# A Decision Support System base line Flexible Architecture to Intrusion Detection

Marcello Castellano, Giuseppe Mastronardi, Angela Aprile
Mirko Minardi, Pierpaolo Catalano, Vito Dicensi, Gianfranco Tarricone
Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari, via Orabona 4
70125 Bari
Email: castellano@poliba.it
mastrona@poliba.it
a.aprile@poliba.it

***Abstract*—Becoming more competitive and more effective in the current scenes of Business and Public Administration, the organizations must be able to approach easily and quickly to the information on the customers and the external conditions of market. In other words it means to head at the improvement of the Decision Support Systems. Decision Support System is concerned with developing systems aimed at helping decision makers solve problems and make decisions.**

**In this paper, we refer to a decision support system based on flexible architecture able to discover the knowledge in a distributed and heterogeneous multi-organization environment. The system architecture has been designed according to the Service Oriented Architecture and Model View Controller design pattern. Moreover it is based on a mining engine whose purpose is to generate and to use suitable e-services for knowledge by driving the user through various stages of the Knowledge Discovery process. An Intrusion Detection application is also described.**

***Index Terms*—data mining, web mining, intrusion detection, mining engine, flexible architecture, knowledge discovery, decision support system.**

## I. Introduction

Decision Support Systems are a specific class of computerized information system that supports business and organizational decision-making activities. Decision support systems are gaining an increased importance in various domains, including business, engineering, the military, and medicine. They are especially valuable in situations in which the amount of available information is prohibitive for the intuition of an unaided human decision maker, and in which precision and optimality are of importance. Decision support systems can aid human cognitive deficiencies by integrating various sources of information, providing intelligent access to relevant knowledge, and aiding the process of structuring decisions. They can also support choice among well-defined alternatives and build on formal approaches, such as the methods of engineering, research, statistics, and security. Proper application of decision-making tools increases productivity, efficiency, and effectiveness, and gives many advantages, allowing to make optimal choices for technological processes and planning operations. On the other hand, the Knowledge Discovery Process aim is to obtain useful knowledge which will enhance future decision making and, in this sense, issues of management of this knowledge within appropriate Decision Support Systems are of crucial importance[40,41].

In security, it is axiomatic that what you can't prevent, you must detect. The goal of Intrusion Detection is to detect security violations in information systems. Intrusion detection is a passive approach to security as it monitors information systems and raises alarms when security violations are detected. Examples of security violations include the abuse of privileges or the use of attacks to exploit software or protocol vulnerabilities [6]. In Intrusion Detection research the challenge is to help dozens of companies (including several of the largest companies in the world) developing comprehensive intrusion detection systems and implementing them. Intrusion detection techniques have been developed to on any type of monitored data or resources available as part of a network or host system. Most intrusion detection techniques detect such malicious activity either at the transaction level or at the operating system (OS) level [7]. It is also shown that transaction level attacks take care of most operating system level attacks [8,9]. But there are many attacks which occur internal to the network such as by a user with lesser privileges accessing data that requires more access rights. Such attacks can be identified by analyzing the transaction logs. A transaction log contains all the transactions made on a database. By analyzing these logs most malicious activity can be identified. In a typical database accessed over a network there may be as many as one million transactions a day and any kind of computational analysis will prove to be costly and tedious. Building an intrusion detection system, IDS, is a complex task of knowledge engineering that requires an elaborate infrastructure. An effective contemporary production-quality IDS needs an array of diverse components and features, including: centralized view of the data, data transformation capabilities, analytic and data mining methods, flexible detector deployment (including scheduling that enables periodic model creation and distribution), real-time detection and alert infrastructure, reporting capabilities, distributed

processing, high system availability, scalability with system load. According this features, an Intrusion Detection application aim is to support a security manager or a security management system in making decisions about security management signalling the discovery of a possible intrusion. This discovery of knowledge is the main process in the Decision Support System that is able to control a computer system security[11,12].

There have been a number of techniques to realize a Knowledge Discovery Process using different approaches, the most recent effective strategy being Mining Techniques[10].

Data Mining is concerned with finding patterns in data which are interesting, according to some user-defined measure of interestingness, and valid, according to some user defined measure of validate. This area has recently gained much attention of industry, due to the existence of large collections of data in different formats, including large data warehouses, and the increasing need of data analysis and comprehension. In addition to mining of data stored in data warehouses, there has recently been also increased interest in Text and Web mining. Web mining is a application of data mining techniques to discover patterns from the Web. According to analysis targets, web mining can be divided into three different types, which are Web usage mining, Web content mining and Web structure mining. Web content mining is the process to discover useful information from the content of a web page. The type of the web content may consist of text, image, audio or video data in the web. Web content mining sometimes is called web text mining because the text content is the most widely researched area. The technologies that are used in web content mining are NLP, natural language processing and IR, information retrieval[5].

In this work we describe a decision support system base line flexible architecture able to discover the knowledge in a distributed and heterogeneous multi-organization environment Moreover we describe a creation process of an applicative line of the system refer to Intrusion detection problem Finally the prototypal development of the applicative line and some experimental results are presented.

This paper is composed as it follows: in the next section we introduce a background based on the service oriented architecture and the cross industry standard process for data mining. In the third section we describe a decision support system base line flexible architecture. In the fourth section we describe a creation process of an applicative line of the system refer to Intrusion detection problem. Finally, a prototype and results are presented.

## II. BACKGROUND

### A. The Cross Industry Standard Process for Data Mining

Knowledge Discovery Process in Database is a creative process which requires a number of different skills and knowledge. Currently there is no standard framework in which to carry out Knowledge Discovery Process in Database projects. This means that the success or failure of a Knowledge Discovery Process in Database project is highly dependent on the particular person or team carrying it out and successful practice can not necessarily be repeated across the enterprise. Knowledge Discovery Process in Database needs a standard approach which will help translate business problems into data mining tasks, suggest appropriate data transformations and data mining techniques, and provide means for evaluating the effectiveness of the results and documenting the experience. The CRISP-DM (CRoss Industry Standard Process for Data Mining) project addressed parts of these problems by defining a process model which provides a framework for carrying out data. Starting from the embryonic knowledge discovery processes used in early data mining projects and responding directly to user requirements, this project defined and validated a data mining process that is applicable in diverse industry sectors. This methodology makes large data mining projects faster, cheaper, more reliable and more manageable. The CRISP-DM model has only three major feedback sources, from data understanding to business understanding, from modelling to data preparation, and from evaluation to business understanding. Precisely, in CRISP-DM, six interrelated phases are used to describe the data mining process : context understanding, data understanding, data preparation, modelling, evaluation, and deployment.

