Intelligent High-Security Access Control

Matjaž Gams and Tea Tušar
Department of Intelligent Systems
Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
E-mail: matjaz.gams@ijs.si, tea.tusar@ijs.si

Keywords: access control, machine learning, intelligent data analysis, outlier detection

Received: July 12, 2007

Access control is an important security issue in particular because of terrorist threats. Access points are increasingly becoming equipped with advanced input sensors often based on biometrics, and with advanced intelligent methods that learn from experience. We have designed a flexible modular system based on integration of arbitrary access sensors and an arbitrary number of stand-alone modules. The system was tested with four sensors (a door sensor, an identity card reader, a fingerprint reader and a camera) and four independent modules (expert-defined rules, micro learning, macro learning and visual learning). Preliminary tests of the designed prototype are encouraging.

Povzetek: Članek opisuje vgradnjo inteligentnih metod v sistem za nadzor vstopa.

1 Introduction

Attacks on civil and institutional objects are becoming a potential threat in several parts of the world. The target might be a bank or a company, and the attack might be motivated by money or ideological reasons, but the essential pattern is the same. Due to the increase of these types of attacks it is important that advanced scientific and technological solutions are applied to real-life applications.

One of the important security tasks is to assure efficient access control, e.g. to differentiate between "proper" access of "fit" employees and all other attempts of access. An example of such an access point is presented in Figure 1.

Figure 1. A case access point: the door opens when two sensors on the right side confirm the identity of the acceding person. The upper sensor is an identity card reader and the lower a fingerprint reader.

These days, protective security uses a multi-layered approach, known as *defence in depth*. Defence in depth means combining several measures to make unauthorized access difficult for an external intruder or an employee, which is either without the needed permissions or not in "the right state of mind". In the defence in depth concept, various security measures complement and support one another, including physical space, security procedures, personnel and technology. Sub-concepts include also security awareness based on the co-operation of staff that fully know their responsibilities, and include human guards and security teams at various levels.

Quite often, the security system already includes one or more biometric sensors (Kolbe and Gams 2006; Ashbourn 2003; Jain et al. 1999). According to M. Kirkpatrick, assistant director of the FBI's criminal justice services division, "the only way to trace a terrorist is through biometrics". Indeed, biometrics is hard to overcome by itself (Wayman 2004; Toledano et al. 2006; Lumini and Nanni 2006). But as can be found on the Internet by simply browsing YouTube (for example search: "MythBusters beat fingerprint security system"), each method can be fooled quite easily once the security mechanism is figured out. Therefore, in this paper we are interested in the introduction of intelligent methods into an existing security access system, thus adding another layer of security.

Intelligence is generally accepted as one of the most important factors when fighting crime and terrorism, but the term is usually limited only to human intelligence. In this paper, we are concerned with intelligent computer methods (Hopgood 2000; Albus and Meystel 2001; Turban et al. 2004) that learn from past experience and react on the basis of the so obtained knowledge (Mitchell 1997; Witten and Frank 2005).

Intelligent methods can be used to monitor the behaviour of people at an access point on two levels. On

micro level, we take advantage of the fact that at an access point, a person always acts in a similar manner that rarely changes over time and is usually different for different people. The way a person accesses the identification and verification sensors depends on his/hers habits and motoric abilities. For example, person #1 always carries his identity card in his wallet and puts the whole wallet near the identity card reader, while person #2 keeps her identity card in the purse and always needs some time before pulling out her identity card and putting it back into place after identification. Similarly, some people throw open the door, while others open the door just enough to slip through. In short, because of people's usual behaviour on the micro level we can learn their pattern and use this information to further verify the accessing person.

Behaviour on the *macro level* refers to the daily routine of the users of the access control system. While a person can have the required permissions to all access points in the system, it usually accesses only some points or the access depends on the current week-day or hour. Similarly, smokers usually exit the building more often than non-smokers etc. The system can also detect dependencies between users, such as person #1 and person #2 always enter or exit at the same access point in a short period of time.

In this way the intelligent module acts as a security guard that is familiar with each employee and notices not only clear violations of attempted access by unauthorised people, but also abnormal states of the employees, e.g. in cases of drug abuse or mental imbalance. Additionally, the learnt patterns can be later analyzed by the security personnel.

Machine learning methods are among the most popular artificial intelligence techniques and have been used in various applications. Several open-source machine learning toolkits are available on the internet, in particular Orange (Demšar et al. 2004) and Weka (Witten and Frank 2005). It is no surprise that machine learning methods are also used for security tasks.

