Transactions on Machine Learning
and Data Mining
Vol. 4, No. 1 (2011) 30-52

ISSN: 1865-6781,
ISBN: 978-3-942952-04-0

ibai Publishing
www.ibai-publishing.org

Moving Targets in Computer Security and Multimedia Retrieval

Giorgio Giacinto

Dip. Ing. Elettrica ed Elettronica
Università di Cagliari, Italy
giacinto@diee.unica.it

Abstract. The Internet era is changing the way Pattern Recognition has been defined in the past years. New applications are emerging whose characteristics can be hardly matched against the typical problem setting. The typical formulation of a pattern recognition problem assumes that data can be subdivided into a number of classes on the basis of the values of a set of suitable features. Supervised techniques assume that data classes are given in advance, and the goal is to find the most suitable set of features and classification algorithm that allows the effective partition of data. On the other hand, unsupervised techniques allow discovering the "natural" data classes in which data can be partitioned, for a given set of features. These approaches are showing their limitations to handle the challenges issued by applications where the definition of data classes is not uniquely fixed. As a consequence, the tasks of feature definition, and classifier training should be adapted to this changing environment. Two applications from different domains share similar characteristics in this respect, namely, Intrusion Detection in computer systems and Multimedia Retrieval. In intrusion detection, the adversary can carefully craft attack patterns so that they are undetected by the employed detector. On the other hand, the retrieval of multimedia data by content is biased by the high subjectivity of the concept of similarity. In this paper, the issues of the two application scenarios will be discussed, and some effective solutions and future research directions will be outlined.

Keywords: Concept drift, computer security, multimedia retrieval, relevance feedback

1. Introduction

Pattern Recognition aims at designing machines that can perform recognition activities typical of human beings [1]. During the history of pattern recognition, a number of achievements have been attained, thanks both to algorithmic development, and to the progress in the computing equipment. The development of new sensors, the availability of computers with very large memory, and high computational speed, has clearly allowed the use of pattern recognition applications in everyday life [2].

The traditional applications of pattern recognition are typically related to problems whose definition is clearly pointed out. Patterns usually correspond to real objects such as persons, cars, etc., whose characteristics are captured by cameras and other sensing devices. Patterns are also defined in terms of signals captured in living creatures, or related to environmental conditions captured on the earth or the atmosphere. Finally, patterns are also artificially created by humans to ease the recognition of specific objects. For example, bar codes have been introduced to uniquely identify objects by a rapid scan of a laser beam, instead of using complex computer vision techniques that would produce high recognition error rates. In summary, the success of pattern recognition applications requires the detailed definition of the objects that have to be classified, and the detailed definition of the characteristics of the classes the objects have to be assigned to.

In order to perform classification, measurable features must be extracted so that the classification can be performed on the basis of the values of the features. Very often, the definition itself of the pattern recognition task suggests some of the features that can be effectively used to perform the recognition. Sometimes, the features are extracted by understanding which process is undertaken by the human mind to perform such a task. As this process is very complex, because we actually don't know exactly how the human mind works, features are also extracted by formulating the problem directly at the machine level.

Then, different pattern classification algorithms are tested in order to assess the best classification accuracy that can be achieved with the set of available features. A classification problem can be solved using different approaches, the feasibility of each approach depending on the ease to extract the related features, and the discriminative power of each representation. Pattern classifiers based on statistical, structural or syntactic techniques are used depending on the most suitable model of pattern representation for the task at hand. Sometimes, a combination of multiple techniques is needed to attain the desired performances.

If the attained accuracy is below the requirements for the application at hand, new features must be devised, or most complex classification mechanisms has to be employed. It turns out that the design of a pattern classifier for a specific application is an iterative process, provided that the definition of data classes is stable, i.e., it does not change with time.

Nowadays, new challenging problems are facing the pattern recognition community. These problems are generated mainly by two causes. The first cause is the widespread use of computers connected via the Internet for a wide variety of tasks such as personal communications, business, education, entertainment, etc. Vast part of our daily life relies on computers, and large volumes of information are often shared via social networks, blogs, websites, etc. The safety and security of data is threatened

in many ways by different subjects, which may misuse the communication content, or stole the credentials for authentication to get access to bank accounts, credit cards, etc. The second cause is the possibility for people to easily create, store, and share, vast amount of multimedia documents. Digital cameras allow capturing an unlimited number of photos and videos, thanks to the fact that they are also embedded in a number of portable devices. This vast amount of media archives needs to be organized in order to ease future search tasks. It is easy to see that it is often impractical to automatically label the content of each image or different portions of videos by recognizing the objects in the scene, as we should have templates for each object in different positions, lighting conditions, occlusions, etc. [3]. It is quite instructive to see the huge effort spent within the LabelMe[1] project, where web users are asked to upload, segment and label images in order to create a quite large and representative database of object templates. This effort can be reduced when the goal is to annotate a corpus of images, as the system asks the user to manually annotate just a few image regions, and then propagates the labels to all the images sharing similar regions [4]. It can be concluded that in those cases in which image classification is to be performed without any constraint on the semantic domain, human interaction is always necessary.

Summing up, the safety and security of Internet communication must be enforced by taking into account the *adversary* nature of malicious activities, designed to evade attack detection mechanisms, while effective techniques for indexing and retrieval of multimedia data should take into account the *variability* of users' interests and goals, that causes the same multimedia content to be used in different contexts with different semantic meanings. Indeed, the user of multimedia retrieval systems can be considered as an *adversary* from the point of view of the retrieval systems, as the search behaviors and goals can rapidly change.

In this paper, I will try to highlight the common challenges that these novel (and urgent) tasks pose to traditional pattern recognition theory, as well as to the broad area of "narrow" artificial intelligence, as the automatic solutions provided by artificial intelligence to some specific tasks are often referred to.

1.1. Challenges in computer security

The detection of computer attacks is actually one of the most challenging problems for three main reasons. One reason is related to the difficulty in predicting the behavior of software programs in response to every input data. Software developers typically define the behavior of the program for legitimate input data, and design the behavior of the program in the case the input data is not correct. However, in many cases it is a hard task to exactly define all possible incorrect cases. In addition, the complexity and the interoperability of different software programs make this task extremely difficult. It turns out that software products always exhibit weaknesses, a.k.a. vulnerabilities, which cause the software to behave in an unpredicted way in response to some particular input data. The impact of the exploitation of these vulnerabilities often involves a large number of computers in a very short time frame.

[1] http://labelme.csail.mit.edu

Thus, there is a huge effort in devising techniques able to detect never-seen-before attacks.

The main problem is the exact definition of the behavior that can be considered as being normal and which cannot. One of the reasons relies in the fact that the vast majority of computers are general-purpose. Thus, the user may run different kind of programs, at any time, and switch among them in any fashion. It turns out that the normal behavior of one user is typically different to that of other users. In addition, new programs and services are rapidly created, so that the behavior of the same user changes over time. Finally, as soon as a number of measurable features are selected to define the normal behavior, attackers are able to craft their attacks so that it fits the typical feature values of normal behavior.