- Context Understanding: this initial phase focuses on understanding the project objectives and requirements from a initial perspective, and then converting this knowledge into a data mining problem definition, and a preliminary project plan designed to achieve the objectives.
- Data Understanding: the data understanding phase starts with an initial data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, to discover first insights into the data, or to detect interesting subsets to form hypotheses for hidden information. There is a close link between Business Understanding and Data Understanding. The formulation of the data mining problem and the project plan require at least some understanding of the available data.
- Data Preparation: the data preparation phase covers all activities to construct the final dataset (data that will be fed into the modelling tools) from the initial raw data. Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. Tasks include table, record, and attribute selection, data cleaning, construction of new attributes, and transformation of data for modelling tools.
- Modelling: in this phase, various modelling techniques are selected and applied, and their parameters are calibrated to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques require specific data formats. There is a close link between Data Preparation and Modelling. Often, one realizes data problems while modelling or one gets ideas for constructing new data.

- Evaluation: at this stage in the project you have built one or more models that appear to have high quality, from a data analysis perspective. Before proceeding to final deployment of the model, it is important to more thoroughly evaluate the model, and review the steps executed to construct the model, to be certain it properly achieves the business objectives. A key objective is to determine if there is some important business issue that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results should be reached.
- Deployment: creation of the model is generally not the end of the project. Usually, the knowledge gained will need to be organized and presented in a way that the customer can use it. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process. In many cases it will be the user, not the data analyst, who will carry out the deployment steps. In any case, it is important to understand up front what actions will need to be carried out in order to actually make use of the created models [4,13,14,15]. Fig. 1 shown the six interrelated phases which are used to the data mining process in a CRoss industry standard process model.

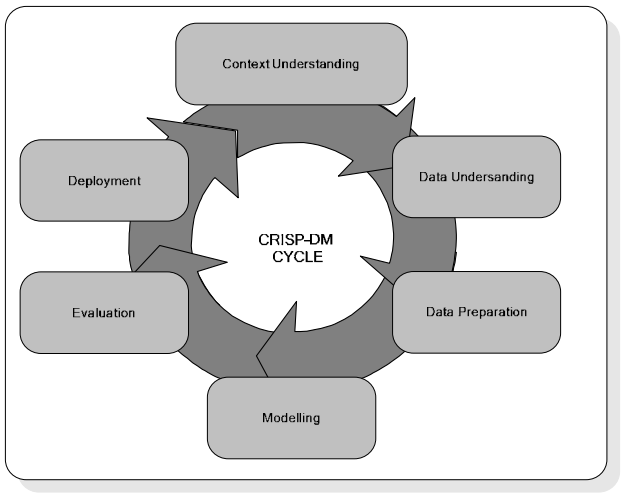


Figure 1. CRISP-DM model

### *B. The Service Oriented Architecture*

The Service-Oriented Architecture (SOA) approach is the latest of a long series of attempts in software engineering trying to encourage the reuse of software components, the SOA provides a scheme to design distributed and orchestrated applications. The SOA is a strategy and a flexible and scalable technical framework for supporting open standards and open application interfaces. Open applications that are built on an open architecture promise to be significantly less expensive to implement and to operate. The architecture is called service-oriented because applications, and functions within applications, appear on the network as a loosely coupled set of services. The term service-oriented architecture refers to a style of building reliable distributed systems that deliver functionality as services, with the additional emphasis on loose coupling between interacting services. A service is a software component that can be accessed via a network to provide functionality to a service requester. Services have the following characteristics:

- Services may be individually useful, or they can be integrated and composed to provide higher-level services and this promotes re-use of existing functionality.
- Services communicate with their clients by exchanging messages: they are defined by the messages they can accept and the responses they can give.
- Services can participate in a workflow, where the order in which messages are sent and received affects the outcome of the operations performed by a service and this notion is defined as service choreography.
- Services may be completely self-contained, or they may depend on the availability of other services, or on the existence of a resource such as a database.
- Services advertise details such as their capabilities, interfaces, policies, and supported communications protocols. Implementation details such as programming language and hosting platform are of no concern to clients, and are not revealed.
- Service in an SOA is an application or function with well-defined interfaces that is packaged as a reusable component for use in a business process.

Simply stated, in an SOA, business processes appear as a set of separate components that can be joined and choreographed to create composite applications and processes and so there are following benefits: flexibility, that is to say, a service can be located on any server, and relocated as necessary, as long as it maintains its registry entry, prospective clients will be able to find it; scalability, that is to say, services can be added and removed as demand varies; replace ability, that is to say, provided that the original interfaces are preserved, a new or updated implementation of a service can be introduced, and outdated implementations can be retired, without disruption to users [1,2,3,14,15].

## III. A DECISION SUPPORT SYSTEM BASE LINE FLEXIBLE ARCHITECTURE

The specialized architecture supplies, in its globalise, a Decision Support System. It is defined as interactive, flexible and adaptable computer-based information system that aids users to formalize domain knowledge so that it is amenable to mechanized reasoning. In this sense the referred Decision Support System is a knowledge based system.

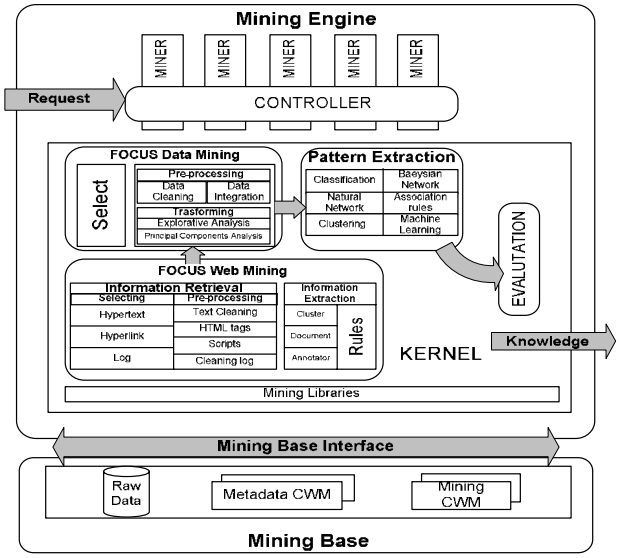


Figure 2. Decision Support System architecture

As shown in Fig. 2 our architecture is composed by two functional components: Web Mining and Data Mining which interfacing them through a Web Warehouse, that is a Data Warehouse filled with data coming from a textual analysis of the content of documents coming from web, that is to say, through a phase of Text Content Mining[37].

Web Mining effects the information retrieval from the WWW and extracts intrinsic information from the content of a document or a cluster of documents. The output of this component is the filling of the Web Warehouse which will contain useful information for the classification of the document through the analysis of the content of the same. Web Warehouse becomes, to this point, the data source for Data Mining component.

Data Mining applies mining techniques to the content of the Web Warehouse. The Data Mining results allows to the final actor of the system to get Knowledge Discovery which can be exploited then by the same for the most varied applications.

### A. *Service Oriented Web Mining Architecture*

The information coming from Web represents an important role in Knowledge Discovery Process. To examine web contents and to extract useful information to a purpose through techniques of Mining is scientifically called Web Mining.[38]. Fig. 3 shown a Web Mining Process.