The rest of the paper is structured as follows: Section 2 reviews the related work. The structure of the proposed system is presented in Section 3, while Section 4 is dedicated to the sensor level of the system. In Section 5 we show how machine learning methods can be used for solving this problem. Details on the used modules are given in Section 6. Finally, Section 7 is dedicated to the experimental evaluation of the presented system, while Section 8 concludes the paper by summarizing the work done.

2 Related work

Existing access control systems usually consist of a combination of identification and verification modules that range from traditional knowledge-based (e.g. passwords or PIN codes) and token-based (e.g. ID cards and smart cards) security modules to more advanced sensors that are based on biometric information from handwriting, fingerprint, face, voice, retina, iris, hand geometry and even vein patterns. See (Kolbe and Gams 2006; Li et al. 2006) for a review of existing biometric sensors applied to access control. Additionally, the most advanced control systems rely on video cameras to monitor the behaviour of users. The so-called *intelligent video systems* (Wilson 2005) enable differentiation between normal and abnormal behaviour of users and are often capable of covering a wide perimeter.

However, the mentioned systems do not implement a higher level of intelligence that would serve as an additional security module. While in (Lamborn and Williams 2006) machine learning methods are used to combine the results of heterogeneous sensors into a global solution, to our best knowledge there is no working access control system that applies machine learning methods to detect abnormalities in user behaviour on micro and macro level. Instead, machine learning is often used on similar applications, such as keystroke dynamic user authentication (Revett et al. 2007), computer network intrusion (Eskin et al. 2002; Lane and Brodley 1999) or intrusion detection in web applications (García Adeva and Pikatza Atxa 2007). The applied algorithms range from probabilistic neural networks and support vector machines to *k*-nearest neighbour and cluster-based algorithms.

3 Structure of the system

To design and test our intelligent methods for access control, we have set up an experimental environment, as presented in Figure 2. It consists of a single access point with a door equipped with an open/close sensor, an identity card reader, a fingerprint reader and a camera. After the two readers have successfully identified and verified the user, the door unlocks. The user opens the door, passes through the door and the door automatically closes. The whole entry process is recorded by the camera, while the input signals of the other three sensors are connected using a controller and inputted into a database. For each successful access, the following four times are registered:

- time of acceptance of the identity card,
- time of acceptance of the fingerprint,
- time of door opening,
- time of door closing.

For the proposed system, the number of registered times is adaptable to a particular configuration, but should not be too small, e.g. less than three, or too big, e.g. hundreds.

After the user passes the access point, the data collected during the access is additionally processed by the four modules of our intelligent system:

- expert-defined rules (cover the unwanted behaviour on the macro level),
- micro learning (learns the patterns of access on the micro level),
- macro learning (learns the patterns of access on both the macro and micro level – it consists of three sub-modules that are presented in more detail in Section 6),
- visual learning.

