Detection of Access to Terror-Related Web Sites Using an Advanced Terror Detection System (ATDS)

Yuval Elovici
Deutsche Telekom Laboratories at Ben-Gurion University, Department of Information Systems Engineering, Ben-Gurion Univ. of the Negev, Beer-Sheva 84105, Israel. E-mail: elovici@inter.netil

Bracha Shapira
Department of Information Systems Engineering, Ben-Gurion Univ. of the Negev, and Deutsche Telekom Laboratories at Ben-Gurion University, Beer-Sheva 84105, Israel. E-mail: bshapira@bgu.ac.il

Mark Last, Omer Zaafrany, and Menahem Friedman
Department of Information Systems Engineering, Ben-Gurion Univ. of the Negev, Beer-Sheva 84105, Israel. E-mail: {mlast, zaafrany, fmenahem}@bgu.ac.il

Moti Schneider
School of Computer Science Netanya Academic College, 1 University st. Netanya, Israel. E-mail: motis@netanya.ac.il

Abraham Kandel
Department of Computer Science and Engineering, Univ. of South Florida, 4202 E. Fowler Ave., ENB 118, Tampa, FL, USA, 33620. E-mail: kandel@cse.usf.edu

Terrorist groups use the Web as their infrastructure for various purposes. One example is the forming of new local cells that may later become active and perform acts of terror. The Advanced Terrorist Detection System (ATDS), is aimed at tracking down online access to abnormal content, which may include terrorist-generated sites, by analyzing the content of information accessed by the Web users. ATDS operates in two modes: the training mode and the detection mode. In the training mode, ATDS determines the typical interests of a prespecified group of users by processing the Web pages accessed by these users over time. In the detection mode, ATDS performs real-time monitoring of the Web traffic generated by the monitored group, analyzes the content of the accessed Web pages, and issues an alarm if the accessed information is not within the typical interests of that group and similar to the terrorist interests. An experimental version of ATDS was implemented and evaluated in a local network environment. The results suggest that when optimally tuned the system can reach high detection rates of up to 100% in case of continuous access to a series of terrorist Web pages.

Received December 2, 2007; revised June 20, 2008; accepted September 14, 2009

 DOI: 10.1002/asi.21249

Introduction

Because of the ease of publishing information on the Web, terrorists increasingly exploit the Internet as a communication, intelligence, and propaganda tool where they can safely communicate with their affiliates, coordinate action plans, raise funds, and introduce new supporters into their networks (Kelley, 2002; Lemos, 2002; Birnhack et al., 2003; Popp, Armour, Senator, & Numrych, 2004; Wagner, 2005; Zhou, Reid, Qin, Chen, & Lai, 2005). The number of terrorist-generated and mainly terrorist propaganda Web sites is increasing rapidly. According to MSNBC News Services (2006), a Pentagon research team was monitoring more than 5,000 Jihadist Web sites back in May 2006. Since 2001, governments and intelligence agencies have been trying to identify terrorist activities on the Web to prevent future acts of terror (Ingram, 2006).

A major concern of many governments is that new terrorist cells will be formed on their territory by the local visitors of terrorist propaganda Web sites who may undergo a process of radicalization and then perform acts of terror (Lieberman & Collins, 2008). This concern has been fully justified by the recent terrorist attacks in Spain and Great Britain. Thus, a Spanish court has concluded that the deadly 2004 Madrid

train bombings were carried out by a local Islamist cell inspired by an Islamic essay published on the Internet (Harding, 2006). Similarly, a British Government report concludes that the July 7, 2005, bombings in London were a low-budget operation carried out by four men who had no direct connection to Al Qaeda and who obtained all the information they needed from the Al Qaeda sites on the Internet (Lyall, 2006).

Currently, most terrorist propaganda Web sites are constantly changing their URLs to prevent their traffic monitoring or removal from the Web by governments. In addition, even if governments could intercept the Internet protocol (IP) address of a surfer accessing the terrorist Web site by monitoring the site's inbound traffic, it would be very difficult to associate the intercepted IP with the surfer identity. Thus, there is a need for new methods and technologies to detect access to terror-related sites by monitoring the traffic of an Internet service provider (ISP) or an organizational computer network where the system administrators know the identity of each user using a specific IP. In this study, we present a methodology that enables monitoring users in a specified environment (e.g., university campus) whose Internet access is going through an organizational gateway.