The above discussion, clearly show that the target of attack detection tasks rapidly moves, as we have an attacker whose goal is to be undetected. As a consequence, each move made by the defender to secure the system can be made useless by a countermove made by the attacker [5]. It is also worth noting that the defender has a partial knowledge of the attack, as the attacker may be able to craft the attacks in a way that drives the defender to select *accidental* features of the attacks as the most discriminative features [6]. The rapid evolution of the computer scenarios makes the detection problem quite hard, as the speed of creation and diffusion of attacks increases with the computing power of today machines.

1.2. Challenges in content-based multimedia retrieval

While in the former case, the computers are the source and the target of attacks, in this case we have the human in the loop. Digital pictures and videos capture the rich environment we experience everyday. It is quite easy to see that each picture and video may contain a large number of concepts depending on the level of detail used to describe the scene, or the focus in the description. Moreover, different users may describe an image using different categories, and the same user may classify the same image in different ways depending on the context.

Sometimes, an image may contain one or more concepts that can be prevalent with respect to others, so that if a large number of people are asked to label the image, they may unanimously use the same label. Nevertheless, it is worth noting that a concept may be also decomposed in a number of “elementary” concepts. For example, an ad of a car can have additional concepts, like the color of the car, the presence of humans or objects, etc. Thus, for a given image or video-shot, the same user may focus on different aspects. Moreover, if a large number of potential users are taken into account, the variety of concepts an image can bear is quite large. Sometimes the differences among concepts are subtle, or they can be related to shades of meaning.

How the task of retrieving similar images or videos from an archive can be solved by automatic procedures? How can we design procedures that automatically tune the similarity measure to adapt to the visual concept the user is looking for? Once again, the target of the classification problem cannot be clearly defined beforehand, and the tasks of feature extraction, selection and combination must be designed to explicitly take into account user needs.

1.3. Summary

Table 1 shows a synopsis of the above discussion, where the three main characteristics that make these two problems look-like similar are highlighted, as well as their differences. Computer security is affected by the so-called adversarial environment, where an adversary can gain enough knowledge on the classification/detection system that is used either to mislead the training phase of the system, or to produce mimicry attacks [7-10]. Thus, in addition to the intrinsic difficulties of the problem that are related to the rapid evolution of design, type, and use of computer systems, a given attack may be performed in apparently different ways, as often the measures used for detection actually are not related to the most distinguishing features. On the other hand, the user of a Multimedia classification and retrieval system can be seen as an adversary, as soon as she continuously challenges the system with requests the system can hardly satisfy [11].

The solutions to the above problems are far from being defined. However, some preliminary guidelines and directions can be given. Section 2 provides a brief overview of related works. A proposal for the design of pattern recognition systems for Intrusion Detection and Multimedia Retrieval will be provided in Sections 3 and 4, respectively. Conclusions are presented in Section 5.

Table 1. Comparison between Intrusion Detection in Computer Systems and Content Based Multimedia Retrieval

	Intrusion Detection in Computer Systems	**Content Based Multimedia Retrieval**
Data Classes	The definition of the normal behavior depends on the Computer System at hand.	The definition of the conceptual data class(es) a given Multimedia object belongs to is highly subjective
Pattern	The definition of pattern is highly related to the attacks the computer system is subjected to	The definition of pattern is highly related to the concepts the user is focused to
Features	The measures used to characterize the patterns should be carefully chosen to avoid that attacks can be crafted to be a mimicry of normal behavior	The low-level measures used to characterize the patterns should be carefully chosen to suitably characterize all high-level concepts

2. Related works

In the field of computer security, very recently the concept of adversarial classification has been introduced [7-10]. The title of the paper by Barreno et al. [7] “Can Machine Learning Be Secure?”, clearly points out the weaknesses of machine learning techniques with respect to an adversary that aims at evading or misleading the detection system. These works propose some statistical models that take into account the cost of the activities an adversary must take in order to evade or mislead the system. The conclusions of these works state that a system is robust against adversary actions as soon as the cost paid by the adversary to evade the system is higher than the chances of getting an advantage.

On one hand, the defender has an imprecise knowledge of the very nature of an attack, as the basic knowledge is made up of the *consequences* of an attack. Then, when some input data related to the attack is available, it is still difficult to assess the essential features that best describe the attack. At the same time, these features should be discriminative enough to avoid the generation of large volumes of false alarms.

Among the proposed techniques that increase the costs of the actions of the adversary, the use of multiple features to represent the patterns, and the use of multiple learning algorithms, provide solutions that not only make the task of the adversary more difficult, but also may improve the detection abilities of the system in spite of the limited knowledge available [8]. Nonetheless, how to formulate the detection problem, extract suitable features, and select effective learning algorithms still remains a problem to be solved.

Very recently, some papers addressed the problem of "moving targets" in the computer security community ([5], [12], [13]). These papers address the problem of changes in the definition of normal behavior, as well as changes in the strategy adopted by the defender. The framework of the so-called concept drift can be used to address these changes ([14], [15]). Concept-drift takes into account that many real-world supervised classification tasks cannot be performed by assuming that each pattern has to be assigned to one of the data classes in an unambiguous way [16]. The drift may occur either for changes in the environment, or for changes in the definition of the data classes provided by the human expert. These changes are often difficult to capture in an explicit way by feature measurements, so the techniques proposed for handling concept drift usually refer to a hidden context, i.e., it is assumed that classification also depends on some variables we are not able to measure. Solutions to concept drift include techniques for instance selection and instance weighting, that allows implementing mechanisms for learning and forgetting, feature weighting, and ensemble methods.

In the field of content based multimedia retrieval, a number of review papers pointed out the difficulties in providing effective features and similarity measure that can cope with the broad domain of content of multimedia archives [17-19]. The shortcomings of current techniques developed for image and video has been clearly shown by Pavlidis [11]. While systems tailored to a particular image domain (e.g., medical images) can exhibit quite impressive performances, the use of these systems on unconstrained domains reveals their inability to adapt dynamically to new concepts [20]. One solution is to have the user manually label a small set of representative images (the so-called relevance feedback) that are used as training set for updating the similarity measure. However, how to implement relevance feedback to cope with multiple low-level representations of images, textual information, and additional information related to the images, is still an open problem ([21], [22]). In fact, while it is clear that the interpretation of an image made by humans takes into account multiple information contained in the image, as well as a number of concepts also related to cultural elements, the way all these elements can be represented and processed at the machine level has yet to be found. At present, this task is performed by semi-automatic procedures, based on machine learning tools and validation by human experts (see for example the "Image Swirl[2]" tool by Google labs, and the

[2] http://image-swirl.googlelabs.com

recent release of the content-based search option within the Google Images search engine).

We have already mentioned the theory of concept drift as a possible framework to cope with the two above problems ([14], [15]). The idea of concept drift arises in active learning, where, as soon as new samples are collected, there is some context that is changing, and changes are also observed in the characteristics of the patterns themselves. This kind of behavior can be seen also in computer systems, even if concept drift captures the phenomenon only partly ([12], [13]). On the other hand, in content-based multimedia retrieval, the problem can be hardly formulated in terms of concept drift, as each multimedia content may actually bear multiple concepts.