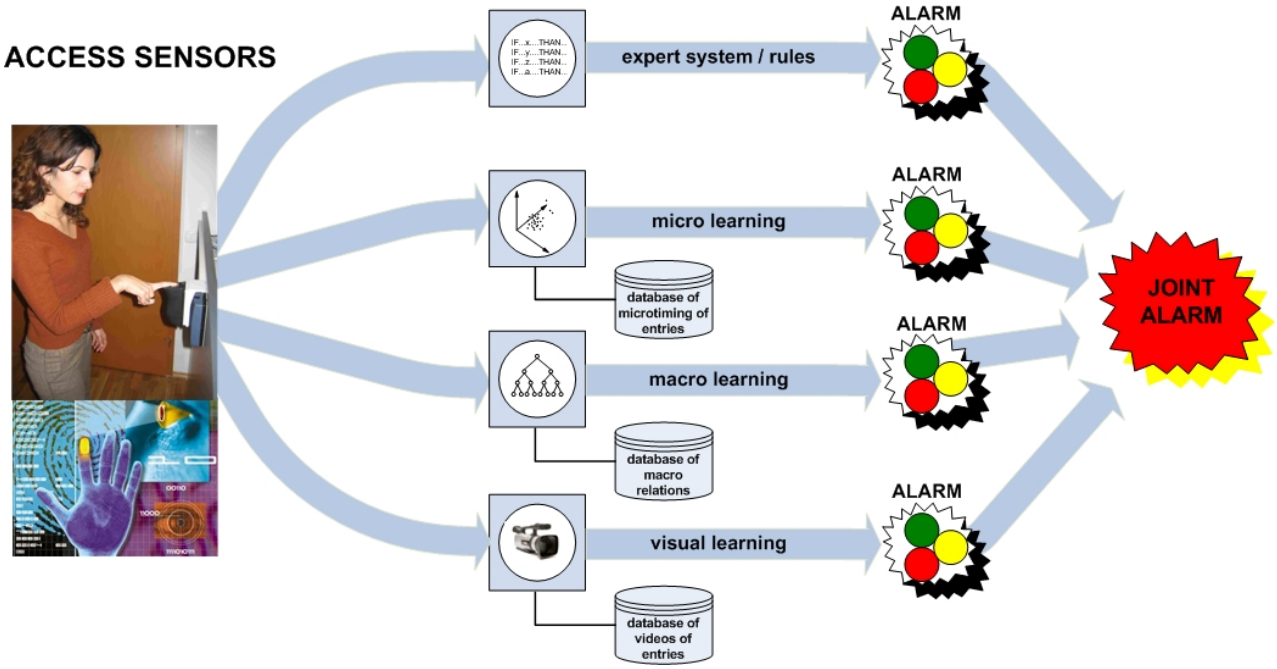

Figure 2. The structure of the intelligent access control system prototype.

Each module performs classification on its own and the final classification is calculated from the basic modules. Each module can classify an access into one of the following three categories:

- OK (according to the module, the access is regular),
- warning (it is unclear whether the access is regular or not),
- alarm (according to the module, the access is irregular).

An example of the system's output is presented in Figure 3. All the texts are in Slovene. Red color represents alarm. The macro module classification consists of the classifications of its three sub-modules.

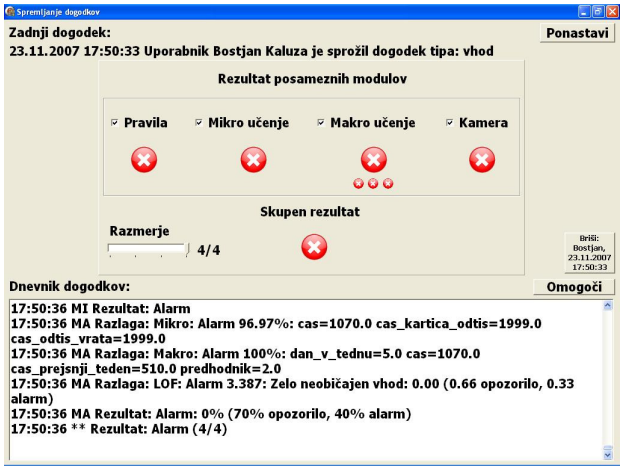

Figure 3. The graphical user interface of our intelligent access control system for an irregular access. The four big circles represent classifications of the four stand-alone modules. The single circle bellow represents the joint and final classification, while the text at the bottom shows the explanations for each module.

All modules perform in parallel and each transmits its decision and explanation when it finishes. Therefore, the classifications in the graphical user interface appear asynchronously. The log of module events consists of the exact time of module decision, followed by its classification, certainty factor and additional explanation of the decision (where possible).

4 Access sensors

The four sensors used in our experiments (door sensor, identity card reader, fingerprint reader and camera) are standard commercially available sensors that do not need additional explanation. While the two used readers allowed only the time of acceptance to be recorded, different identification/verification devices allow the recording also of other features, e.g. time of start of the identification/verification process, confidence rate, time between pressing buttons of a console etc. Our approach is in general independent on the number of such features, but all the experiments were performed using the mentioned four sensors and four recorded times.

4.1 The DOX controller

The input signals from the door sensor, identity card reader and fingerprint reader were collected by DOX – a multi-channel access controller, developed by the company Špica International (see Figure 4). DOX can be connected to various peripheral devices, such as card readers, touch screen consoles, door locks, biometric readers and other activators and sensors.

A single DOX controller is bound to one access point, however, multiple DOX controllers can be dynamically combined thus enabling the use of up to hundreds of input sensors on multiple locations in a

single access control network. For our purposes one DOX controller sufficed.

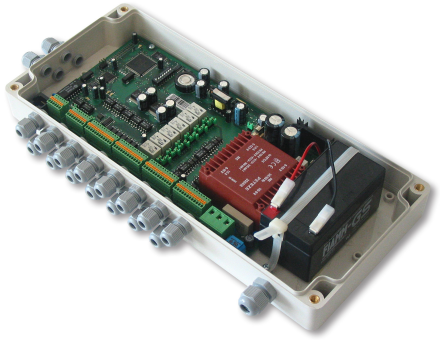

Figure 4. The DOX access controller.

4.2 Time&Space database

The Time&Space software is a commercial product by Špica International for access control and timekeeping. For the purposes of our research, the database of Time&Space needed minor modifications. In particular, a more detailed record of times from the input sensors was implemented. Essentially, all the other features, like scalability, modularity and security, remained the same thus enabling a reliable and flexible database.

5 Machine learning methods

Among the four independent modules of the intelligent system, only one (the expert-defined rules) does not use learning. The remaining three modules basically apply the general schema of machine learning, presented in Figure 5 (Kotsiantis 2007).

