Weka's J48 implementation of C4.5, in particular (Witten & Frank, 2005). The main benefit of decision trees is their ability to explain a decision after the 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 the target variable in the chosen leaf is interpreted as the probability that the entry is regular. The classification problem was introduced as a verification, where each user has his/her own decision tree with two possible outcomes: true, if the claimed identity is valid, and false otherwise.

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 the behavior at the 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 the visualization was intuitive, the large 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 minimum to the maximum normalized value. Each entry is thus represented as a broken line intersecting the 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. Fig. 7 shows a cluster of entries in the learning set and the new entry as a dotted line.

Finally, the macro meta-module combines the classifications of all three macro modules. Then, all the results and visualizations are written into the ontology. In the tested prototype, only weighted voting was implemented; however, several meta-level learning algorithms were applied later. Also, in the tested implementation, only the macro meta-learning was applied, but in principle, an arbitrary subgroup of modules could be connected using meta-learners.

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

When a new entry occurs, the last 30 s of video are analyzed in the following steps. First, the histograms of optical flow are computed and divided into six segments, representing an approximation of the 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 the movement and is used for the verification. Each user has a learning set of valid regular entries, which are used for comparison with the new entry signatures. Finally, the module outputs the classification and probability that the entry is regular as a normalized result from the comparison. More about this method can be found in Perš, Kristan, Perše, and Kovačič (2007).

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

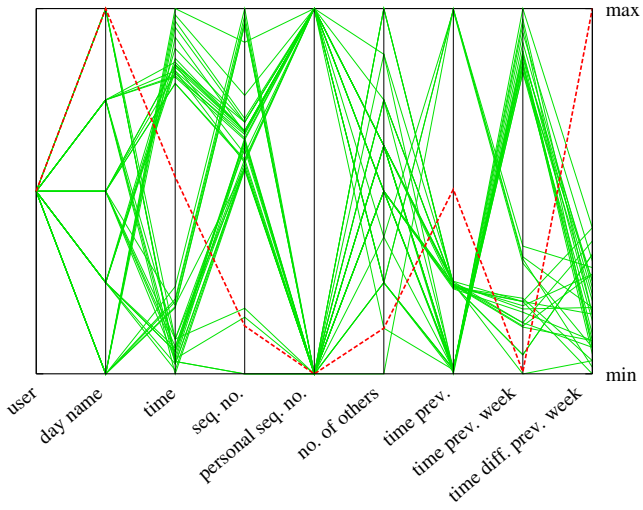

Fig. 7. Multi-dimensional representation of regular entries (thin lines) and a new entry (dotted line) classified as an alarm. There are nine attributes with values normalized between the min and max values.

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 the modules. If the estimated probability does not exceed a threshold value, an alarm is triggered. Note that an alarm can also be triggered by expert rules when there is sufficient certainty (type 1 rules).

The reasoning in the prototype system is performed with a Bayesian network, structured as shown in Fig. 8. Four modules have a direct impact on the event E, i.e., expert rules, micro learning and visual learning, and a macro meta-learning module, while the macro meta-learning module depends only on the three macro modules. The 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, the output from each module is converted to interval the [0,1] representing the a-posterior probability p_{M_i} that the entry is regular. Secondly, given the Bayesian network N and the probabilities p_{M_i}, the estimated probability of an event E is computed from the network.

Finally, the integration module outputs the joint analysis 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 in the ontology. In high-security areas, the cost of a false alarm is negligible compared to the cost of an unrecognized intruder; therefore, the system is set to minimize the latter.

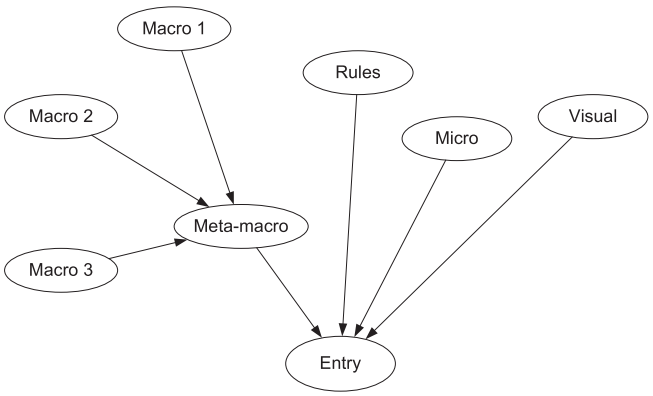

Fig. 8. Bayesian network used for the reasoning.

5. Experimental results

An experimental verification was performed in the prototype environment as described in Section 3.4. It consisted of a learning and an evaluation phase. In this paper we report on one learning and three evaluation experiments.