A different problem is the one of finding specific concepts in multimedia documents, such as persons, cars, etc. In these cases, the concept of the pattern that is looked for may be actually drifted with respect to the original definition, so that it requires being refined. This is a quite different problem from the one that is addressed here, i.e., the one of retrieving semantically similar multimedia documents.

Finally, ontologies have been introduced to describe hierarchies and interrelationships between concepts both in computer security and multimedia retrieval ([23], [24]). These approaches are suited to solve the problems of finding specific patterns, and provide complex reasoning mechanisms, while requiring the annotation of the objects.

3. Moving Targets in Computer Security

3.1. Intrusion Detection as a Pattern Recognition Task

The intrusion detection task is basically a pattern recognition task, where data must be assigned to one out of two classes: attack and legitimate activities. Classes can be further subdivided according to the IDS model employed. For the sake of the following discussion, we will refer to a two-class formulation, without losing generality. It is worth recalling that, from the point of view of the defender, only the class related to the *normal* behavior of the system is completely known, while a partial and imprecise knowledge is often available for the attack class. This fact heavily affects the different phases of the design of intrusion detection mechanisms. In the following, the phases of the design of an Intrusion Detection System (IDS) are reported, the challenges faced are discussed, and feasible solutions are proposed.

The IDS design can be subdivided into the following steps:

1. Data acquisition. This step involves the choice of the data sources, and should be designed so that the captured data allows distinguishing as much as possible between attacks and legitimate activities.
2. Data preprocessing. Acquired data is processed so that patterns that do not belong to any of the classes of interest are deleted (noise removal), and incomplete patterns are discarded (enhancement).
3. Feature selection. This step aims at representing patterns in a feature space where the highest discrimination between legitimate and attack patterns, is attained. A

feature represents a measurable characteristic of the computer system's events (e.g. number of unsuccessful logins).

4. Model selection. In this step, using a set of example patterns (training set), a model achieving the best discrimination between legitimate and attack patterns is selected.

5. Classification and result analysis. This step performs the intrusion detection task, matching each test pattern to one of the classes (i.e. attack or legitimate activity), according to the IDS model. Typically, in this step an alert is produced, either if the analyzed pattern matches the model of the attack class (misuse-based IDS), or if an analyzed pattern does not match the model of the legitimate activity class (anomaly-based IDS).

3.2. Intrusion Detection and Adversarial Environment: key points

The aim of a skilled adversary is to realize attacks without being detected by security administrators. This can be achieved by hiding the traces of attacks, thus allowing the attacker to work undisturbed, and by placing "access points" on violated computers for further stealthy criminal actions. In other terms, the IDS itself may be deliberately attacked by a skilled adversary. A rational attacker leverages on the weakest component of an IDS to compromise the reliability of the entire system, with minimum cost.

3.2.1. Data Acquisition

To perform intrusion detection, it is needed to acquire input data on events occurring on computer systems. In the data acquisition step these events are represented in a suitable way to be further analyzed. Some inaccuracy in the design of the representation of events will compromise the reliability of the results of further analysis, because an adversary can either exploit lacks of details in the representation of events, or induce a flawed event representation. Some inaccuracies may be addressed with an a posteriori analysis, that is, verifying what is actually occurring on monitored host(s) when an alert is generated.

3.2.2. Data pre-processing

This step is aimed at performing some kind of "noise removal" and "data enhancement" on data extracted in the data acquisition step, so that the resulting data exhibit a higher signal-to-noise ratio. In this context the noise can be defined as information that is not useful, or even counterproductive, when distinguishing between attacks and legitimate activities. On the other hand, enhancements typically take into account a priori information regarding the domain of the intrusion detection problem. As far as this stage is concerned, it is easy to see that critical information can be lost if we aim to remove all noisy patterns, or enhance all relevant events, as typically at this stage only a coarse analysis of low-level information can be performed. Thus, the goal of the data enhancement phase should be to remove those patterns that can be considered noisy with high confidence.

3.2.3. Feature extraction and selection

An adversary can heavily affect both the feature definition and the feature extraction tasks, as the defender often has a partial or incomplete knowledge of the attacks. With reference to the feature definition task, an adversary can interfere with the process if this task has been designed to automatically define features from input data. With reference to the feature extraction tasks, the extraction of correct feature values depends on the tool used to process the collected data. An adversary may also inject patterns that are not representative of legitimate activity, but not necessarily related to attacks. These patterns can be included in the legitimate traffic flow that is used to verify the quality of extracted features. Thus, if patterns similar to attacks are injected in the legitimate traffic pool, the system may be forced to choose low quality features when minimizing the false alarm rate [6].

The effectiveness of the attack depends on the knowledge of the attacker on the algorithm used to define the "optimal" set of features, the better the knowledge, the more effective the attack. As "security through obscurity" is counterproductive, a possible solution is the definition of a large number of redundant features. Then, random subsets of features could be used at different times, provided that a good discrimination between attacks and legitimate activities in the reduced feature space is attained [25]. In this way, an adversary is uncertain on the subset of features that is used in a certain time interval, and thus it can be more difficult to conceive effective malicious noise.

3.2.4. Model Selection

Different models can be selected to perform the same attack detection task, these models being either cooperative, or competitive. Again, the choice depends not only on the accuracy in attack detection, but also on the difficulty for an attacker to devise evasion techniques or alarm flooding attacks. As an example, two recent papers from the same authors have been published in two security conferences, where program behavior has been modeled either by a graph structure, or by a statistical process for malware detection ([26], [27]). The two approaches provide complementary solutions to similar problems, while leveraging on different features and different models.

No matter how the model has been selected, the adversary can use the knowledge on the selected model and on the training data to craft malicious patterns. However, this knowledge does not imply that the attacker is able to conceive effective malicious patterns. For example, a machine learning algorithm can be selected randomly from a predefined set so that the attacker is unaware of the algorithm that is running in a given time frame [7]. As the malicious noise has to be well-crafted for a specific machine learning algorithm, the adversary cannot be sure of the attack success. Finally, when an off-line algorithm is employed, it is possible to randomly select the training patterns: in such a way the adversary is never able to know exactly the composition of the training set [28].

3.2.5. Classification and result analysis

An attacker may cause the IDS to be ineffective by either being able to evade the detection mechanisms, or by forcing the IDS to produce high volumes of false alarms, i.e., by *overstimulating* the IDS. To overstimulate or evade an IDS, a perfect

knowledge of the features used by the IDS is necessary. Thus, if such knowledge cannot be easily acquired, the impact can be reduced. This result can be attained for those cases in which a high-dimensional and possibly redundant set of features can be devised. Handling high-dimensional feature spaces typically require a feature selection step aimed at retaining a smaller subset of high discriminative features.
In order to exploit all the available information carried out by a high-dimensional feature space, ensemble methods have been proposed, where a number of machine learning algorithms are trained on different feature subspaces, and their results are then combined. These techniques improve the overall performances, and harden the evasion task, as the function that is implemented after combination in more complex than that produced by an individual machine learning algorithm [29-31].