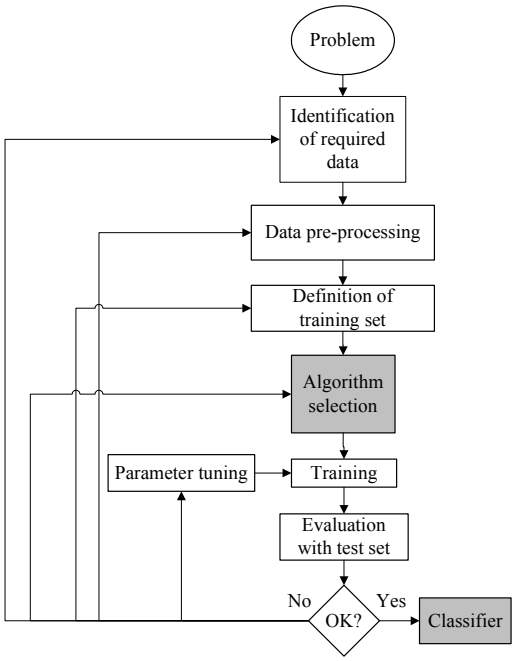

Figure 5. The process of applying supervised machine learning to a real-world problem (Kotsiantis 2007).

Inductive machine learning is the process of learning from instances (examples in a training set). As a result, a classifier is created that can be used to classify new instances. During the process, several steps are needed to achieve the desired functionality. For the security access control classification, the following properties of the classifier are the most important:

- the classifier should be as accurate as possible,
- the structure of the constructed knowledge should be comprehensible to the designers and system engineers as well as maintenance personnel,
- the classification result should be accompanied by an explanation that is comprehensible to all security personnel.

The first step in applying machine learning methods to a real-world problem is collecting the input data and features (Zhang et al. 2003). In our access control system, we had two major sources of input data and features – on the micro and macro level.

On the micro level, we record the video of the access together with times of identification, verification and door opening and closing. On the macro level, we note different macro features, such as time in the day, day of the week, persons that preceded the current person etc. The applied method of gathering features is a "brute-force" method, meaning that all the available features were gathered in the first stage.

The next step was gathering of data. In our experimental setting, only positive examples (regular accesses) were given as input data. Each access was recorded and stored into the database. It was then up to the individual modules to pre-processes the data for their specific needs.

Hodge and Austin (2004) have introduced a survey of contemporary techniques for outlier detection in the context of eliminating "bad" instances. However, in our case it was easy to visually eliminate the bad entries due to unexpected human errors during access. Namely, for high-security access, each person has to enter in a typical manner with as little variation as possible. If any unusual variation appears, the system is supposed to notice the deviation. A reasonable amount of robustness is necessary, but nothing more. In summary, it was not difficult to eliminate all but normal entries of all the tested persons. In a realistic application, it should not be difficult to eliminate those entries that stand out as outliers within the group of entries of one person.

The next step is feature selection – the process of identifying and removing as many irrelevant and redundant features as possible (Yu and Liu 2004). This reduces the dimensionality of the data and for large datasets enables machine learning algorithms to operate faster and more effectively. In our case there was no problem with too many features, but it is well known that typically some features depend on one another thus often unduly influencing the accuracy of supervised machine learning classification models. This problem is sometimes addressed by constructing new features from the basic feature set (Markovitch and Rosenstein 2002), which was also one of our intentions. However, the

relatively small changes we introduced were basically all hand-crafted.

On the basis of the created datasets, the choice of specific learning algorithms was performed. The algorithms were chosen from the machine learning toolkits Weka and Orange, both freely available from the internet. In addition, we have designed algorithms on our own. The first choice of algorithms was based on the task itself since it is well known that specific algorithms are suitable for specific types of problems. The second choice was based on algorithms' explanation capabilities. Two types of algorithms were finally chosen, one for constructing decision trees and one for outlier detection, as described in the next section.

6 Stand-alone modules

Our intelligent access control system consists of four stand-alone modules: expert-defined rules, micro learning, macro learning (combining the three sub-modules) and visual learning. They are presented in more detail in this section.

6.1 Expert-defined rules

Even existing commercial access systems use several rules to control basic behaviour on the macro level. Such rules describe, for example, the maximal time a door can be open or if a person can enter a building on a Saturday. In addition to taking into account these "common" rules we gathered new rules by consulting a security expert. Through knowledge acquisition we have obtained sufficient information to design altogether nine rule templates. Each rule template must be filled with exact data using a rule editor. As a result, several or several tens of concrete rules can be created. Some rules are quite simple while others demand specialized variables and routines to be performed properly.