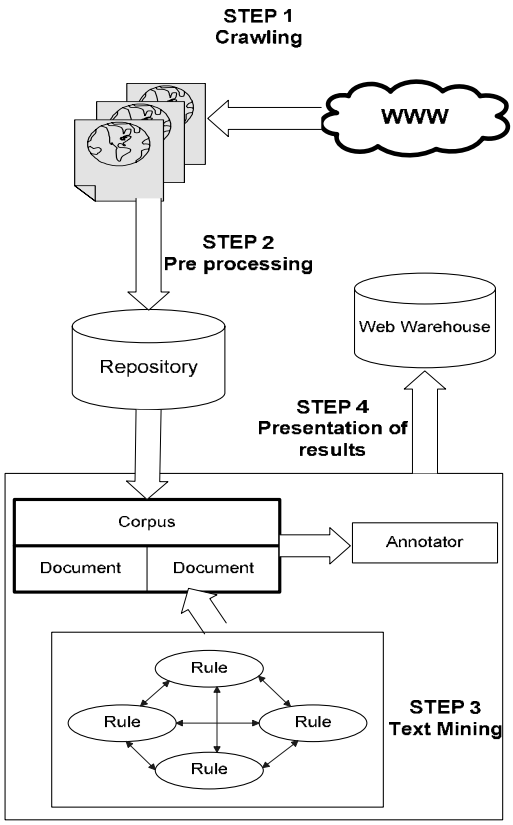


Figure 3. Web Mining Process

The whole process of Web Mining has been conceived and projected according to the service oriented architecture (SOA) of which we illustrate the scheme in Fig. 4.

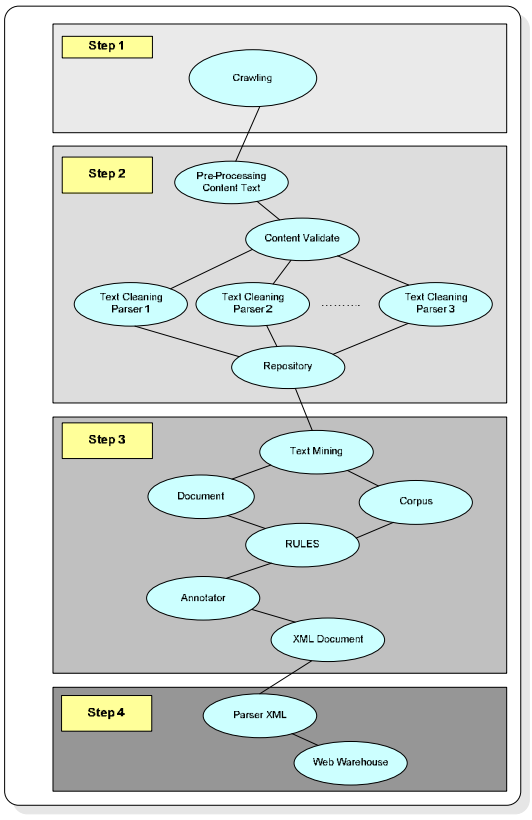


Figure 4. Web Mining Model

Web Mining Process, in our architecture, is implemented through four steps.

The first step foresees the recovery of the useful information from Web. This information consists in textual contents present in web pages. The references solution covers the first step of Web Mining process [34,35], dealing with automatic retrieval of all relevant documents and ensuring at the same time that the not-relevant ones macaws fetched as few as possible. The recovery of the useful information is effected through crawling departing from one or more URL. The crawling is implemented by a Web Crawler, as shown in the Fig.5. Web Crawler's architecture fits the guides lines of to Focused Crawler[36], as it is designed to only gather documents on to specific topic, thus reducing the amount of network traffic and downloads. It is composed from four components: Master, Slaves, Scribes, H-Information. Every component has a specific assignment. H-Information (Human Information) is the point of departure of the whole process of crawling. It allows the system administrator to interact with Web Crawler inserting a list of sites Web believed of particular importance for the service that the system must realize. Master is the kernel of Web Crawler that allows the start and the management of the whole process of recovery of the information. Master calls Slaves and passed them URLs of the pages in accord with the built list by the H-Information. Slaves accede to the Web for crawling the pages which are associated to URLs indicated by Master. Different *Slaves* can operate at the same time rendering more efficient, reducing the answer times and avoiding simultaneous accesses to the same resources [39]. Slaves call Scribes [28,29,30,31,32,33]. Scribes receive Web pages crawled from Slaves. For each Scribe there is a Slave. Scribe parses entire page and captures different information according to the type of the examined document.

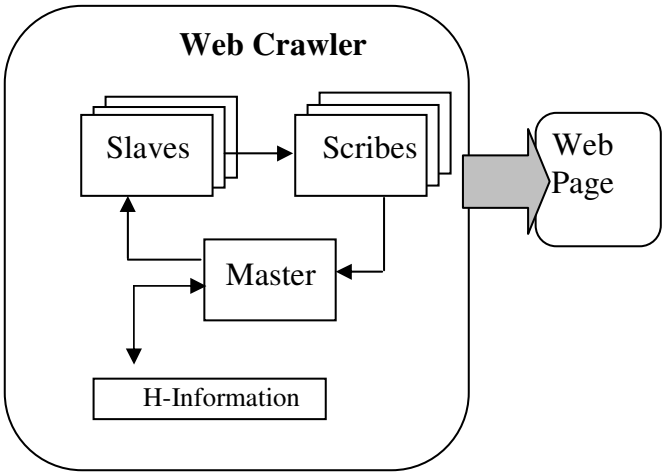


Figure 5. Web Crawler Architecture

The second step is the pre-processing. It foresees the creation of a repository of the information from the web during the execution of the first step. During this phase it is had to provide to effect a control and a cleaning (Text Cleaning) of the pages reached during the first step. The definition of Template, which contains keywords of the universe, allows a choice on the single content of the examined web page. If it contains information that correspond to one of the Template defined for the system in matter, it is accepted and on it a cleaning of the content is effected to eliminate all what to be held "garbage", that is to say tag, banner, information not tightly connected to the content of the same page. Otherwise the page is discarded. If the page has found correspondence with one of the Template defined for the system, the " polished up" information of the page are memorized in a repository.

The third step is the fundamental step on which the whole Web Mining process founds him. In this step a first phase of Information Extraction and the following phase of Text Mining are effected. It founds on the use of Text Mining tools that receiving in input the data which are contained in the repository of the second step and they are able to extract "intrinsic information" contained in the document or in all documents.