The main assumptions of this study are as follows:

1. Detecting users that access terror-related sites can be achieved by monitoring and analyzing the content of the data downloaded from the Web by the users that are being monitored.
2. The legal aspects of such cyber surveillance, as well as the legality of posting or reading terrorist-generated information, are specific to the laws and regulations of each country and are beyond the scope of this work. Some of these sensitive issues are extensively discussed in Birnhack et al. (2003).
3. The monitoring is performed at the local area network or at the ISP level to be able to associate the access to terror-related sites with surfer identities or computers from which the access was made (in case a public computer is being used). It is important to point out that the monitoring of Web surfers does not require any user identification as long as they are not suspected to access terrorist-generated content.
4. By means of content monitoring and analysis of Web pages routinely accessed by a prespecified group of Web surfers, it is possible to infer the surfers' normal areas of interest (Elovici, Shapira, Last, Kandell, & Zaafrany, 2004). A *prespecified* group refers to the users in a specific organization that share common interests and can be identified by the system administrator (as they have to use their credentials to gain access to the organizational servers). Examples of such groups include students in the same department at a university, employees of the same company or department, etc. While most users are expected to be interested in the same topics, the group is not assumed to be completely homogenous implying that the group members may differ in their browsing patterns and preferences. In any case, at the time of training the system the group is not expected to include users who access terrorist content on a regular basis for whatever reason (terrorist support, terrorism research, etc.). In case of monitoring at the ISP level, the group would refer to the ISP subscribers.
5. The monitoring can be applied to groups of multilingual Web users, like the students in Israeli universities who routinely browse pages in Hebrew, English, Arabic, and even more languages. The system can learn the areas of users' interest in every language they normally use, and then utilize the induced language-specific profiles for identifying abnormal behavior in each language where there are a sufficiently large number of known terrorist Web sites.

The Advanced Terrorist Detection System (ATDS) presented in this article is aimed at tracking down access to terror-related sites by analyzing the content of information downloaded from the Web. The system operates in two modes: the training mode activated offline and the detection mode operating in real-time. In the training mode, ATDS derives information interests of a prespecified group of users by applying data mining (clustering) algorithms to the Web pages viewed by those users. Similarly, the system represents information interests of the illegal group (terrorists in our example) by analyzing and clustering known pages that relate to this illegal group. In the detection mode, ATDS performs real-time monitoring of the traffic emanating from the same group of users, analyzes the content of the Web pages they download from the Web, and raises a flag if a user accesses information that is not expected from a group member i.e., the content of the information accessed is *very* dissimilar to the normal content in the monitored group. Once an *abnormal* user is detected, further analysis of the information that she has downloaded determines whether the abnormal user is involved in viewing illegal content (e.g., terrorist propaganda). The abnormal user's information is compared with the illegal group's interests to determine if the user is just innocently accessing atypical information or if she is actually suspected to have interests in the illegal content (Shapira, Elovici, Last, & Kandel, 2008).

Some may argue that there are many nonterrorists individuals visiting terror-related Web sites that will be detected by the system. They may include security informatics researchers or antiterrorism Web bogglers and hobbyists that monitor such extremist group Web sites on their own accord. There are also occasional Web surfers that may end up on such a site out of curiosity, perhaps after watching a news story or reading a magazine article, or just by clicking a wrong link on the list of search engine results. We have to refer to the above type of abnormal users, while arguing that still a vast majority of *constant* visitors to such sites is expected to include activists and supporters rather than counter terrorism investigators.

Our research is focused on monitoring access to terrorist propaganda Web sites that target current and potential supporters of terrorist ideas rather than eavesdropping on covert communication channels used by active terrorist groups. The latter may include encrypted e-mails, restricted forums, and even steganography messages.

The reminder of the paper is organized as follows. In the Related Work section, we present previous work on detecting

DOI: 10.1002/asi

terrorist activities on the Web. In the ATDS Design Considerations and Goals section, we present the design considerations for the development of ATDS. In the Advanced Terrorist Detection System (ATDS) Architecture section, ATDS architecture and its detection algorithm are described in detail. In the Evaluating ATDS Performance section, we discuss the experiments conducted to evaluate the ATDS performance and tune its parameters. We concludes with summary and discussion of future research issues.

Related Work

Since the September 11th attack, many information technology (IT) research projects are trying to contribute to the vast homeland security efforts. One of the six critical mission areas defined by the National Strategy for Homeland Security (Chen & Wang, 2005; Chen, 2006; Chen et al., 2004) deals with Intelligence and gathering terror-related information: "IT researchers can help build new intelligence gathering and analysis capabilities for an intelligence and warning system that can detect future terrorist activities" (Office of Homeland Security, 2002). The research described in this article fits in this mission, as it offers a method for online identification of potential terrorists accessing terrorist-generated content on the Web. In this section, we survey previous studies whose goal is to identify terror-related activities on the Web, and then we explain the uniqueness of our approach.