An example of a rule template is: *The module triggers MESSAGE if user(s) SET_USERS do not exit the building in time TIME_LIMIT. MESSAGE* can be either a warning or an alarm, while *TIME_LIMIT* is an integer value measured in seconds. *SET_USERS* is an example of a special routine, which enables to choose a particular user, a particular group of users or all users of the system. This rule obviously demands a special hand-coded routine that checks all the persons that entered the buildings and did not exit and compares the time spent in the building with *TIME_LIMIT*.

The rules in the current implementation are not being chained as in a typical expert or a rule-based system. Rather, the inference engine checks all the rules on the list one by one and triggers messages in those rules that match the preconditions. Rules are transformed into SQL queries and are executed on the Time&Space database. If any rule triggers a warning (or an alarm), the classification of the whole module is warning (or alarm).

6.2 Micro learning

The micro learning module uses only the information gathered on the micro level. From the four input times recorded during each access (see Section 3), it calculates three time intervals, measured in microseconds:

- time between acceptance of the identity card and acceptance of the fingerprint,
- time between acceptance of the fingerprint and door opening,
- time between opening and closing of the door.

Each access can be thus represented as a point in a three-dimensional space defined by these three times (see Figure 6). All regular accesses of a person compose the training set for this person. When a new access by this person (or someone who impersonates this person) is made, the new three-dimensional point is compared to the existing "regular" points. If it does not match the regular accesses, the module classifies it as either a warning or an alarm (depending on how much it differs from the training set).

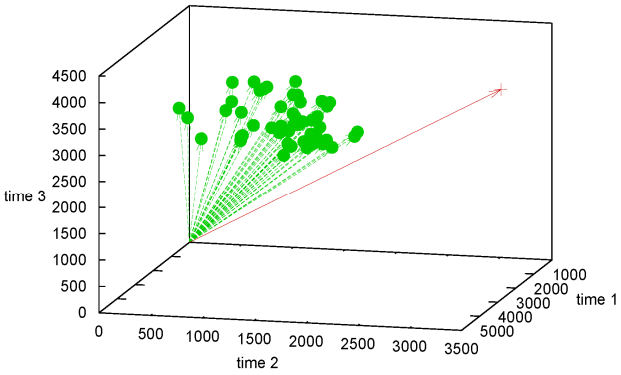

Figure 6: The three-dimensional representation of access times used in micro learning.

Outlier detection algorithms can be used for the purpose of micro learning. In (Tušar and Gams 2006) we reviewed existing outlier detection algorithms and chose the so-called Local Outlier Factor (LOF) for our task (Breunig et al. 2000; Breunig 2001). This algorithm was preferred to the others because it calculates a real number representing the "degree of outlierness" for every point (the greater the value of LOF, the more outlying the point). This gives us the possibility to define warnings and alarms by simply setting the bounds to the value returned by LOF. For example, taking the values 1.1 and 1.3 as bounds, all points that have the LOF value lower than 1.1 are classified as OK, the points that have the LOF value between 1.1 and 1.3 are classified as warning, while all the other points (having the LOF value greater than 1.3) are classified as alarm. While this gives us great flexibility, it also requires the proper setting of these bounds for the algorithm to work properly.

6.3 Macro learning

The macro learning module is composed by three sub-modules, where each captures a different aspect of learning the behaviour of users on the macro level. All sub-modules use the attributes gathered from the macro

level, while two sub-modules use also the attributes from the micro level.

The used macro attributes are, for example, the time in the day, the day of the week (e.g. Saturday), the specific date in relation to the month (e.g. each first Monday in a month) etc. Each date also relates to a specific user in a specific way – e.g. normal working days, vacations, reported sick leaves etc. The third relation deals with previous entries in a specific time period, e.g. the last hour or on each Monday. The data features include timing of entries of the person himself or in relation to any other access of any other person. For example, appropriate input data features and examples enable finding patterns such as person #1 and person #2 always enter at the same access point inside a one minute interval.

It is important to note that learning on the macro level could be much more powerful if we had more than one access point in the system. The combination of entries at different access points gives the macro module more information and therefore facilitates the verification of users.

6.3.1 Macro tree learning

This sub-module constructs decision trees by using only attributes on the macro level. All regular entries of a person were used as positive learning examples, while all other entries were used as negative learning examples. The sub-module can classify a new access either as OK or as an alarm.

Several algorithms from machine learning toolkits Weka and Orange were tested for this task and they performed similarly. We finally chose the J48 algorithm, which is the Java implementation of Quinlan's algorithm C4.5 for constructing decision trees (Quinlan 1993). A major addition to the plain classification using decision trees was the conception of the explanation. When a classification occurs, the user interface displays on the screen the tree used for this classification. The path leading from the root to the chosen leaf is then coloured according to the classification output – either green for an OK access or red for an alarm. This gives the security personnel a comprehensive explanation of the decision of the sub-module.