A technique that should be further investigated to provide for additional hardness of evasion, and resilience to false alarm injection is based on the use of randomness [25]. Thus, even if the attacker has a perfect knowledge of the features extracted from data, and the learning algorithm employed, then in each time instant he cannot predict which subset of features is used. This can be possible by learning an ensemble of different machine learning algorithms on randomly selected subspaces of the entire feature set. Then, these different models can be combined by choosing a subset of them at random during the operational phase.

3.3. HMM-Web and HMMPayl - Detection of attacks against web-applications

As an example of an Intrusion Detection solutions designed according to the above guidelines, we provide an overview of HMM-Web and HHMPayl, two IDS designed to protect web applications and developed in our lab. HMM.web is a host-based intrusion detection system capable to detect both simple and sophisticated input validation attacks against web applications [32]. HMMPayl is a network-based system that analyzes the payload of HTTP packets at the byte level [29]. Both systems employ ensembles of Hidden Markov Models (HMM) that are trained using samples of normal HTTP requests. Attacks are detected by looking for anomalous (not normal) web application queries.

HMM-Web is made up of a set of application-specific modules (Figure 1). Each module is made up of an ensemble of Hidden Markov Models, trained on a set of normal queries issued to a specific web application. During the detection phase, each web application query is analyzed by the corresponding module. A decision module classifies each analyzed query as suspicious or legitimate according to the output of HMM. A different threshold is set for each application-specific module based on the confidence on the legitimacy of the set of training queries. Each query is made up of pairs <attribute,value>. The sequences of attributes in each query is processed by a HMM ensemble, while each value is processed by a HMM tailored to the attribute it refers to. As the Figure shows, two symbols (’A’ and ’N’) are used to represent all alphabetical characters, and all numerical characters, respectively. All other characters are treated as different symbols. This encoding has been proven useful to enhance attack detection and increase the difficulty of evasion and overstimulation.

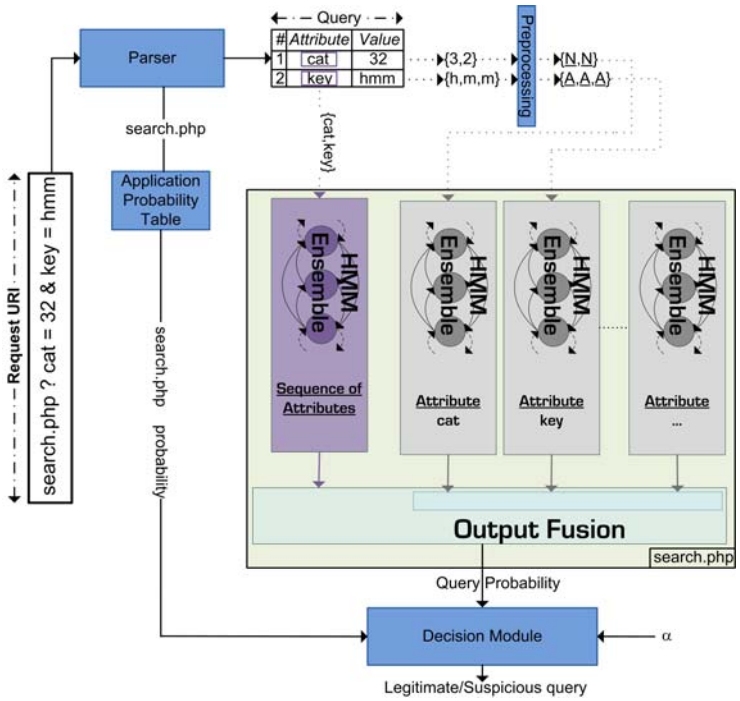

Fig. 1. Architecture of HMM-Web

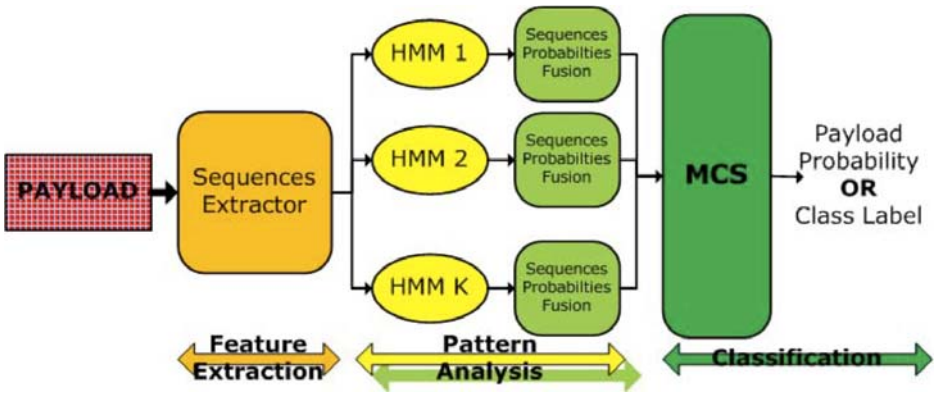

Fig. 2. Architecture of HMMPayl

HMMPayl performs payload processing in three steps as shown in Figure 2. The extraction of sequences of equal length, using a sliding window approach, allows the HMM to produce an effective statistical model which is sensitive to the " details" of the attacks (e.g., the bytes that have a particular value). Since HMM are particularly robust to noise, their use during the Pattern Analysis phase guarantees to have a system which is robust to the presence of attacks (i.e., noise) in the training set. In the Classification phase a Multiple Classifier System approach is adopted in order to

improve both the accuracy and the difficulty of evading the IDS. Besides, the MCS paradigm guarantees that the weaknesses of classifiers due to a suboptimal choice of initial parameters are mitigated.

Figure 3 shows the performance of HMM-Web on a dataset of real traffic collected at our institution. A set of attack has been crafted so that they are effective against the web server. It can be seen that the use of multiple models allows increasing the detection rate when small values of false alarms are allowed. It is worth noting that the detection rate takes into account the adversarial environment, as the attacks have been crafted so that they are similar to the normal traffic.