A general Text Mining scheme founds its functioning on Document, Corpus, Rules and Annotators. A Document is defined as a single aggregation of text coming from a same source, a Corpus is all Documents, that is, a Document Clustering, coming from one or more web sources, the Rules are the most important part of Text Mining and their definition from the programmer will allow to get or not a good result of the whole Web Mining process. Before applying the rules, Text Mining software handles the phase of Information Extraction effecting a tokenization and lemmatization of the text that it will allow an accurate analysis of the document during the Text Mining. Subsequently to the Information Extraction, the application of one or more rules on a Document or on a Corpus will provide the extraction of "hidden information" that will be made available to the system in XML format, the useful format to fourth step. The rules can interact between them to create a more general rule. The accuracy in the discovery of intrinsic information of the document, is better if more rules are defined. But it is important to not define a lot of rules for avoiding the problem of "too much accurate", because in this case the process results could not useful. The discovery information in this phase is momentarily memorized in the Annotators which have the assignment of memorize all useful information in a Document or in a Corpus. The further analysis on the Annotators, that is to say, elimination of the duplicates, elimination of the errors, etc. are the start of the discovery of the real information called Skill. The Skills are keywords which allows to the system to have a general vision of the semantic content of the Document or the Corpus[22].

Fig.6 shown as the application of Rules on the Keywords (nodes) allows the individualization of Annotators.

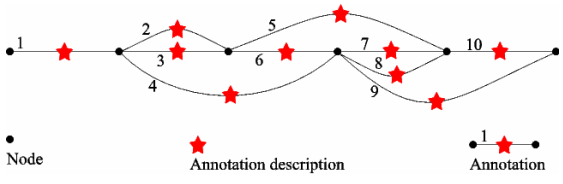


Figure 6. Rules and Annotators scheme

The fourth is last step of the process, it foresees the phase of presentation of the obtained results. In our architecture, the information drawn out during the execution of the third step is stored in a second repository. This repository created during this phase will follow the rules of the Web Warehousing according to the Whoweda schemas. The repository are composed by URLs and nodes. URLs are the reference URLs of the document, that is to say, "referenced from" and "it references who" whereas the nodes are the skills found in the third step. To fill this repository is necessary to effect a parsing of the result in XML supplied by software during the third phase for recover the necessary information. Use of Whoweda schemas is fundamental in the implementation of Web Warehouse because our system is directed to the execution of Data Mining on the data produced by the whole Web Mining process. Following Whoweda schemas is possible realize more views and more schemas on the whole Web Warehouse. This allows fast interrogation of Web Warehouse in the Data Mining phase[34]. The assignment to effect Data Mining and therefore to define views and schemas for the interrogation is submitted to the opportune programming of a Miner in Mining Engine.

### *B. Service Oriented Data Mining Architecture*

In the previous paragraphs, have been introduced CRISP-DM process and SOA. Adopting these methodologies, in this paragraph, we present a flexible architecture that modularizes the full lifecycle of knowledge discovery process into reusable components in order to create the interaction among these different components of a mining architecture and the assembling of their logic into a manageable application, these components are called "Miners". In this sense all operations in CRISP-DM steps are defined as Miners and all Miners are autonomous. CRISP-DM data mining methodology is described in terms of a hierarchical process model, consisting of sets of tasks described at four levels of abstraction, from general to specific: phase, generic task, specializing task and process instance.

At the first level, the Knowledge Discovery process is organized into a number of phases; each phase consists then of several generic tasks.

The second level, meant to be general enough to cover all possible Data Mining situations and applications. Each Miner has to be as complete and stable as possible, and this particularly to make the model valid for yet unforeseen developments such as new modelling techniques or upgrading.

The third level is where is described how miners should be carried out in certain specific situations.

At the top level, the data mining process is organized in six phases. According to CRISP-DM, the data mining context drives mapping between the generic level and the specializing level, with four different dimensions: application domain, data mining problem type, technical aspects and techniques/tool to be applied. An example of Miner model for the phase of pre-processing and transformation is shown in Figure 7 in which is clear how a generic data cleaning task at the second level could be implemented in different situations, such as the cleaning of numeric values or categorical values. Adopting the service oriented architecture approach, Miner Orchestration represents the layer supporting the composition of complex processes by constructing different data mining specializing tasks, that is to say miners.

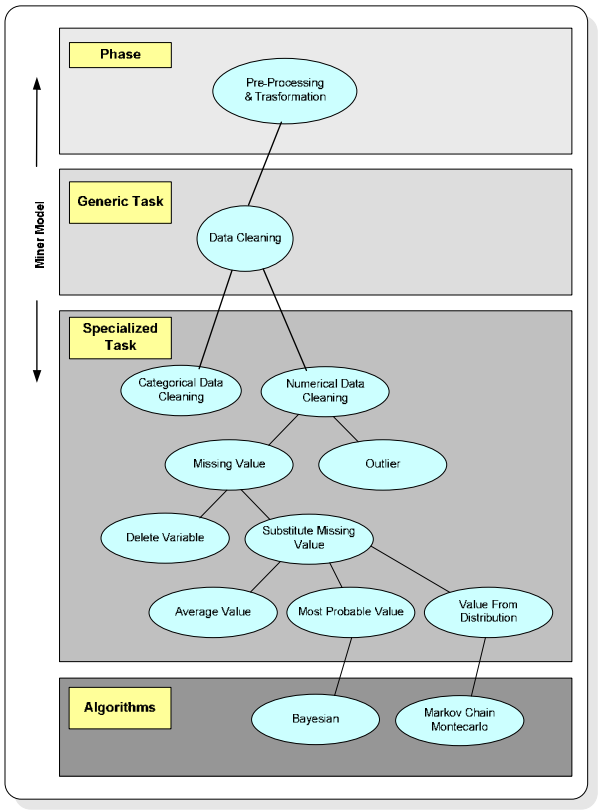


Figure 7. Miner Model

In particular Miner Orchestration is defined as the process able to create a workflow of building blocks and in this sense Miner Orchestration provides more complex functionality in the Knowledge Discovery Process. Miner Orchestration must be dynamic, flexible, and adaptable to meet the changing needs of a organization. To achieve these goals we propose a our design of Mining Engine Architecture. It is mainly composed by a Controller, more Miners and a Kernel.

The Controller handles the process flow, calling the appropriate Miners and determining the next steps to be complete. Miners are building blocks which represents the operations for realize a service.

The Kernel is the core of the Mining Engine where the new mining models has built. Finally, the Mining Base is the knowledge repository of the whole architecture and its functions are those of repository for raw data, knowledge metadata and mining mode [4].

## IV. THE SPECIALIZING ARCHITECTURE: THE KNOWLEDGE DISCOVERY PROCESS FOR INTRUSION DETECTION

The primary assumptions of intrusion detection are: user and program activities are observable, for example, via system auditing mechanisms; and more importantly,

normal and intrusion activities have distinct behaviour. Intrusion detection therefore includes these essential elements:

- Resources to be protected in a target system, for example, network services, user accounts, system kernels, etc.
- Models that characterize the normal or legitimate behaviour of the activities involving these resources.
- Techniques that compare the observed activities with the established models. The activities that are not normal are flagged as intrusive.