The Internet propaganda campaign actively managed by Al-Qaeda and other terrorist organizations is described in detail by a recent U.S. Senate report (Lieberman & Collins, 2008). We differentiate between two categories of efforts for gathering and analyzing terror-related activities on the Web. The first aims at gathering and analysis of terror-related content being published and transferred on the Web (sites, e-mails, chats, forms and other Web infrastructure) to learn about terrorist networks, characterize their behavior and dynamic evolution, and hopefully prevent a planned attack (Chen et al., 2004; Gerstenfeld, Grant, & Chiang, 2003). The second category includes techniques to detect terrorists (people, not content) using the Internet, preferably while they are online to stop them and prevent their future activities (Abbasi & Chen, 2005). The study presented in this article relates to the latter, i.e., it aims at detecting suspected terrorists while they browse the Web.

One known major effort of the first category is the Dark Web research project (Reid et al., 2004), conducted at the Artificial Intelligence Lab of the MIS department of the University of Arizona, which aims to develop and evaluate scalable techniques for collecting and analyzing terrorism information. An ongoing effort yielded 500,000 terror-related Web pages created by 94 U.S. domestic groups, 300,000 Web pages created by 41 Arabic-speaking groups, and 100,000 Web pages created by Spanish-speaking groups. This collection is being analyzed using several approaches (e.g., statistical analysis, cluster analysis, visualization). Social network analysis (SNA), recently proven as an efficient method to identify relationships between human groups on the Internet, is a major approach used for analysis. SNA enables identification of central players, gatekeepers, and outliers in the terrorist groups (Zhou et al., 2005; Chen et al., 2004).

Wu, Gordon, DeMaagd, and Fan (2006) presented a new statistical approach, called principal clusters analysis, used for analyzing millions of user navigations on the Web. Their new technique identifies prominent navigation clusters on different topics. The method can potentially be used to identify terror-related navigations to collect terror-related data for analysis. Research groups, such as the Intelligence and Terrorism Information Center (http://www.terrorism-info.org.il), routinely publish current URLs of terror-related Web sites. Furthermore, the method can determine information items that are useful starting points to explore a topic, as well as key documents to explore the topic in greater detail. They also claim that the method is able to detect trends by observing navigation prominence over time. Earlier studies (Perkowitz & Etzioni, 2000; Shahabi, Zarkesh, Adibi, & Shah, 1997) used Web navigation mining techniques to analyze Web-navigation data. They perform unsupervised clustering on user navigation data on the Web to identify groups of users, resulting with clusters that were hard to interpret by humans. Our approach uses a similar idea of clustering navigation data of users in a defined environment (e.g., University campus or some organization) to learn typical user interests in the environment (represented as cluster centroids). We then use this knowledge in the detection phase to detect atypical users.

For the second category, namely, the identification of terrorists (people) on the Internet, Abbasi and Chen (2005) used authorship analysis to identify Arabic Web content to automatically identify content submitted by already known terrorists, by comparing writing style features of content (i.e., lexical, syntactical, structural, and content-specific features) using statistical and machine learning approaches. While the authors detected terrorists by looking at similarities to known terrorists' features, we look at identifying users who are dissimilar to typical users in their environment, and only then do we measure the similarity of the abnormal content viewed by these users to terrorist content. In addition, Abbasi and Chen aim at classifying an online message, as authored by a terrorist, based on various writing style features (structure, lexical, etc.) where content is not dominant (15 content-related features of 418 features for the classifier). Our approach is mainly content-based, because we look at the user's areas of interest to try to identify if they are *normal* in their environment.

Provos and Honeyman (2002) tried to confirm the rumor about terrorists using steganographic messages and developed a method to reveal such messages. However, the rumor is not yet confirmed, as an analysis of two million images downloaded from eBay auctions and one million images obtained from a USENET archive was not able to find a single hidden message. It is hard to tell whether the rumor is incorrect or the proposed method of detecting steganography is not effective.

DOI: 10.1002/asi

Baumes et al. (2006) suggest a way to identify a hidden group in a communication network where the group members are planning an activity over a communication medium without announcing their intentions. They developed algorithms for separating nonrandom planning-related communications from random background communications, while their new algorithms do not assume the existence of a planning time-cycle in the stream of communications of a hidden group.

De Vel, Liu, Caelli, and Caetano (2006) combined a Bayesian network model for encoding forensic evidence during a given time interval with a hidden Markov model (EBN-HMM) for tracking and predicting the degree of criminal activity as it evolves over time. Their evaluation results suggest that their approach can identify the expert classification of forensic data. One of the important ATDS goals is also to monitor the evolution of terrorist interests over time.

The motivation of our proposed detection system is different from the motivation of the studies described above. Whereas other systems try to detect already known terrorists, our system tries to identify new potential terrorists, who are still far below the radar of the law enforcement authorities, using the online behavior of those people. Our system uses intrusion detection inspired methodology to detect potential terrorists by defining abnormal users whose abnormal activities may indicate access to terrorist content. While in anomaly-based intrusion detection systems, the activities to be identified as normal or abnormal include behavior in the network environment (Leckie et al., 2004), our system looks at the content accessed by the user to distinguish between normal and abnormal (potentially suspected terrorist) users. Such explicit content analysis was used for detection of insider threats where the content of their communication was analyzed and compared with the normal content (Symonenko et al., 2004). We are not aware of any previous work by other researchers that involved content-based analysis of the Web traffic to detect abnormal content accessed by the Web users.