6.3.2 Macro micro tree learning

This sub-module works as the previous one, with the difference that here, also the attributes from the micro level are included. Typically, the trees constructed using macro micro tree learning include some macro and some micro attributes, often gathered separately in sub-trees.

6.3.3 Macro micro LOF learning

In this sub-module, the macro and micro attributes are used by the algorithm LOF (see Subsection 6.2). While the LOF algorithm in micro learning worked only on three attributes, meaning that its results could be easily visualized, the number of attributes here is higher and other methods (such as parallel coordinates) need to be used for explanation purposes.

Classifications of all the three sub-modules are combined together using voting to represent the joint macro module classification.

6.4 Video learning

This module is essentially different from all the other ones as it learns from the camera recordings by using histograms of optical flows. More on this method can be found in (Kolbe et al. 2005; Perš et al. 2007; Sidenbladh and Black 2003).

6.5 Joint classification

In the first version, the system tackles the decisions of the four modules as equally important parts. The results are first sorted according to their value – from OK, through warning to alarm. Then, using a threshold value that is set by the security personnel that handles the system, the results of the modules are combined into the joint result in the following way. If, for example, the threshold is set to 2/4 (two modules out of four), the joint classification is equal to that of the second of the four sorted results. The higher the threshold value the stricter the system. More advanced methods are in progress (Verlinde et al. 2000; Gams 2001).

7 Empirical verification

While our implementation includes four sensors (door sensor, identity card reader, fingerprint reader and camera) and four independent modules (expert-defined rules, micro learning, macro learning and visual learning), we left out the camera and visual learning module from the presented experiments as it was not available for the tests.

Five people accessed this system – each first completing around 40 regular entries that served as learning examples for the three modules. After that, each person made another ten regular entries for testing the system. For the purpose of scientific evaluation we have performed the "fake-identity" experiment, in which each person successfully cheats the identity card reader and the fingerprint reader so that it can impersonate any other person. In this way, the testing data of one person can be used also as testing data for the other four people.

The results of this experiment are presented in Table 1. The first row represents the correct entries of the right persons. The first column represents "false" identities under which the classification system "saw" the access. There is a number of attack scenarios in which the attacker bypasses the sensory system, e.g. by a stolen identity card and a fake fingerprint. Another case would be a break in the database corrupting static data and thus faking identity, but not being able to fake the classification system output which appears directly on a screen as a result of dynamic computing.

pretender	real	#1			#2			#3			#4			#5			all		
		A	W	OK	A	W	OK	A	W	OK	A	W	OK	A	W	OK	A	W	OK
#1	rules	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10	0	0	50
	micro	0	0	10	1	2	7	10	0	0	4	5	1	10	0	0	25	7	18
	macro	0	3	7	8	2	0	10	0	0	10	0	0	9	1	0	37	6	7
	together	**0**	**3**	**7**	**8**	**2**	**0**	**10**	**0**	**0**	**10**	**0**	**0**	**10**	**0**	**0**	**38**	**5**	**7**
#2	rules	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10	0	0	50
	micro	0	1	9	0	0	10	5	5	0	2	0	8	6	4	0	13	10	27
	macro	10	0	0	0	1	9	1	9	0	0	10	0	10	0	0	21	20	9
	together	**10**	**0**	**0**	**0**	**1**	**9**	**5**	**5**	**0**	**2**	**8**	**0**	**10**	**0**	**0**	**27**	**14**	**9**
#3	rules	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10	0	0	50
	micro	9	1	0	7	1	2	0	0	10	1	0	9	4	4	2	21	6	23
	macro	10	0	0	4	6	0	0	0	10	1	9	0	10	0	0	25	15	10
	together	**10**	**0**	**0**	**7**	**3**	**0**	**0**	**0**	**10**	**1**	**9**	**0**	**10**	**0**	**0**	**28**	**12**	**10**
#4	rules	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10	0	0	50
	micro	4	4	2	0	0	10	0	0	10	0	1	9	0	1	9	4	6	40
	macro	2	7	1	0	4	6	0	0	10	0	0	10	0	0	10	2	11	37
	together	**6**	**3**	**1**	**0**	**4**	**6**	**0**	**0**	**10**	**0**	**1**	**9**	**0**	**1**	**9**	**6**	**9**	**35**
#5	rules	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10	0	0	50
	micro	10	0	0	9	1	0	10	0	0	7	1	2	0	0	10	36	2	12
	macro	10	0	0	9	1	0	10	0	0	10	0	0	0	1	9	39	2	9
	together	**10**	**0**	**0**	**9**	**1**	**0**	**10**	**0**	**0**	**10**	**0**	**0**	**0**	**1**	**9**	**39**	**2**	**9**
all	rules	0	0	50	0	0	50	0	0	50	0	0	50	0	0	50	0	0	250
	micro	23	6	21	17	4	29	25	5	20	14	7	29	20	9	21	99	31	120
	macro	32	10	8	21	14	15	21	9	20	21	19	10	29	2	19	124	54	72
	together	**36**	**6**	**8**	**24**	**11**	**15**	**25**	**5**	**20**	**23**	**18**	**9**	**30**	**2**	**18**	**138**	**42**	**70**