Two different approaches of data mining based intrusion detection techniques have been proposed up to now: misuse detection and anomaly detection.

In misuse detection, each instance in a data set is labelled as 'normal' or 'intrusion' and a learning algorithm is trained over the labelled data. These techniques are able to automatically retrain intrusion detection models on different input data that include new types of attacks, as long as they have been labelled appropriately. Unlike signature-based intrusion detection systems, models of misuse are created automatically, and can be more sophisticated and precise than manually created signatures. A key advantage of misuse detection techniques is their high degree of accuracy in detecting known attacks and their variations. Their obvious drawback is the inability to detect attacks whose instances have not yet been observed.

Anomaly detection, on the other hand, builds models of normal behaviour, and automatically detects any deviation from it, flagging the latter as suspect. Anomaly detection techniques thus identify new types of intrusions as deviations from normal usage. While an extremely powerful and novel tool, a potential draw-back of these techniques is the rate of false alarms. This can happen primarily because previously unseen, yet legitimate, system behaviours may also be recognized as anomalies, and hence flagged as potential intrusions [19,20]. Precisely, the main problem related to both anomaly and misuse detection techniques resides in the encoded models, which define normal or malicious behaviours.

Some recent intrusion detection system, IDS, provide mechanisms to write new rules that extend the detection ability of the system, such rules are usually hand-coded by a security administrator. More intelligent IDS are able to automatically build a set of models. All the techniques used to build intrusion detection models need a proper set of audit data. Data must be labelled as either "normal" or "attack" in order to define the suitable behavioural models that represent these two different categories. One of the main issues related to pattern recognition in intrusion detection is the use of a proper data set, containing user profiles on which the data mining processes work in order to extract the patterns. In principle, an efficient set of tagged data for the detection has to contain all of the possible user behaviours [21].

Raw audit data is first pre-processed into records with a set of "intrinsic", that is to say, general purposes, features, e.g., duration, source and destination hosts and ports, number of bytes transmitted, etc. Data mining algorithms are then applied to compute the frequent activity patterns, in the forms of association rules and frequent episodes, from the audit records. Association rules describe correlations among system features what shell command is associated with what argument; whereas frequent episodes capture the sequential temporal co-occurrences of system events what network connections are made within a short time-span. Together, association rules and frequent episodes form the statistical summaries of system activities. When sufficient normal audit data is gathered, a set of normal frequent patterns can be computed and maintained as baseline[23].

Several types of algorithms are particularly useful for mining audit data:

- Classification places data items into one of several predefined categories. The algorithms normally output classifiers, for example, in the form of decision trees. In intrusion detection one would want to gather sufficient normal and abnormal data and then apply a classification algorithm to teach a classifier to label or predict new, unseen audit data as belonging to the normal or abnormal class.
- Link analysis determines relations between fields in the database records.
- Correlation of system features in audit data, for example, between command and argument in the shell command history data of a user, can serve as the basis for constructing normal usage profiles.
- Sequence analysis models sequential patterns.

These algorithms can determine what time-based sequence of audit events occur frequently together. These frequent event patterns provide guidelines for incorporating temporal statistical measures into intrusion detection models. For example, patterns from audit data containing network-based denial of service attacks suggest that several per host and per service measures should be included.

### *A. Defining Miners for Intrusion Detection*

For the implementation of Intrusion Detection in the reference architecture, SOA, our contribute is not as much as to define how the service works, as it is a well-known problem, but to demonstrate how it can be simply designed through the creation of an Orchestration of already implemented Miners as building blocks.

According to the CRISP model, the process of creation of Miners Orchestration can be divided into steps: Data Pre-processing and Transformation, Modelling, Evaluation and Deployment of Result, with each step made up by one or more Miners collaborating with one another.

More in detail, the Data Pre-processing and Transformation step firstly aims at understanding which data are necessary for the implementation.

In order to solve the issue related to data set building, two main approaches are possible: the former relies on simulating a real-world network scenario, the latter builds the set using actual traffic. The first approach is usually adopted when applying pattern recognition techniques to intrusion detection. The most well-known dataset is the

KDD Cup 1999 Data. This set was created in order to evaluate the ability of data mining algorithms to build predictive models able to distinguish between a normal behaviour and a malicious one. The KDD Cup 1999 Data contains a set of "connection" records coming out from the elaboration of raw data. Each connection is labelled as either "normal" or "attack". The connection records are built from a set of higher-level connection features that are able to tell apart normal activities from illegal network activities. But now numerous research works analyze the difficulties arising when trying to reproduce actual network traffic patterns by means of simulation. Actually, the major issue resides in effectively reproducing the behaviour of network traffic sources. Collecting real traffic can be considered as a viable alternative approach for the construction of the traffic data set. Although it can prove effective in real-time intrusion detection, it still presents some concerns. In particular, collecting the data set by means of real traffic needs a data pre-classification process. In fact, as stated before the pattern recognition process needs a data set in which packets are labelled as either "normal" or "attack". Indeed, no information is available in the real traffic to distinguish the normal activities from the malicious ones in order to label the data set [25,26,27,28].

After this step in which data have been collected, they must be prepared, cleaned, and stored in a data warehouse for the pre-processing and transformation. This latter consists in the creation and cleaning of the dataset that will represent the input for the following analysis.

The Modelling step implements data mining techniques for the creation of the mining model of the Intrusion Detection service. A number of data mining techniques have been found useful in the context of misuse and anomaly detection. Among the most popular techniques are pattern recognition, association rules, clustering, support vector machines, decision tree, classification and frequent episode programs.

The end of this step is developing a framework of applying these data mining techniques to build intrusion detection models. This framework consists of one of these techniques as well as a support environment that enables system builders to interactively and iteratively drive the process of constructing and evaluating detection models. The end product is concise and intuitive classification rules that can detect intrusions [29].

Finally, the Evaluation and Deployment of Result step importance lies in the assessment of the statistically acceptable, trivial, spurious, or just not relevant to the application generated rules. One way to validate the rules discovered is to use validation operators that let a human expert validate large numbers of rules at a time with relatively little input from the expert. Validation operators can be Similarity-based rule grouping, Template-based rule filtering, Redundant-rule elimination.

Our study has shown that Data Mining can be used to support and partially automate the investigation in Intrusion Detection process. More over not all data mining techniques are suitable for heterogeneous environments. To overcome these limitations, in this paper, we propose a SOA approach, in this way, we retrain each dataset with the best techniques in order to find all possible alarms. In the other hand this approach is more expensive because requires a great number of mining interactions. Thus, when designing and configuring such a system, a preliminary phase of trade off evaluation is mandatory.

### *B. Orchestrating Miners for Intrusion Detection*

In previous paragraph have been introduced Miners and Orchestration as theoretical approach for the realization of an IDS; in our experiments we have implemented them by using standards as Java, Web Services and BPEL4WS.

Fig. 8 shows the BPEL model that represents the orchestration among the different Miners that collaborate to accomplish the process of Intrusion Detection service.