ATDS Design Considerations and Goals

In this section, we describe the design considerations and goals underlying the development of ATDS, followed by details of its architecture and elaboration of its advanced detection algorithms. The design goals for ATDS development are as follows:

1. Detecting access to terror-related site based on the retrieved content. ATDS should be able to detect access to terror-related sites by monitoring the content that surfers download from the Web.
2. Type of monitored content. The current version of ATDS focuses on network traffic content containing textual HTML pages. Due to this constraint, ATDS has to discard nontextual Web traffic content such as images.
3. Online detection. ATDS should be able to detect online access to terrorist-related content. Such online detection should enable law enforcement agencies to identify surfers accessing terrorist Web sites from public computers in a university campus or an Internet café. To achieve this goal ATDS should comply with the required performance (mainly speed).
4. Identify the suspected surfer or/and his/her computer. The system alarm should include the suspected computer IP address, its physical location, and the identity of the current computer user.
5. Detection should be based on passive eavesdropping on the network. ATDS should monitor the network traffic without being noticed by the users. A network snifter can achieve passive eavesdropping if ATDS is deployed within an organization or by installing ATDS on the ISP or LAN infrastructure if ATDS is to be deployed in a large-scale ISP-based environment.
6. Operation based on anomaly detection. ATDS learns the typical information interests of a prespecified group of users being monitored (group profile) based on the content of pages that they download from normal (nonterror-related) Web sites. The detection of abnormal users will be based on identification of users accessing atypical content, such as terror-related content.
7. ATDS performance will be controlled by the following system parameters:

    a. The number of HTML pages to be included in the detection process per IP. ATDS can monitor and collect several HTML pages for each monitored IP. This parameter is called the queue size. In our experiments we examined the effect of this parameter in the range of 2–32 pages for each IP.
    b. The minimal required level of similarity between a user accessed HTML page and the known information interests of the prespecified group which the user belongs. ATDS measures the similarity between each page that a user access and the typical interests of the group to which she belongs to identify abnormal behavior within the group. Our assumption is that a user that belongs to the normal group (e.g., a student in the monitored department at a certain university) should have similar interests to the typical interests of the group. Otherwise, the user will be identified as an atypical user and will be further examined for her/his relation to the suspected group (i.e., terrorists).
    c. The minimal number of pages collected per IP required to be similar to the typical group profile to be considered as a typical user.

Advanced Terrorist Detection System (ATDS) Architecture

In this study, we suggest a new *content-based anomaly detection system* titled ATDS that analyzes the content of Web pages accessed by a prespecified group of users as an input for detecting abnormal activity. The underlying intuitive assumption of ATDS is that the content that users download from the Web reflects their interests. This assumption is the basis of many personalization models, algorithms, and systems (Mobasher, Cooley, & Srivastava, 2000) that generate user *profiles* from the content of pages accessed by users.

DOI: 10.1002/asi

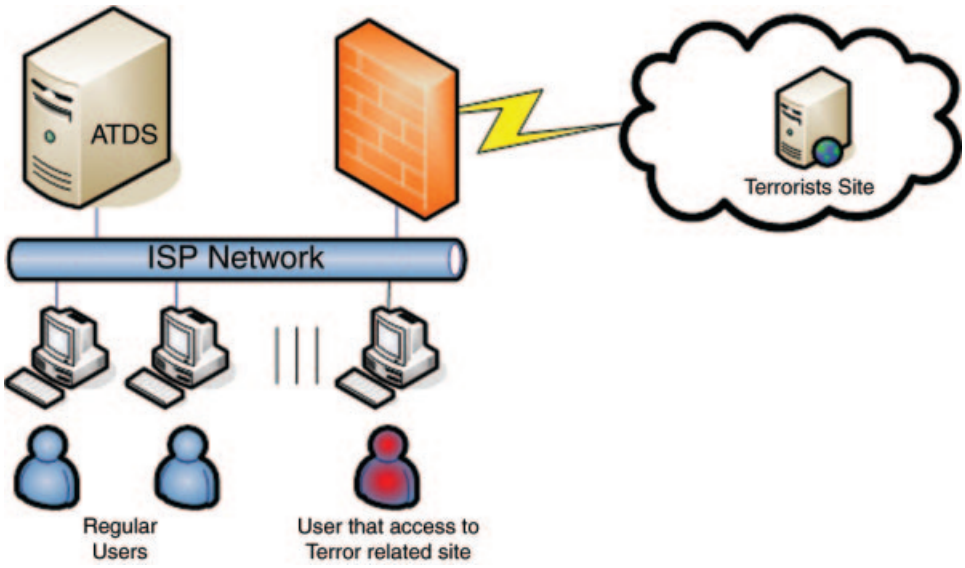

FIG. 1. Possible ATDS detection environment.