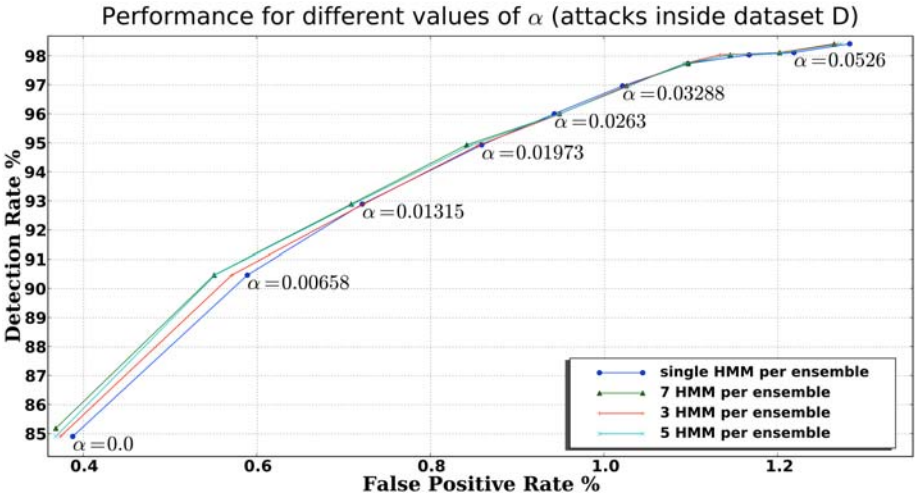

Fig. 3. ROC curve of HMM-Web. Comparison of HMM ensemble of different size

Figure 4 shows the performances of HMMPayl on three datasets of traffic, namely the GT dataset collected at the Georgia Institute of Technology, the DIEE dataset collected at our institution, and the DARPA dataset generated by the simulation of traffic in a computer network. The performances are compared to McPAD, an IDS that detects attacks against web applications by modeling the payload of normal traffic by a Multiple Classifier System (MCS) approach ([31]). In particular, each classifier in the MCS is trained on different feature representations of the payload.

A set of attacks, belonging to different categories, has been generated so that they represent the typical threats against web applications. A subset of attacks has been crafted so that they exhibit statistical characteristics similar to those of the normal traffic. In this way, we model an attacker that tries to evade the system by creating attack patterns that are similar to the normal traffic with respect to the statistical frequencies of bytes in the payload.

Performances are reported in terms of the Partial AUC, i.e., the Area Under the ROC Curve in the range [0, 0.1] of the false positive rate, so that false positive rate greater than 10% are not taken into account. Reported results show that both HMMPayl and McPAD provide good results in detecting attacks. In addition, HMMPayl can outperform McPAD by suitably choosing the length of the sequences.

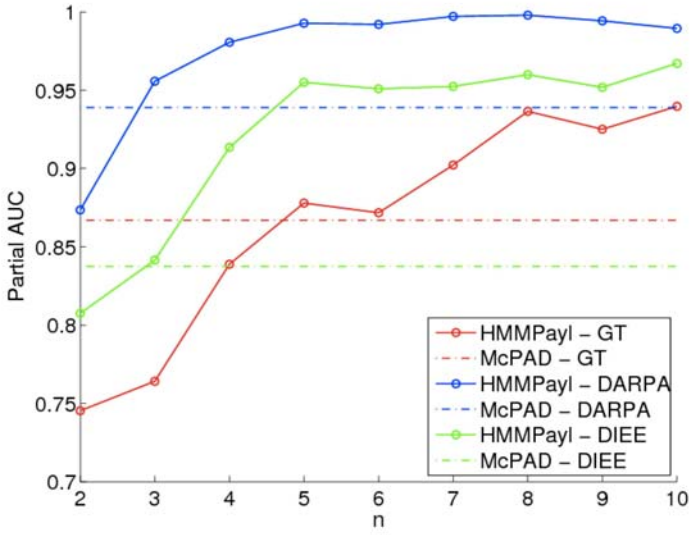

Fig. 4. Partial AUC values for the generic attacks dataset. n is the length of sequences extracted from the payload by HMMPayl

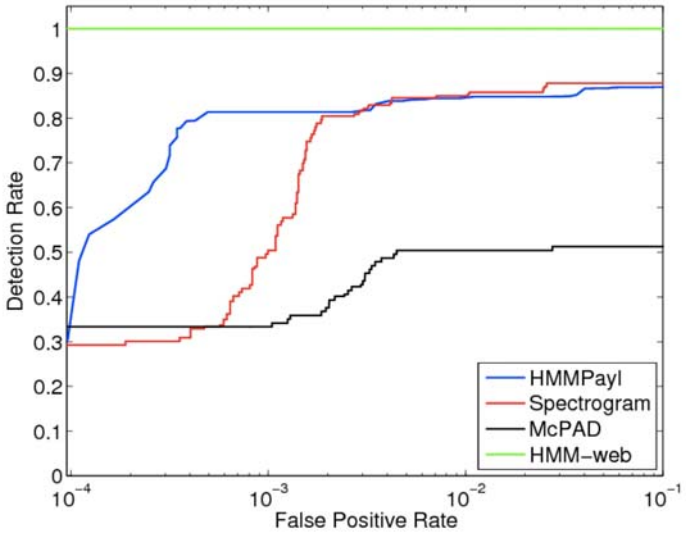

Fig. 5. ROC curves of HMMPayl, Spectrogram, McPAD, and HMM-Web on the DIEE dataset, XSS-SQL Injection attacks.

Figure 5 shows the ROC curves obtained by testing HMMPayl, McPAD, and HMM-Web on the DIEE dataset and the XSS-SQL Injection attacks. The performances of Spectrogram, an IDS based on Markov-Models to model the payload at the byte level, are also reported [33]. It can be seen that HMM-Web outperforms all other methods thanks to the fact that it does not work at the byte level, but at the application level, i.e., HMM-Web takes into account the semantic of web application requests. The main drawback of HMM-Web is the design phase, which requires the perfect knowledge of the web applications. As much as the complexity of web application increases, the design of HMM-Web may be too costly. On the other hand HMMPayl, Spectrogram and McPAD just require the availability of traces of network traffic that are analyzed at the byte level. HMMPayl provides good performances in all the considered range of false alarm rate, while Spectrogram is slightly better for high values of false alarm rate. On the other hand, McPAD provides lower performances, thus showing that Markov Models are well suited for modeling the normal behavior of web application, and detecting attacks even if they are crafted to mimicry normal web traffic.

4. Multimedia Retrieval

The design of a content-based multimedia retrieval system requires a clear planning of the goal of the system [17]. As much as the multimedia documents in the archive are of different types, are obtained by different acquisition techniques, and exhibit different content, the search for specific concepts is definitely a hard task. In these cases, a large number of different pattern recognition tasks should be defined by taking into account the multiple conceptual classes each single multimedia document may be assigned to. It is easy to see that as much as the scope of the system is limited, and the content to be searched is clearly defined, than the task can be more easily managed. On the other hand, the design of a *general purpose* multimedia retrieval engine is a challenging task, as the system must be capable to adapt to the changing behavior of the user. Let us recall that intrusion detection systems for computer network should be resilient to the changing behavior of attackers. To this end, we discussed the benefits of extracting large number of features, and combining multiple models. We shall see that similar solutions applies to multimedia retrieval tasks, as they allow to take into account a changing environment and act accordingly.

In the following, a short review of the challenges a designer should face is presented, and references to the most recent literature are given. In addition, some results related to a proof of concept research tool are presented.