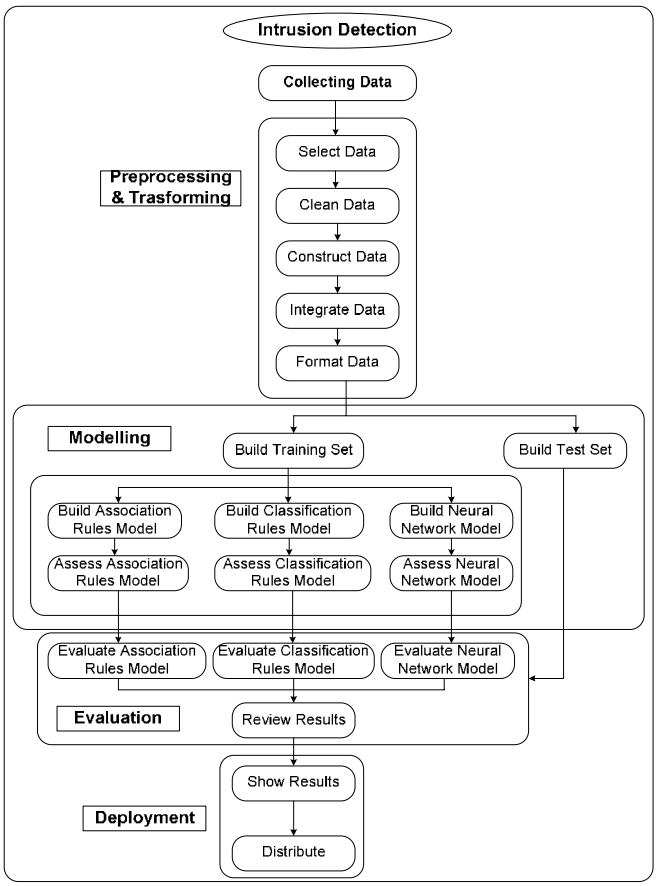


Figure 8. Intrusion Detection Miners Orchestration

As shown in the previous figure, the workflow is organized in the main phases of Selecting, Data Pre-processing and Transformation, Modeling, Evaluation and Deployment of Result. Initial set of data are retrieved, selected and then pre-processed in order to build the Training Set necessary for the further step of the mining Modeling. Built models are tested and evaluated to assess their value and reliability and finally presented to the decision maker who analyzes the effectiveness of the knowledge extracted. The following rows present the essential XML code of the relative BPEL process:

```
<?xml version="1.0" encoding="UTF-8"?>

<process expressionLanguage="Java"

name="IntrusionDetectionProcess"

<sequence name="MainSequence">
        <receive
name="CollectingData">...</receive>
        <flow
name="PreProcessingTrasformation">...
            <invoke
name="SelectData">...</invoke>
            <invoke
name="ConstructData">...</invoke>
            <invoke
name="FormatData">...</invoke>
            <invoke
name="CleanData">...</invoke>
            <invoke
name="IntegrateData">...</invoke>
        </flow>
        <flow name="Modelling">...
            <invoke
name="BuildTrainingSet">...</invoke>
            <flow name="Build&Assess">...
                <invoke
name="BuildAssociationRulesModel">...</invoke>
                <invoke
name="BuildClassificationRulesModel">...
                </invoke>
                <invoke
name="BuildNeuralNetworkModel">...</invoke>
                <invoke
name="AssessAssociationRulesModel">...</invoke
>
                <invoke
name="AssessClassificationRulesModel">...
                </invoke>
                <invoke
name="AssessNeuralNetworkModel">...</invoke>
            </flow>
            <invoke
name="BuildTestSet">...</invoke>
        </flow>
        <flow name="Evaluation">...
            <invoke
name="EvaluateAssociationRulesModel">...</invo
ke>
            <invoke
name="EvaluateClassificationRulesModel">...
            </invoke>
            <invoke
name="EvaluateNeuralNetworkModel">...</invoke>
            <invoke
name="ReviewResults">...</invoke>
        </flow>
        <flow name="DeploymentofResults">...
            <invoke
name="ShowResults">...</invoke>
            <reply
name="Distribute">...</reply>
        </flow>
    </sequence>
</process>
```

The sample Intrusion Detection process has been created through our reference architecture; in particular, the collection of Miners has been implemented through Java Web services and Weka technology, the Controller through a BPEL4WS Engine and the Actions through BPEL4WS process definition files [4].

## V. PROTOTYPE

All what it has been written in the preceding paragraphs assumes some real characteristics in experimentation phase, in fact it is decided to experiment there studies with the realization of a prototype that puts in evidence as all the phases of a process of knowledge extraction from not structured information, like those present on the web, can concretely be applied to Intrusion Detection issue.

The realized prototype foresees the possibility to individualize an intrusion inside a system through the analysis of the web traffic produced by a user. Our prototype will be able to notice and to signal a possible intruder to the system administrator, wherever for intruder it is intended a user that has pass the first phase of authentication (if foreseen) in the system but is believed "not correct" through the analysis of the web visited contents. The output of the prototype is the input for a Miner of the Engine. The input consists in a repository that contains information around the context of the visited pages by the consumer. A temporary user profile is created from these information, this profile is compared to the user profile memorized inside the system. From the comparison illegal accesses can be found, that is, survey of intrusions. The process for the creation of the temporary user profile foresees various phases each with a well precise assignment. The first phase consists of tracing a chronology of the visited pages. The URLs are memorized in a log file located in the home directory of the user. The first component of the prototype, that has the assignment to extract the useful content from the visited pages, accesses the log file. The prototype is based on an multithread algorithm that allows parallel appeals. This allows to analyze more pages contemporarily to optimize the run-time of the whole process of extraction. The access to the pages happens in a entirely transparent way to the consumer that keeps on sailing without any kinds troubles (decelerations, pop-up, etc.). Once reached the URL it is passes to the following phase that consists of filtering the content of the pages. In other words, the pages come "cleaning up" of the whole useless content. This is a very important and delicate phase because it influences the result of Text Mining. For this purpose the prototype is endowed with parser both for structured information (XML, RSS, etc.) both for not structured information or semi-structured information (HTML, etc). In terms of code the parsers are methods of a Java classes. The prototype recognizes the type of page, that is, the type of contained information in the page and it recalls the relative parser. The filtering consists, for instance, in the elimination of tag, publicity, scripts, links from the source of the page. It is eliminated the whole information that is not useful to the goals of the individualization of the context of the page. The greater difficulty, in this phase, is the parsering of pages which contain not structured information. In this kind of pages, that doesn't have a defined and rigid structure, the quality of the extracted useful information depends from the effectiveness and the efficiency of the parser. The following phase to the filtering consists in the memorization of the drawn out information inside a repository. A Text Mining tool makes reference to this repository. The content of the repository has a well defined structure that allows to the Text Mining tool a better individualization of the information. It consists in a database which contain