An accepted assumption in the user-modeling domain (Das, Datar, Garg, & Rajaram, 2007) is that a user's access to a site (referred also as click-through information) is a statistically sufficient indication of her interest in the site's content. A stronger assumption is that the users in the monitored group have similar interests (i.e., we are monitoring a homogenous group), and, thus, we can identify them by the content they download from the Web. Consequently, their *typical common interests*, or simply the group profile, can represent the monitored users. Individual user profiles can then be compared with the group profile to identify whether she has similar interests to those of the group. This assumption is the basis for numerous collaboration filtering systems (Hanani, Shapira, & Shoval, 2001), where users are offered documents or knowledge judged as relevant by members of the stereotypes to which the users relate.

In this study, we refer only to the textual content of Web pages, excluding images, music, video clips, and other complex data types as well as encrypted data. We assume that the content viewed by the users represents their information needs and their areas of interest and, therefore, can be used as input data for learning the group's profile for detecting an abnormal user. We define as abnormal a user that accesses information that is not expected to be viewed by a member of the related group. We then examine if the abnormal user is similar to the terrorists group. One possible detection environment of the proposed system is described in Figure 1. The system monitors all the network traffic and compares between the textual content of an HTML page downloaded from the Web by one of the users with the group profile that was computed offline for the monitored group. If the textual content of a downloaded page is *consistently and significantly* dissimilar to the typical common interests while being similar to the terrorists' interests, then an alert is issued. The alert indicates that the user accessed abnormal content that could be related to a terrorist Web site. The terrorist user's areas of interest are derived and represented by the system in the learning phase from a set of known terror-related sites. The degree of *consistency and significance* of dissimilarity that triggers an alarm is controlled by several system parameters. The alert would ideally result in the detection of the IP address and the physical location of the computer that is accessing this information, assisting in a quick and reliable identification of the abnormal user. This requires, of course, cooperation of the ISP or the network administration of the organization to which the group belong. Some may argue that ATDS cannot detect users hidden behind a network address translator (NAT). Zakin et al. (2007) presented an approach for clustering the sessions emanating from the NAT to identify the intruders. Each cluster should ideally include only the sessions emanating from a specific computer and its content can be analyzed by ATDS.

ATDS has two modes of operation, described below.

The Learning Mode is where the learning of the group's typical common interests is performed. In this mode, the Web traffic of a group is monitored, and all intercepted HTML Web pages are collected and stored in a database during a specified period of time. The training group should be carefully selected to ensure that it does not contain abnormal users that may corrupt the training set by frequently accessing terrorist Web sites. The Web pages are converted and represented as a set of vectors where each vector represents a single accessed Web page. Each entry in the vector represents a weighted term (see more information on the vector space model in Salton and Buckley, 1988). The collected data are used to derive and represent the typical common interests of the group's users by applying techniques of unsupervised clustering. The output of the learning mode is a set of centroid vectors that represent the typical common interests of the group of users. It is important to note that in case of heterogeneous groups, some centroids may represent interests, which are shared only by a subset of users. User clustering methods can explicitly identify such

DOI: 10.1002/asi

user subgroups. However, our system is based on *Web page clustering*, which does not require identification of normal users. In the learning mode, we also apply the same clustering procedure to a training set of terror-related pages (collected from known terrorist Web sites) to learn and represent terror-related interests. The learning mode is performed as a batch process, whereas the detection mode involves online processing. Detailed description of the learning mode can be found below.

The detection mode is aimed at detecting abnormal users who access content that is not within the typical common interests of the monitored group and examines if they might belong to a terrorist group. In the detection mode, all network traffic is intercepted, HTML Web pages retrieved from the Web by the monitored users are processed, and their textual content is represented as a vector of weighted terms. The detection algorithm that examines whether the page is within the typical common interests of the group processes the vectors. The detection algorithm behavior is controlled by several parameters.

After a user was identified as atypical to her group, we compare her interests with the typical terrorists interests (also represented as clusters). Only if the abnormal user's interests are similar to those of terrorist interests, then she is declared a suspected member of a terrorist group. This enhanced analysis is aimed at reducing the false-positive rate, i.e., suspecting users as terrorists only because they are atypical to their normal group.

In the following subsections, we describe in detail the learning and the detection modes of ATDS.