Table 1. Evaluation of the expert-defined rules, micro learning and macro learning on 5 persons.

Whatever the case, the experiment enables to check the success rate of the stand-alone modules and the integrated system (the threshold was set to 3/3). For example if considering only micro learning, it seems obvious that some people enter in a different manner due to different physical properties, but some of them are physically and motorically similar and therefore it is possible to observe this phenomenon in Table 1. E.g. person #1 (a small woman) caused the alarm of the micro module in nine and ten out of ten cases when entering as person #3 or person #5 (both tall men). When entering as person #4 (a small man), the micro module triggered the alarm only four times, but classified the access as OK only two times. Surprisingly, person #2 (a strong middle sized man) typically entered in a similar way as person #1 (a small woman), so she was able to successfully mislead the micro module in nine out of ten cases.

As seen from Table 1, each module has its own strong and weak points. The integrated system without the visual learning classified as OK 88% of all the tested regular entries and as alarm 69% of all the irregular entries as presented in Table 2 and Table 3.

In practical experiments of several additional scenarios, the expert-defined rules and the video learning module proved quite successful on their own, and the overall performance improved as well. Due to various tests with different scenarios and under different circumstances, overall statistics of those tests are not presented through tables.

	A	W	OK
rules	0%	0%	100%
micro	0%	2%	98%
macro	0%	10%	90%
together	**0%**	**12%**	**88%**

Table 2. Statistics for regular accesses.

	A	W	OK
rules	0%	0%	100%
micro	50%	15%	36%
macro	62%	25%	14%
together	**69%**	**18%**	**13%**

Table 3. Statistics for irregular accesses (impersonation of users).

8 Conclusions

We have designed and tested an intelligent high-security access control system consisting of four sensors (door sensor, identity card reader, fingerprint reader and camera) and four independent modules (expert-defined rules, micro learning, macro learning and visual learning). The emphasis was on modifying and applying

intelligent machine learning methods to distinguish regular entries from faulty or fake ones.

The methods were tested in an experimental setting, where each of the five tested persons tried to impersonate the other four. Results from these tests are encouraging.

The experiment can also be seen as introducing intelligence into the environment. Indeed, the applied methods introduce intelligence on top of existing hardware and software solutions, improving their performance and making activities comprehensible to human users, while at the same time not burdening them. A small drawback in using this system is that it first needs to learn the regular behaviour of users, which means that it can be used only after an amount of accesses of a user have been made. Also, if a person acquires come kind of disability (for example, by braking an arm), the learning must start anew. Furthermore, several parameters need to be set in order for the system to function properly. While this can sometimes be difficult, it gives the system the necessary flexibility that enables its application for different necessities.

In summary, the machine intelligence security layer that learns from previous entries seems to be an important additional mechanism improving overall security and quality of life in modern times.

Acknowledgement

The overall project was funded partly by the Slovenian Ministry of Defence and partly by the Slovenian Research Agency.

The authors thank all the members of the research team, in particular Jana Krivec, Robert Blatnik, Erik Dovgan, Boštjan Kaluža, Aleš Tavčar.

References

[1] Albus, J. S., Meystel, A. M. (2001). *Intelligent Systems: Architecture, Design, Control*. Wiley-Interscience.

[2] Ashbourn, J. (2003). *Practical Biometrics: From Aspiration to Implementation*. Springer.

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

[4] Breunig, M. M., Kriegel, H.-P., Ng, R. T., Sander, J. (2000). LOF: Identifying density-based local outliers. In: *Proceedings of the International Conference on Management of Data SIGMOD'00*, pp. 93–104.

[5] Demšar, J., Zupan, B., Leban, G. (2004). *Orange: From experimental machine learning to interactive data mining*, White Paper, Faculty of Computer and Information Science, University of Ljubljana.

[6] Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *Data Mining for Security Applications*. Kluwer Academic Publishers.

[7] Gams, M. (2001). *Weak intelligence: Through the principle and paradox of multiple knowledge*. Advances in Computation: Theory and Practice, vol. 6. Nova Science Publishers.