specific fields for the type of drawn out information. After to created the repository of the web contents it is passes to the phase of analysis of the contained documents in the repository. This analysis is effected through a Text Mining software. We have to use GATE 3.1 as Text Mining tool. It founds its functioning on Corpus, Document, Annotator and Rules, whose functionality has been explained in the previous paragraphs. For the programming of Gate, and in particular, for the planning of its rules, Java is used as programming language with relative API that allow to use of the GATE functionalities. The sequence of application of rules on the corpus has a meaningful and fundamental importance, in fact if our phase of Text mining is defined by three rules that we call R1, R2, R3, a different temporal application of these three rules could supply different results in the process of knowledge extraction. In our prototype it is tried to individualize the topology of web pages which are visited by the consumer, for this issue two rules are been implemented. The first tries to understand if the content of the document is textual or multimedia type. If it is textual type, the second rule is applied, this second rule looks for to catalogue the content of the text, using a matching of the text tokens with a list of keywords. Rules are based on lists of keywords which are opportunely defined for the Text Mining context. The following is an example of Java rule that is defined for our prototype:

```
Phase:      Skill
Input: Token Lookup
Options: control = appelt

Rule: SkillDocument

(
 {Lookup.majorType == ContentDocument}
):skill
-->
{
gate.AnnotationSet skill =
(gate.AnnotationSet)bindings.get("skill");
gate.Annotation skillAnn =
(gate.Annotation)skill.iterator().next();
gate.FeatureMap features =
Factory.newFeatureMap();
features.put("rule", "Skill");
outputAS.add(skill.firstNode(),
skill.lastNode(), "Skill", features);

}
```

This rule, fundamental in our prototype, called SkillDocument, individualizes the contents of the document making a matching on all tokens of the same document through the list. This list, called ContentDocument, contains all the Keywords able to catalogue a document using its content. The result of the application of all these rules to the document or to the Corpus (we remember that a corpus is a collection of documents) is a XML file that underlines, among the TAGs, the skill which is individualized by the rule. The name of the TAG is defined inside the rule in the following code line:

```
outputAS.add(skill.firstNode(),
skill.lastNode(), "Skill", features);
```

which is, in this case, Skill. The output by the process of Text Mining is a XML document in which, among the TAGs, there are the Skill which are individualized in the whole process. This allows a parsing of the XML produced file and to supply, easily, the second repository which will contain, for every web content visited by the user, all the Skills that identify the same content (sport, finance ,e-mail, on line auction, etc...). It will be assignment of a Miner, opportunely projected in the Mining Engine, to examine the repository that is produced in this phase, to analyze the contents and to notice the possible intrusion. The miner will verify if the contents of the web pages, visited by the user, corresponding to the user profile and, if this isn't true, will signal a possible intrusion in the system. It will be the whole system to take the opportune decisions around the received signalling. A typical signalling could be 'to leave from the user profile'. But what means 'to leave from the user profile'? Usual a user, that visit WWW documents interesting for him, always , turns his attention to the same kind of documents. It doesn't exclude the fact that a user can decide to also visit web page which don't have connection with his profile but in such case it can affirm, with good probability, that, after a brief period of time, user will already return to visit contents to him known. Then the miners that examines the repository will have two assignments: the first will be to verify if the visited pages have a informative content corresponding to the user profile, holding the 50% threshold as a good threshold for the signalling of the possible intrusion, and in such case to signal the intrusion to the system. The second assignment will be to hold adjourned the user profile during the time. In fact in the time a user can change his preferences and the profile must be modified corresponding to the visited contents. It is reasonable to think that the preferences of a user slowly change in the time with thresholds that leave them of 20-30% respect to the actual profile. Fig. 9 shown a graphic interpretation of the prototype that implements an Intrusion Detection System referred to a DSS. The underlined areas represent the behaviour of the user. Green area is subtended by the curve that represents the ideal user profile, in this area the user is classified by the system as valid. Yellow area represents a difference respect to the ideal profile between 25% and 50%. The system classifies a user behaviour in this area as anomalous signalling a possible intrusion. Red area represents a difference respect to the ideal user profile >50%. A user behaviour in this area is classified by the system as Intrusion Detection.

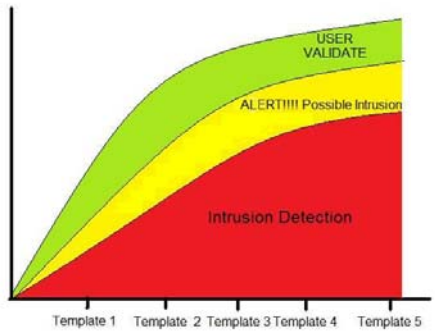


Figure 9. Graphic Interpretation of the Prototype of ID.

## VI. CONCLUSION

In this paper has been presented a Knowledge discovery process architecture able to build Intrusion Detection application in a modular and flexible way. To this purpose, the guide lines of the SOA and the Orchestration model have been considered as a way to manage a workflow of reusable building blocks with well defined tasks and able to interoperate with one another for the creation of new services. The main advantages offered by this architecture are the quick designing of a process according to one's own business needs and the creation of new flexible services without resorting to substantial changes.

## ACKNOWLEDGMENTS

The authors acknowledge the financial support provided by the Italian Ministry of Education, University and Research which has made possible the realization of this work as result of our research activities.

## REFERENCES

[1] M. Castellano, N. Pastore, F. Arcieri, V. Summo, and G. Bellone de Grecis, "A Knowledge Center for a Social and Economic Growth of the Territory", *IEEE Computer Society Press*, International Conference On System Sciences, Big Island Hawaii, 3-6 January 2005.

[2] M.Castellano, N. Pastore, F. Arcieri, V. Summo, and G. Bellone de Grecis, "An e-Government Cooperative Framework for Government Agencies", *IEEE Computer Society Press,* International Conference On System Sciences, Big Island Hawaii, 3-6 January 2005.

[3] M.Castellano, N.Pastore, F.Arcieri, V. Summo, and G. Bellone de Grecis, "A Flexible Mining Architecture for Providing New E-Knowledge Services", *IEEE Computer Society Press,* International Conference On System Sciences, Big Island Hawaii, 3-6 January 2005.

[4] M. Castellano, N. Pastore, F. Arcieri, V. Summo, and G. Bellone de Grecis, "Orchestrating Knowledge Discovery Process", *E-Service Intelligence: Methodologies, Technologies and Application*, Springer, pp 447-496.

[5] M. Castellano, F. Fiorino, F. Arcieri, V. Summo, and G. Bellone de Grecis, "A Web Mining Process for e-Knowledge Service", *E-Service Intelligence: Methodologies, Technologies and Application*, Springer, pp 447-496. A Web Mining

[6] K. Julisch, "Data mining for Intrusion Detection: a Critical Review", *IBM Research*, Zurich Research Laboratory.