Learning the Typical Common Interests of a Group of Users

During the learning mode, ATDS computes the typical common interests of a group of users based on HTML pages that they download from the Web during a specific period of time. Figure 2 describes the learning mode. The various modules of ATDS involved in the computation of the typical common interests are described below:

Sniffer. The sniffer is responsible for intercepting the monitored users' traffic by collecting IP packets transmitted over the communication lines. The sniffer discards all packets whose destination or source IP is not under surveillance. Nontransmission control protocol (TCP) packets are also discarded. A new TCP flow is opened upon intercepting of a full TCP handshake protocol and copies of all received packets, belonging to an existing TCP flow, are routed to the HTTP filter module. To represent a TCP flow, we adopted the 4-tuple structure: <source-IP-address, source port, destination-IP-address, destination port> as suggested by MOT (Mao, Chen, Wang, Zheng, & Deng, 2001).

HTTP filter. This module is in charge of identifying new HTTP sessions within the intercepted TCP flow, identifying

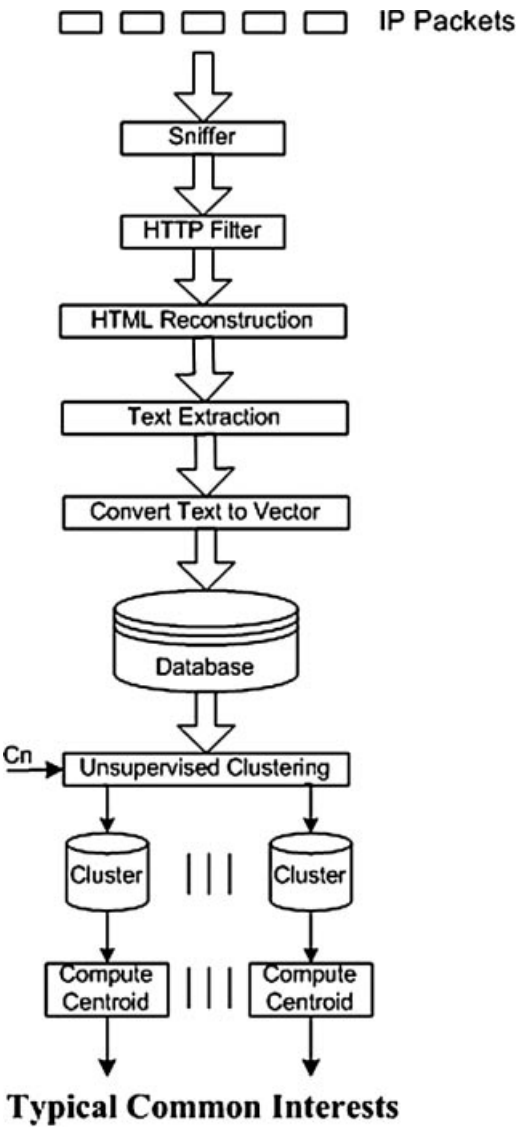

FIG. 2. The learning mode.

the content type (text, video, audio etc.) of each HTTP session and filtering out all nontextual HTTP sessions. A type identification test is applied to the HTTP header response. If the identified content type is not text or HTML (but, for example, audio, video, etc), then the HTTP request header and the HTTP response packet are discarded from the memory. Each packet belonging to that HTTP session is discarded as well. If content-type header is missing, then an attempt to identify the media type is made. If the media type remains unknown, then the session is not considered a text or HTML HTTP session (Fielding et al., 1999).

A new HTTP session is opened up within the TCP structure for each HTTP response of type text or HTML. The next incoming packets are routed to their identified HTTP session according to the packets acknowledgment number. A TCP flow is completed according to patterns described by Yun Mao (Feldman 2000; Mao et al., 2001). Packets of HTTP sessions that contain the textual data payload are sent to the HTML reconstruction module.

DOI: 10.1002/asi

HTML reconstruction. The HTML reconstruction module rebuilds packets belonging to HTTP sessions that carry an HTML file to their original file before they were fragmentized to packets. The reconstruction is performed according to the packets' sequence number. A procedure for checking lost packets and end of document is applied to each HTTP session. The packets of a complete document are reconstructed to their original form (i.e. a document). An identification procedure is performed to find out if the document is in the form of HTML. All documents that don't meet this condition are deleted. Once a complete HTML file is reconstructed (all its packets are found), it is sent to the text extraction module.

Text extraction. This module is responsible for extracting the textual content from the HTML files by removing all the HTML tags that are used for formatting the HTML documents. The extracted text is sent to the convert text to vector module.