4.1. Scope of the retrieval system

First of all, the scope of the system should be clearly defined. A number of content-based retrieval systems tailored for specific applications have been proposed to date. Some of them are related to sport events, as the playground is fixed, camera positions are known in advance, and the movements of the players and other objects (e.g., a ball) can be modeled [17]. Other applications are related to medical analysis, as the

type of images, and the objects to look for can be precisely defined. On the other hand, tools for organizing personal photos on the PC, or to perform a search on large image and video repositories are far from providing the expected performances.

In addition, the large use of content sharing sites such as Flickr, YouTube, Facebook, etc., is creating very large repositories where the tasks of organizing, searching, and controlling the use of the shared content, require the development of new techniques. While the answer to the question: *this archive contains documents with concept X?* may be fairly simple to be given, the answer to the question: *this document contains concept X?* is definitely harder. To answer the former question, a large number of false positives can be produced, and you will often find the target document mingled with a large set of non-relevant documents. On the other hand, the latter request requires a complex reasoning system that is far from the current state of the art.

4.2. Feature extraction

The description of the content of a specific multimedia document can be provided in multiple ways. First of all, a document can be described in term of its properties provided in textual form (e.g., creator, content type, keywords, etc.). This is the model used by Digital Libraries, where standard descriptors are defined, and guidelines for defining appropriate values are proposed. However, apart from descriptor such as the size of an image, the length of a video, etc., other keywords are typically provided by domain experts. In the case of very narrow-domain systems, it is possible to agree on an ontology that helps describing standard scenarios. On the other hand, when multimedia content is shared on the web, different users may assign the same keyword to different contents, as well as assign different keywords to the same content [34]. Thus, more complex ontologies, and reasoning systems are required to correctly assess the similarity among documents [35].

Multimedia content at the machine level is usually described by low-level and medium-level features ([11], [17]). These descriptions have been proposed by leveraging on the analogy that the human brain use these features to assess the similarity among visual contents. While at present this analogy is not deemed valid, these features may provide some additional hint about the concept represented by the pictorial content. Currently, very sophisticated low-level features are defined that take into account multiple image characteristics such as color, edge, texture, etc. [36]. Indeed, as soon as the domain of the archive is narrow, very specific features that are directly linked with the semantic content can be computed [20]. On the other hand, in a broad domain archive, these features may prove to be misleading, as the basic assumptions do not hold [11].

Finally, new features are emerging in the era of social networking. Additional information on multimedia content is currently extracted from the text in the web pages where the multimedia document is published, or in other web sites linked to the page of interest. Actually, the links between people sharing the images, and the comments that users post on each other multimedia documents, can provide a rich source of valuable information [37].

4.3. Similarity models

For each feature description, a similarity measure is associated. On the other hand, when new application scenarios require the development of new content descriptors, suitable similarity measures should be defined. This is the case of the exploitation of information from social networking sites: how this information can be suitably represented? Which is the most suitable measure to assess the influence of one user on other users? How we combine the information from social networks with other information on multimedia content? It is worth noting that the choice of the model used to weight different multimedia attributes and content descriptions heavily affect the final performance of the system. On the other hand, the use of multiple representations may allow for a rich representation of content that the user may control towards feedback techniques.

4.4. The human in the loop

While the above discussion provided some aspects that must be taken into account in order to capture the rich semantic content of multimedia, to properly react to the goal of each user involved in a retrieval task, human interaction must be included as parte of the process [21]. The involvement can be implemented in a number of ways. Users typically provide tags that describe the multimedia content. They can provide implicit or explicit feedback, either by visiting the page containing a specific multimedia document in response to a given query, or by explicitly reporting the relevance that the returned image exhibits with respect to the expected result. Finally, they can provide explicit judgment on some challenge proposed by the system that helps learning the concept the user is looking for [38]. Computers may ease the task for humans by providing a suitable visual organization of retrieval results, that allows for a more effective user interaction [39].

The retrieval system processes the information provided by the user in order to understand the goal of the search, and modify the search behavior accordingly. A number of approaches based on learning algorithms have been proposed. The main challenges are related to: i) the scarcity of training data, as user interaction usually produces a limited number of documents that the users marks as being relevant or not to the retrieval goal; ii) the imbalance between relevant and non-relevant documents, as in large databases it is easy that a large number of non-relevant documents exhibit similar content w.r.t. relevant documents; iii) the ambiguity of textual tags used to describe the image content.

4.5. ImageHunter: a prototype content-based retrieval system

A large number of prototype or demonstrative systems have been proposed to date by the academia, and by computer companies[3]. ImageHunter is a proof-of-concept system designed in our Lab[4] (Figures 6-8). This system performs visual query search

[3] An updated list can be found at http://savvash.blogspot.com/2009/10/image-retrieval-systems.html
[4] http://prag.diee.unica.it/amilab/WIH/

on a database of images from which a number of low-level visual features are extracted. The user can then provide the systems her feedback by marking the images that are relevant to the query (the other images are considered as being non relevant to the query). The system exploits this additional information to perform a new search, in order to retrieve more relevant images. In the following, some details on the proposed system are given.

4.5.1 Features

The system is partially based on the LIRE[5] library to perform feature extraction, and to Apache Lucene[6] to conveniently index the feature-based representation of images, and provide for a fast search engine. Actually the system extracts the following features:

- seven color based descriptors, namely, *Scalable Color, Color Layout, RGB Histogram, HSV Histogram, Fuzzy Color, Jpeg Histogram, and ABIF32*;
- three texture and shape features, namely. *Edge Histogram, Tamura, and Gabor Filters*;
- two descriptors that merge color and texture characteristics, namely, *CEDD (Color and Edge Directivity Descriptor)*, and *FCTH (Fuzzy Color and Texture Histogram)*.

4.5.2 Relevance Feedback

ImageHunter employs a nearest neighbor technique to compute a relevance score for each image in the archive, and to weight each feature space according to its effectiveness in representing relevant images as close points ([40], [41]). The use of the nearest-neighbor paradigm is motivated by its use in a number of different pattern recognition fields, where it is difficult to produce a high-level generalization of a class of objects, and just the similarities over a small set of prototypes can be computed. As soon as the user provide the feedback in terms of relevant and non-relevant images, the system compute a relevance score for the images in the archive as follows:

$$rel(I) = \left(\frac{1}{n/t+1}\right) \cdot rel_{NN}(I) + \left(\frac{n/t}{1+n/t}\right) \cdot rel_{BQS}(I) \qquad (\mathbf{1})$$

where n is the number of non-relevant images, and t is the total number of images retrieved after the latter iteration. The two terms $rel_{NN}(I)$ and $rel_{BQS}(I)$ are computed as follows:

$$rel_{NN}(I) = \frac{\left\| I - NN^{nr}(I) \right\|}{\left\| I - NN^{r}(I) \right\| + \left\| I - NN^{nr}(I) \right\|} \qquad (\mathbf{2})$$

where $NN^{r}(I)$ and $NN^{nr}(I)$ denote the relevant and the non relevant Nearest Neighbor of I, respectively, and $\left\| \cdot \right\|$ is the metric defined in the feature space at

[5] http://www.semanticmetadata.net/lire/
[6] http://lucene.apache.org/

hand. The use of the nearest neighbor paradigm allows to rapidly adapt the search according to the user needs, by exploring the feature space around each relevant image;

$$rel_{BQS}(I) = \frac{1 - e^{1 - d_{BQS}(I) / \max_i d_{BQS}(I_i)}}{1 - e} \quad (3)$$

where *i* is the index of all images in the database *and* d_{BQS} is the distance of image *I* from a reference vector computed according to the Bayes decision theory. The reader is referred to[40] for further details. This vector represents the point of the feature space that maximizes the likelihood of finding relevant images.