[7] M. El-Sayed, C. Ruiz, and E. A. Rundensteiner, "FS-Miner: efficient and Incremental Mining of Frequent Sequence Patterns in Web logs", *ACM WIDM'04*, Washington DC, November 2004, pp. 12-13.

[8] R. Kemmerer, and G. Vigna, "Hi-DRA: Intrusion Detection for Internet Security", *IEEE Proceeding*, October 2005, vol. 93 pp. 1847-1857.

[9] F. Valeur, D. Mutz, and G. Vigna, "A Learning Based Approach to the Detection of SQL Attacks", *Conference on Detection of Intrusion and Malware and Vulnerability Assessment (DIMVA*), Vienna Austria, July 2005.

[10] W. Lee, SJ Stolfo, and KW Mok, "Data mining approaches for intrusion detection", *Proceedings of the 7th USENIX Security Symposium*, 1998.

[11] Y. Cai, D. Clutter, G. Pape, J. Han, M. Welge, and L. Auvil, "MAIDS: Mining Alarming Incidents from Data Streams, *ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'04)*, ACM Press, New York, 2004, pp. 919-920.

[12] V. Chatzigiannakis, G. Androulidakis, and B. Maglaris, "A Distributed Intrusion Detection Prototype Using Security Agents", HP OpenView University Association, 2004.

[13] www.CRISP.com.

[14] Service Oriented Architecture White Paper.

[15] R. Wirth, and J. Hipp, "CRISP-DM: Towards a Standard Process Model for Data Mining", *DaimlerChrysler Research & Technology FT3/KL PO BOX 2360 89013 Ulm*, Wilhelm-Schickard-Institute and University of Tübingen Sand 13, Germany.

[16] L. Srinivasan , and J. Treadwell, "An Overview of Service-oriented Architecture, Web Services and Grid Computing", HP Software Global Business Unit, November 2005.

[17] D.E. Denning, "An Intrusion Detection Model", *IEEE Trans-actions on Software Engineering*, 1987.

[18] M. Joshi, V. Kumar, and R. Agarwal, "Evaluating Boosting Algorithms to Classify Rare Classes: Comparison and Improvements", *First IEEE International Conference on Data Mining*, San Jose CA, 2001.

[19] M. Esposito , C. Mazzariello, F. Oliviero, S.P. Romano, and C. Sansone, **"**Real Time Detection of Novel Attacks by Means of Data Mining", *ICEIS Conference*, ACM, 2005.

[20] W. Lee, and S. J. Stolfo, "Combining Knowledge Discovery and Knowledge Engineering to Build IDS", *Computer Science*, Columbia University, New York.

[21] W. Lee, "A data mining framework for building intrusion detection models", *IEEE Symposium on Security and Privacy*, Berkeley California, May 1999, pages 120.132.

[22] GATE – General Architetcture for Text Engineering, http://gate.ac.uk/

[23] S.J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. Chan, "Cost-based modeling for fraud and intrusion detection results from the jam project", *Proceeding of the 2000 DARPA Information Survivability,* DISCEX '00, 2000.

[24] J. McHugh, "Testing intrusion detection systems", *A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory,* ACM Transactions on Information and System Security 3**,** 2000, pp. 262–294.

[25] V. Paxson, and S. Floyd, "Difficulties in simulating the internet", *IEEE/ACM*, Transactions on Networking 9, 2001, pp. 392–403.

[26] M. Mahoney, "A Machine Learning Approach to Detecting Attacks by Identifying Anomalies in Network Traffic", Florida Istitute of Technology, 2003.
[27] W. Lee, and S. J. Stolfo, "Mining Audit Data to Build ID Model", Columbia University, New York.
[28] Richard D. Hackathom, "Web Farming for the Data Warehouse. The Morgan Kauimann Series in Data Management Systems", Jim Gray, Series Editor. November 1998. ISBN 1-55860-503-7.
[29] L. Faulstich, M. Spiliopoulou, V. Linnemann. WIND: "A Warehouse for Internet Data", *Proceedings of British National Conference on Databases* , pp. 169-183, London, 1997.
[30] R. Kimball and R. Merz: "The Data Webhouse Toolkit, Building the Web-Enabled Data Warehouse", John Wiley & Sons, January 2000.
[31] MARINHO, Leandro Balby Marinho, Girardi Rosario: "Mineração da Web", *Revista Eletrônica de Iniciação Cientfica*, São Luiz, Jun. 2003.
[32] Adriana Marotta, Regina Motz, Raul Ruggia, "Managing Source Schema Evolution in Web Warehouses", Instituto de Computación, Facultad de Ingeniería Universidad de la República. Montevideo, Uruguay, 2001.
[33] Saurav S. Bhowmick, Wee Keong Mg, Sanjay Madria, "Web Schemas in WHOWEDA", Data Warehousing and OLAP. McLean, Virginia, United States. Year 2000, pp. 17 24, ISBN:1-58113-323-5.
[34] Cooley, R. et al, "Web Mining: Information and Pattern Discovery on the World Wide Web", *In Proceeding of IEEE International Conference Tools* with AI. Newport Beach, California, USA, pp. 558-567, (1997).
[35] Etzioni, O., "The World Wide Web: Quagmire or GoldMine", Communication of the ACM, Vol. 39, No. 11, pp. 65-68, (1996).
[36] Chakrabarti, S. et al, Focused Crawling, "A New Approach to Topic-Specific Web Resource Discovery", *In Proceeding on the 8th International Word Wide Web Conference*,. Toronto, Canada, pp. 1623-1640, (1999).
[37] Brin, S. and Page, L., "The Anatomy of a Large Scale Hypertextual Web Search Engine", *In Proceeding of the 7th International World Wide Web Conference*, Brisbane, Australia, pp. 107-117, (1998).
[38] R. Kimball and R. Merz, "The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse", John Wiley & Sons, January 2000.
[39] O.Etzioni, "The world wide web: Quagmire or gold mine", *Comm.of the ACM*,39(11):6568,1996.
[40] "Philosophies and Methodologies for Knowledge Discovery, Deployment, and Development of Decision Support Systems", PMKD, 2006, Krakow, Poland, 4 – 8 September 2006.
[41] M. J. Druzdzel et all, "Decision Support System", school of information sciences and intelligent system program, University of Pittsburgh .

**Marcello Castellano** was born in 1961. He received "Laurea cum Laude" in Computer Science in 1985 from University of Bari (Italy). Currently he is Assistante Professor at the Department of Electrical and Electronic Engineering of the Polytechnic of Bari, Italy. Previously, he has been staff member researcher at National Institute of Nuclear Physics, and computer specialist at Italian National Council of Researches. He received a scientific associate contract from Center European of Nuclear Researcher and Visiting Researcher at New Mexico State University and Gran Sasso International Laboratory (Italy). He serves as reviewer in several scientific international journals and conferences. Dr Castellano's main research interests are in machine learning, data analysis and mining. In these fields, he authored highquality scientific papers in international journals.