Convert text to vector. In ATDS, we represent the textual documents as a vector of weighted terms (according to the vectors-space model; Salton & Buckley, 1988). To transform the HTML pages to a vector representation, we used an off-the-shelf product called Extractor [http://www.extractor.com/]. The Extractor receives as input a document (text, html, e-mail) and uses a patented keyphrase extraction genetic algorithm (GenEx) to analyze the recurrence of words and phrases, their proximity to each another, and the uniqueness of the words in a particular document. The Extractor's output is a set of up to 30 keyphrases for each document. Extractor assigns a weight to each keyphrase in a document representing the importance of a keyphrase to the given document. A detailed description of the Extractor keyphrase extraction algorithm is provided in Turney (2000). The relatively low number of terms (30 or less) used by the Extractor to represent a document prevents the need to apply a dimensionality reduction process or handle long vectors, which could increase the processing time in the next stages of the learning mode (and thus decrease the performance). The vectors generated in the learning mode are saved in a database. The vectors collection takes place during a predefined period of time (e.g., a week), or until a sufficient number of vectors are stored.

Unsupervised clustering. The clustering module accesses the collected vectors stored in the database and performs unsupervised clustering (e.g., using the popular k-means algorithm) resulting in n clusters. The number of clusters Cn is one of the system parameters specified by the user.

Compute clusters centroids. For each cluster, a centroid vector is computed to represent the cluster by averaging the vectors included in the cluster using the following equation:

$$Cv_i = \left(\frac{\sum_{j=1}^{k_i} Av_j}{k_i} \right)$$

where

Cv_i is the ith centroid vector,
Av_j is the jth vector in the cluster
k_i – Number of vectors in the ith cluster

Each centroid vector Cv_i is expected to represent one of the topics commonly accessed by the users of the group. We refer to the centroid vectors as typical common interests of the users or simply the group profile.

The learning phase should be transparent to the users, though the system administrators may notify the users that their Web accesses are recorded by a machine and ask their permission to do that if needed.

During the learning mode, we can also learn and represent terror-related interests (terror as an example of an illegal group but it could be any other group). For this we can download representative pages from known terrorist Web sites and represent them as vectors and clusters, as described above (we do not need to apply the sniffing modules as we do not collect pages from the network). Specifically, we apply the text extractor, Convert text to vectors, unsupervised clustering and compute clusters centroids modules. The result of these modules is a set of centroids of the clusters representing *typical terrorist interests*.

Detection Mode

Once the learning mode is complete and the group profiles are computed (in batch), the system can move to the detection mode. In the detection mode, all network traffic is monitored and HTML Web pages are intercepted to detect users accessing atypical content. Each new incoming intercepted Web page is converted into a vector of weighted terms and similarity is computed between the current vector and the known group profiles. The decision of whether the user who accessed a recent incoming page or a series of pages is an atypical (abnormal) user is made by the detection algorithm, which considers the history of the user's accesses to the Web. If, in the learning mode, a profile of typical terrorist content has been built, then the system will also check the similarity of abnormal users to terrorist content. The detection mode presented in Figure 3 involves eight modules, of which the first five—sniffer, HTTP filter, HTML reconstruction, text extraction, and convert text to vector—are identical to those in the learning mode, i.e., the pages that users access are captured, filtered, and transferred to a vector representation. We, therefore, describe here only the other three modules: similarity checker; vectors classifier; and anomaly detector including its detection algorithm.

Similarity checker. The similarity checker computes the similarity between a new incoming access vector (denoted by Av) and each of the centroid vectors of the typical common interests or the typical terrorist interests. The cosine distance measure (Frakes & Baeza-Yates, 1992) is used for computing similarity. The output of this module is the maximal similarity between the incoming vector and one of the