[8] García Adeva, J. J., Pikatza Atxa, J. M. (2007). Intrusion detection in web applications using text mining. *Engineering Applications of Artificial Intelligence*, vol. 20, no. 4, pp. 555-566. Elsevier.

[9] Hodge, V. J., Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85-126. Springer.

[10] Hopgood, A. A. (2000). *Intelligent Systems for Engineers and Scientists*. CRC Press.

[11] Jain, L. C., Halici, U., Hazashi, I., Lee, S. B., Tsutsui, T. (1999). *Intelligent Biometric Techniques in Fingerprint and Face Recognition*. CRC Press.

[12] Kolbe, M., Gams, M. (2006) Towards an intelligent biometric system for access control. In: *Proceedings of the 9th International multiconference Information Society - IS 2006*, vol. A, pp. 118-122. Jožef Stefan Institute.

[13] Kolbe, M., Gams, M., Kovačič, S., Perš, J. (2005). A system for automatic behaviour recognition based on computer vision (in Slovene). In: *Proceedings of the 8th multiconference Information Society IS 2005*, pp. 357-361. Jožef Stefan Institute.

[14] Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica*, vol. 31, no. 3, pp. 249-268. Slovene Society Informatika.

[15] Lamborn, P., Williams, P. J. (2006). Data fusion on a distributed heterogeneous sensor network. In *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6242, pp. 62420R.1-8. SPIE, Bellingham, Washington.

[16] Lane, T., Brodley, C. E. (1999). Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information System Security*, vol. 2, no. 3, pp. 295-331. Association for Computing Machinery.

[17] Li, C., Yang, Y.-X., Niu, X.-X. (2006) Biometric-based personal identity-authentication system and security analysis, *Journal of China Universities of Posts and Telecommunications*, vol. 13, no. 4, pp. 43-47. Elsevier.

[18] Lumini, A., Nanni, L. (2006). An advanced multi-modal method for human authentication featuring biometrics data and tokenised random numbers. *Neurocomputing*, vol. 69, no. 13-15, pp. 1706-1710. Elsevier.

[19] Markovitch, S., Rosenstein, D. (2002). Feature generation using general constructor functions. *Machine Learning*, vol. 49, no. 1, pp. 59-98. Springer.

[20] Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill.

[21] Perš, J., Kristan, M., Perše, M., Kovačič, S. (2007). Motion based human identification using histograms of optical flow. In: *Proceedings of the 12th Computer Vision Winter Workshop CVWW 2007*, pp. 19-26. Graz University of Technology.

[22] Quinlan, J. R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

[23] Revett, K., Gorunescu, F., Gorunescu, M., Ene, M., Tenreiro de Magalhaes, S., Dinis Santos, H. M. (2007). A machine learning approach to keystroke dynamics based user authentication. *International Journal of Electronic Security and Digital Forensics*, vol. 1, no. 1, pp. 55-70. Inderscience Enterprises.

[24] Sidenbladh, H., Black, M. J. (2003). Learning the statistics of people in images and video. *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 181-207. Kluwer Academic Publishers.

[25] Toledano, D. T., Pozo, R. F., Trapote, A. H., Gómez, L. H. (2006). Usability evaluation of multi-modal biometric verification systems. *Interacting with Computers*, vol. 15, no. 5, pp. 1101-1122. Elsevier.

[26] Turban, E., Aronson, J. E., Liang T.-P. (2004). *Decision Support Systems and Intelligent Systems*. Prentice Hall.

[27] 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*, vol. A, pp. 136-139. Jožef Stefan Institute.

[28] Verlinde, P., Chollet, G., Acheroy, M. (2000). Multi-modal identity verification using expert fusion. *Information Fusion*, vol. 1, no. 1, pp. 17-33. Elsevier.

[29] Wayman, J., Jain, A., Maltoni, D. Maio, D. (eds.) (2004). *Biometric Systems: Technology, Design and Performance Evaluation*. Springer.

[30] Wilson, D. L. (2005). Intelligent video systems for perimeter and secured entry access control. In *Proceedings of the 39th Annual IEEE International Carnahan Conference on Security Technology ICCST 2005*, pp. 260-262. IEEE.

[31] Witten, I. H., Frank, E. (2005). *Data Mining – Practical Machine Learning Tools and Techniques*, Chapter 4: Algorithms: The basic methods, pp. 97-105. Elsevier.

[32] Yu, L., Liu, H. (2004). Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research*, vol. 5, pp. 1205-1224. MIT Press.

[33] Zhang, S., Zhang, C., Yang, Q. (2003). Data Preparation for Data Mining. *Applied Artificial Intelligence*, vol. 17, no. 5-6, pp. 375-381. Taylor & Francis.