5.1. Learning phase

In the learning phase, four people were recorded accessing the system. Each individual completed 40 regular entries that were used as positive learning examples. The negative learning examples for one individual were the entries of the other three people. We built decision trees for the macro modules, constructed learning sets for the LOF algorithm in the micro and macro module and a comparison set for the visual learning module, and adjusted the system parameters. After the learning was completed, the system was ready to operate.

5.2. Evaluation phase

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 a *fake-identity* scenario, where an adversary has stolen an employee's identity. We recorded the 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). Each user made 31 regular entries, serving as the testing 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 the integration was slightly changed, omitting the visual-learning module. The experiments were run on already-learned and tuned modules from the first phase, while the probabilities in the Bayesian network were obtained with a 10-fold-cross validation.

In the first experiment the identities of the employees were swapped. We took four employees that were known to the system and shuffled their identities in order to simulate a scenario where an employee hands over his/her identity. The dataset contained 496 examples with a distribution of 75.00% negative examples (fake identity).

The performance of the system and the modules in the first experiment 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 an accuracy, e.g., the left-most number represents the percentage of irregular entries that were predicted as regular by the expert rules. The last column presents the overall accuracy of a module. The system achieved an overall accuracy of 95.77%. The expert rules always predicted *OK*, because all the entries were formally regular according to the entry procedure. The micro learning performed well in detecting both irregular and regular entries, while the macro learning had 10.08% more mistakes. The high accuracy of the micro module was expected because it is relatively easy to distinguish the movement of a couple of people given sufficient learning examples.

In the second experiment we used the entries of the intruders, which were unknown to the system, and assigned them the identities of the employees. In this way we simulated a stolen-identity scenario. The dataset had 496 examples with a distribution of 75.00% negative examples.

The measurements on the second dataset are shown in Table 2. The system achieved an overall accuracy of 96.57%. In contrast to the results in Table 1, where macro learning classified 16.13% false positives, the number of false positives in Table 2 is only 1.88%. However, the trend in the micro learning is just the opposite; the overall accuracy is comparable in both datasets. The decline in the micro-learning performance was to be expected, since it is more difficult to classify new, unseen behavior than to distinguish between the known cases.

In the third, most relevant, experiment, we invited security experts from the Slovenian Ministry of Defense to test the system with an on-line simulation of various security attacks. For the purpose of scientific experimentation, the following eight scenarios were proposed, 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 a 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 a secure room using a single identity;
6. burglary: an attacker disables the hardware protection by force;
7. fake identity: an attacker accesses a secure room with a stolen identity card and a forged fingerprint;
8. kidnapping: an attacker forces an employee to enable access to a secure room.

Each scenario was imitated several times by different users and in a different order, as requested by the security experts. In total, 45 irregular entries and 15 regular entries were performed. The video learning module was active.

The results described in Table 3 are separated into 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 the disabling of sensors (e.g., the burglary scenario).

Table 1
System and module performance in the off-line *swapped-identity* experiment with four employees only.

Modules	Scenarios %				Overall accuracy %
	Irregular entries		Regular entries		
	OK	*alarm*	*OK*	*alarm*	
Expert rules	100.00	0.00	100.00	0.00	25.00
Micro learning	5.91	94.09	92.74	7.26	93.75
Macro learning	16.13	83.87	83.06	16.94	83.67
Integration	1.08	98.92	86.29	13.71	95.77

Table 2
System and module performance in the off-line *stolen-identity* experiment with four employees and three intruders.

Modules	Scenarios %				Overall accuracy %
	Irregular entries		Regular entries		
	OK	*alarm*	*OK*	*alarm*	
Expert rules	100.00	0.00	100.00	0.00	25.00
Micro learning	22.04	77.96	92.74	7.26	81.65
Macro learning	1.88	98.12	82.26	17.74	94.15
Integration	0.00	100.00	86.29	13.71	96.57

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

Modules	Scenarios %					Overall accuracy %
	Irregular entries			Regular entries		
	OK	*alarm*	*failed*	*OK*	*alarm*	
Expert rules	84.44	15.56	0.00	100.00	0.00	36.67
Micro learning	0.00	88.89	11.11	93.33	6.67	90.00
Macro learning	0.00	88.89	11.11	86.67	13.33	88.33
Visual learning	8.89	88.89	4.44	73.33	26.67	85.00
Integration	0.00	100.00	0.00	86.67	13.33	96.67

The system achieved an overall accuracy of 96.67%, identifying all the irregular entries and being too suspicious of two regular entries. Once again, the expert rules classified with a low accuracy (36.67%), but when an entry was classified as an alarm, it was indeed so. The rules were also more robust compared to the other modules, which, for example, failed to recognize the burglary scenario. The micro- and macro-learning modules recognized the irregular entries with the same accuracy, but macro learning made more mistakes when classifying the regular entries. It should be noted that all the tests were performed within two hours, which is not well suited to macro learning. The visual learning was slightly more robust than the learning modules, but achieved a lower accuracy.