The search for relevant images is thus performed by combining an exploitation term (i.e., the $rel_{BQS}(I)$ term), and an exploration term (i.e., the $rel_{NN}(I)$ term). When few relevant images are available, then $rel_{BQS}(I)$ is called to bias the search towards the region of the feature space where it is likely to find relevant images according to the Bayes decision theory. On the other hand, when a large number of relevant images are available, the search is performed in multiple regions, according to the NN paradigm.

For each different feature space *f*, different scores $rel^f(I)$ are computed. Relevance feedback information can be further exploited to combine the relevance scores through a weighted sum

$$rel_{tot}(I) = \sum_{f=1}^{F} w_f \cdot rel^f(I) \quad (4)$$

where *F* is the number of feature spaces, and the weights w_f are computed as follows ([41]):

$$w_f = \frac{\sum_{I_i \in R} d^f_{\min}(I_i, N)}{\sum_{I_i \in R} d^f_{\min}(I_i, R) + \sum_{I_i \in R} d^f_{\min}(I_i, N)} \quad (5)$$

where *R* and *N* denote the sets of relevant and non relevant images, respectively, and $d_{\min}$ is defined as

$$d^f_{\min}(I_i, T) = \min_{I_k \in T} d^f(I_i, I_k) \quad (6)$$

Thus, a large weight is assigned to those feature spaces where the average distance between the closest pairs of relevant images is small compared to the average distance between the closest pairs made up of relevant and non-relevant images.

4.5.3 Performances of the systems

Figures 6 to 8 represent an example of the results attained by the system. Basically, relevance feedback provided by the user allows exploring the archive in a dynamical way, as the system modify the search pattern in agreement with the user's feedback.

Fig. 6. A screenshot of ImageHunter. The user submitted the picture of a tiger and the system return the 23 most similar images according to a large number of low-level features.

Fig. 7. A screenshot of ImageHunter. The user performs 4 rounds of relevance feedback. At each round of relevance feedback the user marks the most relevant images among those returned by the system. The user evaluated a total of 92 images, 19 of them relevant to the user's query (relevant images are reported in the right pane)

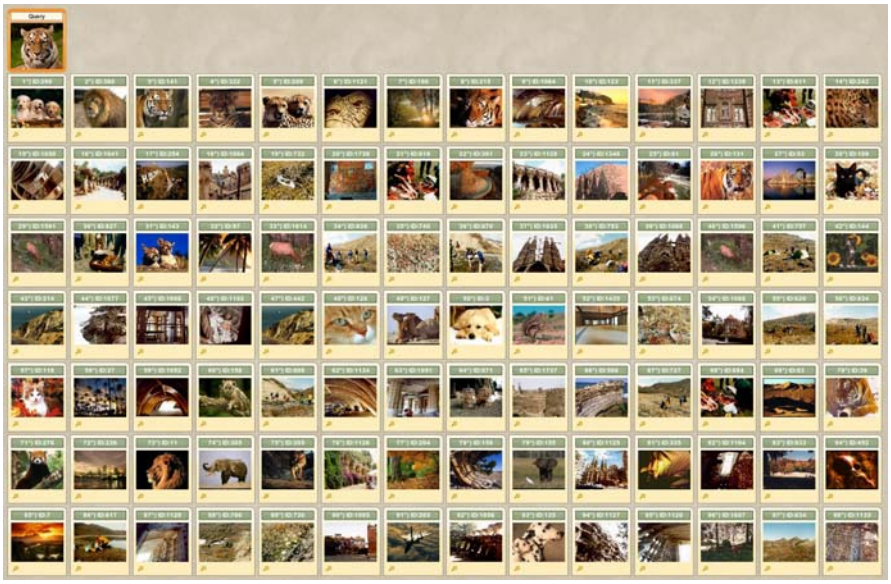

Fig. 8. If the system would present the user the first 92 images that are most similar to the query, only 14 of them are found to be relevant.

While the results are encouraging, they are limited, as this implementation does not take into account any textual description associated with the images. On the other hand these results clearly point out the need for the human in the loop, and the use of multiple features, that can be dynamically selected according to the user's feedback.

5. Conclusions

This paper aimed to provide a brief introduction on two challenging problem of the Internet Era: Intrusion Detection in computer systems, where humans leverage on the available computing power to misuse other computers, and Multimedia Content Retrieval tasks, where the humans would like to leverage on computing power to solve very complex reasoning tasks. Completely automatic learning solutions cannot be devised, as attacks as well as semantic concepts are conceived by human minds, and other human minds are needed to look for the needle in a haystack.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, second ed., Wiley-Interscience (2000)
2. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: a review. IEEE Trans. on Pattern Analysis and Machine Intelligence 22, 4–37 (2000)

3. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: A database and web-based tool for image annotation. Int. J. Comput. Vision 77, 157–173 (2008)
4. Tang, J., Chen, Q., Yan, S., Chua, T.S., Jain, R.: One person labels one million images. In: Proc. of the international conference on Multimedia, MM '10, pp. 1019–1022, ACM, New York (2010)
5. Sheldon, F.T., Vishik, C.: Moving toward trustworthy systems: R&D essentials. Computer 43, 31-40 (2010)
6. Perdisci, R., Dagon, D., Lee, W., Fogla, P., Sharif, M.: Misleading worm signature generators using deliberate noise injection. In: IEEE Symposium in Security and Privacy, pp. 15-31 (2006)
7. Barreno, M., Nelson, B., Sears, R., Joseph, A.D., Tygar, J.D.: Can machine learning be secure? In: Proc. of the 2006 ACM Symposium on Information, Computer and Communications Security, pp. 16–25, ACM, New York (2008)
8. Biggio B., Fumera G., Roli F.: Multiple classifier systems for adversarial classification tasks. In: Benediktsson, J., Kittler, J., Roli, F. (Eds.) Multiple Classifier Systems, LNCS, vol. 5519, pp. 132–141, Springer-Verlag, Berlin (2009)
9. Dalvi, N., Domingos, P., Sanghai, M.S., Verma, D.: Adversarial classification. In: Proc. of the tenth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data mining, pp. 99–108, ACM, New York (2004)
10. Skillicorn D.N.: Adversarial knowledge discovery. IEEE Intelligent Systems 24, 54–61 (2009)
11. Pavlidis, T.: Limitations of content-based image retrieval, http://theopavlidis.com/technology/CBIR/PaperB/vers3.htm (2008)
12. Maggi, F., Robertson, W., Kruegel, C., Vigna, G.: Protecting a moving target: Addressing web application concept drift. In: Kirda, E., Samesh, J., Balzarotti, D. (Eds.) Recent Advances in Intrusion Detection, LNCS, vol. 5758, pp. 21–40, Springer, Berlin (2009)
13. Stavrou, A., Cretu-Ciocarlie, G.F., Locasto, M.E., Stolfo, S.J.: Keep your friends close: the necessity for updating an anomaly sensor with legitimate environment changes. In: Proc. of the 2nd ACM workshop on Security and Artificial Intelligence, pp. 39–46., ACM, New York (2009)
14. Bonnell Harries, M., Sammut, C., Horn, K.: Extracting hidden context. Machine Learning 32, 101–126 (1998)
15. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23, 69–101 (1996)
16. Hand, D.J.: Classifier technology and the illusion of progress. Statistical Science 21, 1–14 (2006)
17. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 40, 1-60 (2008)
18. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content- based multimedia information retrieval: State of the art and challenges. ACM Trans. Multimedia Comput. Commun. Appl. 2, 1–19 (2006)
19. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Trans. on Pattern Analysis and Machine Intelligence 22, 1349–1380 (2000)
20. Sivic, J., Zisserman, A.: Efficient visual search for objects in videos. Proceedings of the IEEE 96, 548–566 (2008)
21. Huang, T.S., Dagli, C.K., Rajaram, S., Chang, E.Y., Mandel, M.I., Poliner, G.E., Ellis, D.P.W.: Active Learning for Interactive Multimedia Retrieval. Proceedings of the IEEE 96, 648-667 (2008)
22. Richter, F., Romberg, S., Hörster, E., Lienhart, R.: Multimodal ranking for image search on community databases. In: Proc. of the Intern. Conf. on Multimedia Information Retrieval, pp. 63–72, ACM, New York (2010)

23. Joshi, A., Undercoffer, J., Pinkston, J.: Modeling computer attacks: An ontology for intrusion detection. In: Hartmanis, J., Goos, G., van Leeuwen, J. (Eds.) Recent Advances in Intrusion Detection, LNCS, vol. 2820, pp. 113–135, Springer-Verlag, Berlin (2003)
24. Kompatsiaris, Y., Hobson, P. (Eds.): Semantic Multimedia and Ontologies - Theory and Applications. Springer-Verlag, Berlin (2008)
25. Biggio, B., Fumera, G., Roli, F.: Adversarial pattern classification using multiple classifiers and randomization. In: Kasparis, Y., Roli, F., Kwak, J. (Eds.) Structural, Syntactic, and Statistical Pattern Recognition, LNCS, vol. 5342, pp. 500-509, Springer, Berlin (2008)
26. Bayer, U., Comparetti, P., Hlauschek, C., Krügel, C., Kirda, E.: Scalable, behavior-based malware clustering. In: 16th Annual Network and Distributed System Security Symposium (NDSS 2009) (2009)
27. Kruegel, C., Kirda, E., Zhou, X., Wang, X., Kolbitsch, C., Comparetti, P.: Effective and efficient malware detection at the end host. In: Proc. of USENIX'09 - Security Symposium, pp. 351-366, USENIX Association, Berkeley (2009)
28. Cretu, G.F., Stavrou, A., Locasto, M.E., Stolfo, S.J., Keromytis, A.D.: Casting out demons: Sanitizing training data for anomaly sensors. In: Proc. of the IEEE Symposium on Security and Privacy, pp. 81–95 (2008)
29. Ariu, D., Tronci, R., Giacinto, G.,: HMMPayl: An intrusion detection system based on hidden markov models. Computers & Security 30, 221-241 (2011)
30. Corona, I., Giacinto, G., Mazzariello, C., Roli, F., Sansone, C.: Information fusion for computer security: State of the art and open issues. Information Fusion 10, 274–284 (2009)
31. Perdisci, R., Ariu, D., Fogla, P., Giacinto, G., Lee, W.: McPAD: A multiple classifier system for accurate payload-based anomaly detection. Computer Networks 53, 864–881 (2009)
32. Corona, I., Ariu, D., Giacinto, G.: HMM-web: A framework for the detection of attacks against web applications. In: Proc. of the IEEE International Conference on Communications (ICC '09), pp. 1–6 (2009)
33. Song, Y., Keromytis, A.D., Stolfo, S.J., Spectrogram: A mixture-of-markov-chains model for anomaly detection in web traffic. In: NDSS, 2009 (2009)
34. Li, X., Snoek, C.G.M., Worring, M.: Learning social tag relevance by neighbor voting. IEEE Trans. on Multimedia 11, 1310–1322 (2009)
35. Bertini, M., Del Bimbo, A., Serra, G., Torniai, C., Cucchiara, R., Grana, C., Vezzani, R.: Dynamic pictorially enriched ontologies for digital video libraries. IEEE Multimedia 16, 42–51 (2009)
36. Chatzichristofis, S, Boutalis, Y.: CEDD: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In: Gasteratos, A., Vincze, M., Tsotsos, J. (Eds.) Computer Vision Systems, LNCS, vol. 5008, pp. 312–322, Springer, Berlin (2008)
37. Tian, Y., Srivastava, J., Huang, T., Contractor, N.: Social multimedia computing. Computer 43, 27–36 (2010)
38. Thomee, B., Huiskes, M.J., Bakker, E., Lew, M.S.: Visual information retrieval using synthesized imagery. In: Proc. of the 6th ACM Intern. Conf. on Image and Video Retrieval (CIVR 2007), pp. 127–130, ACM, New York (2007)
39. Nguyen, G.P., Worring, M.: Interactive access to large image collections using similarity-based visualization. J. Vis. Lang. Comput. 19, 203–224 (2008)
40. Giacinto, G.: A nearest-neighbor approach to relevance feedback in content based image retrieval. In: In Proc. of the 6th ACM Intern. Conf. on Image and Video Retrieval (CIVR 2007) , pp. 456–463, ACM, New York (2007)
41. Piras, L., Giacinto, G.: Neighborhood-based feature weighting for relevance feedback in content-based retrieval. In: Proc. of the 10th Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '09), pp. 238–241, IEEE Computer Society, Los Alamitos (2009)

Vitae

Giorgio Giacinto is Associate Professor of Computer Engineering at the University of Cagliari, Italy. His main research interests are in the area of pattern recognition and its application. His main contributions are in the field of combination (a.k.a. fusion) of multiple classifiers, computer security, and multimedia retrieval. Giorgio Giacinto also contributes to researches in the fields of biometric personal authentication, video-surveillance, and remote sensing image classification. Giorgio Giacinto is author of more than seventy papers in international journals and conference proceedings, including six book chapters. He is currently serving as associate editor of the "Information Fusion" journal. Giorgio Giacinto is involved in several technical committees of international workshops and conferences on pattern recognition and applications, and regularly serves as a reviewer. He is a Senior Member of the ACM and the IEEE.