6. Discussion and conclusion

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

In a practical evaluation[1] we presented three experiments, which demonstrated encouraging results. It was clear that each module has its own strong and weak points. However, an advanced combination and integration overcomes the individual 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 90.00%, while the default accuracy (which is rather meaningless) was 75.00%. The accuracy of the integrated system was 96.67%.

In each system there is a fine line between being too sensitive and not being sensitive enough to small changes in behavior. Although some of the methods, e.g., the Bayesian network, are quite robust, any practical application needs some fine tuning of the system parameters. One of the first major benchmarks painfully reminded us of the difference between a laboratory test and a field test, i.e., one of the 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 must include a mechanism for continuous learning and adaptation to the user over time.

The complex methods implemented seem to be excessive for a simple commercial application. In its current form the system is more appropriate for high-security areas. Namely, the joint-verification methods turned out to be very hard to bypass. A single method can be fooled relatively easily, but deceiving different methods within a normal time interval is a much harder task.

In summary, intelligent risk analysis at an access point represents an improvement in terms of risk analysis and has the potential to demonstrate this in real-time applications.

[1] A short video of the third experiment is available online: http://www.youtube.com/watch?v=BNDgfFRQkU4.

Acknowledgment

This study has received funding, partly from the Slovenian Ministry of Defense (MORS) and partly from the Slovenian Research Agency (ARRS). The authors also thank the 20 members of the project team, in particular Jana Krivec, Robert Blatnik and Aleš Tavčar, as well as the security experts and supervisors.

References

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

Bontempi, G., & Borgne, L. (2005). An adaptive modular approach to the mining of sensor network data. In *Proceedings of the workshop on data mining in sensor networks*. Newport Beach, CA: SIAM SDM.

Brazdil, P., Giraud-Carrier, C., Soares, C., & Vilalta, R. (2009). *Metalearning: Applications to data mining*. Berlin Heidelberg: Springer.

Breunig, M. (2001). *Quality driven database mining*. Ph.D. thesis, University of Munich.

Depren, O., Topallar, M., Anarim, E., & Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications, 29*(4), 713–722.

Fierrez-Aguilar, J., Garcia-Romero, D., Ortega-Garcia, J., & Gonzalez-Rodriguez, J. (2005). Adapted user-dependent multimodal biometric authentication exploiting general information. *Pattern Recognition Letters, 26*(16), 2628–2639.

Friedman-Hill, E. (2009). Jess, the rule engine for the java platform. <http://www.jessrules.com>.

Horrocks, I., Patel-schneider, P. F., & Harmelen, F. V. (2003). From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics, 1*, 2003.

Lamborn, P., Williams, P. J. (2006). Data fusion on a distributed heterogeneous sensor network. In *Proceedings of SPIE* (Vol. 6242, pp. 1–8). Orlando, FL, USA.

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

Perš, J., Kristan, M., Perše, M., Kovačič, S. (2007). Motion based human identification using histograms of optical flow. In *CVWW 2007: proceedings of the 12th computer vision winter workshop*. Graz, Austria (pp. 19–26).

Protégé, 2009. Open source ontology editor and knowledge-base framework. <http://protege.stanford.edu>.

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

Stephen, B., & Petropoulakis, L. (2005). An ambient software monitoring system for unsupervised user modelling. *Expert Systems with Applications, 28*(3), 557–567.

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

Tušar, T., & Gams, M. (2006). Outlier detection in an access control system (in Slovene). In *Proceedings of the 9th international multiconference information society – IS 2006* (pp. 136–139). Ljubljana, Slovenia: Jozef Stefan Institute.

Vilalta, R., & Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial Intelligence Review, 18*, 77–95.

W3C (2004). SWRL: A Semantic web rule language combining OWL and RuleML. <http://www.w3.org/Submission/SWRL>.

Wahyudi, W. A., & Syazilawati, M. (2007). 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.

Wang, H. (1997). Intelligent agent-assisted decision support systems: Integration of knowledge discovery, knowledge analysis, and group decision support. *Expert Systems with Applications, 12*(13), 323–335.

Wilson, D. L. (2006). Intelligent video systems for perimeter and secured entry access control. In *Proceedings of the 39th annual IEEE international carnahan conference on security technology ICCST* (pp. 260–262).

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). Morgan Kaufmann.

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

Zhang, Y., Zhang, X. J., Liu, Z. J. (2007). Irregular behavior recognition based on two types of treading tracks under particular scenes. In *Proceedings of the second international conference KSEM 2007*. Melbourne, Australia (pp. 508–513).