DOI: 10.1002/asi

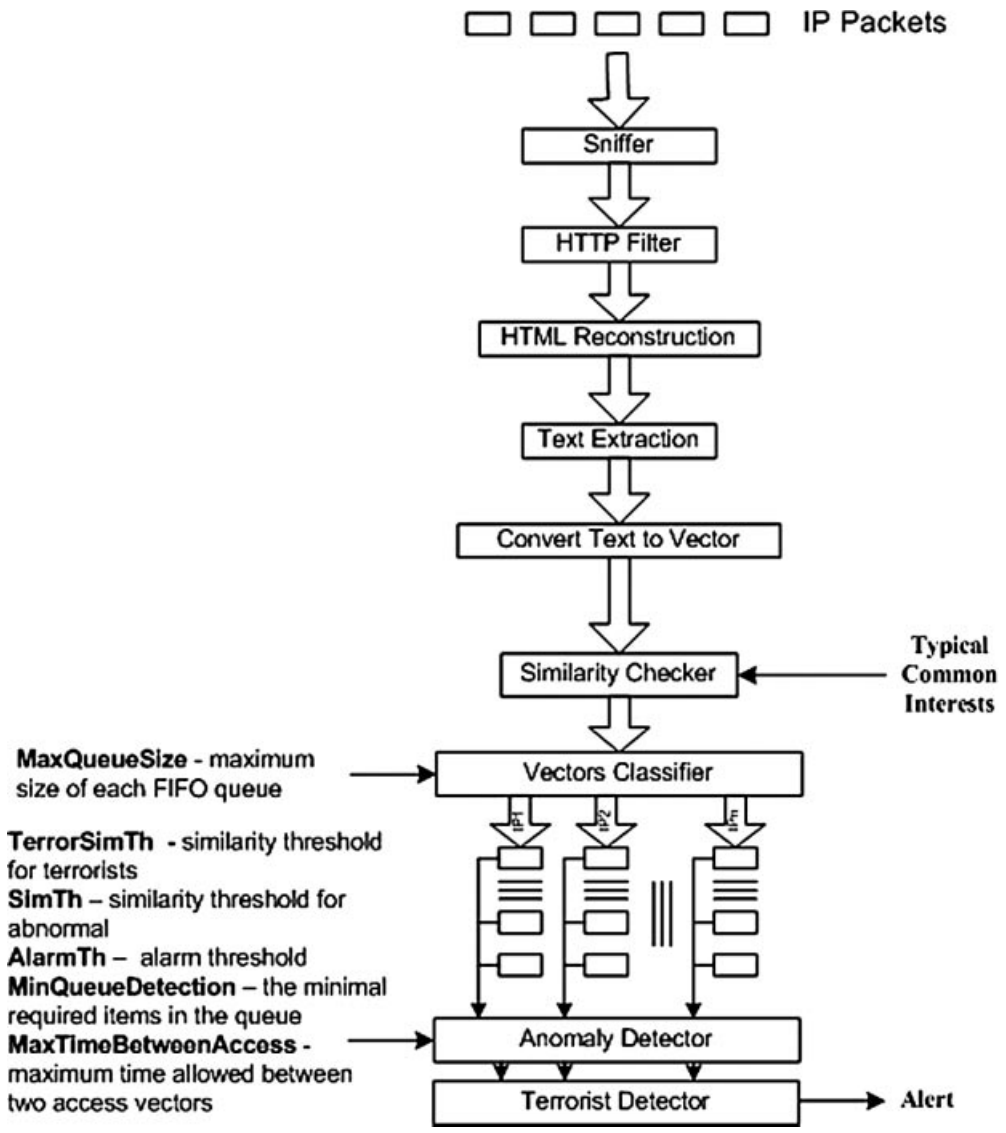

FIG. 3. Detection mode of ATDS.

centroid vectors (denoted by $MaxSim(Av)$), as described in the following equation:

$$MaxSim(Av) = Max\left(\frac{\sum_{i=1}^{m}(tCv_{i1}\cdot tAv_i)}{\sqrt{\sum_{i=1}^{m}tCv_{i1}^2\cdot\sum_{i=1}^{m}tAv_i^2}},\ldots,\frac{\sum_{i=1}^{m}(tCv_{in}\cdot tAv_i)}{\sqrt{\sum_{i=1}^{m}tCv_{in}^2\cdot\sum_{i=1}^{m}tAv_i^2}}\right)$$

where

tCv_{ij} is the ith term of the jth centroid vector.
Av – the access vector.
tAv_i – the ith term of the access vector Av.
m – The number of unique terms in each vector.
n – Number of clusters.

Vectors classifier. Vector classifier classifies the incoming vectors by their source IP. ATDS maintains a First In, First Out (FIFO) queue for each monitored IP address in the network. For every new intercepted access vector, the vectors classifiers push in the appropriate FIFO queue (based on Av IP) the $MaxSim(Av)$, along with the timestamp of the access vector interception. The system parameter MaxQueueSize controls the size of the FIFO queue.

Anomaly detector. Upon updating the content of one of the queues, the abnormal detector module compares each $MaxSim(Av_i)$ in the queue with the system parameter denoted by *SimTh* (a configured similarity threshold). If the $MaxSim(Av_i)$ is lower than *SimTh* than the *Avi* is marked as abnormal. The system parameter *AbnormalTh* controls the percentage of the queue vectors that should be marked as abnormal to identify the user as an abnormal user.

Terrorist detector. If the system has built a profile of typical terrorist content in the learning mode, then the *similarity_checker* is applied between the vector of each user

DOI: 10.1002/asi

who was identified as abnormal by the anomaly detector and the centroids of the terrorist content clusters. If $MaxSim(Av_i)$ is higher than the *TerrorSimTH,* then the user is identified as a suspicious user and an alarm is issued. The detection algorithm is summarized in Algorithm 1, as follows.

Input:
typicalCommonInterests - a set of n vectors representing the centroids of n clusters.
IPs – a list of the IP's under surveillance.
MaxQueueSize – maximum size of each FIFO queue.
MinQueueDetection – the minimal required number of access vectors in a queue.
SimTh – similarity threshold – determines the minimal similarity of user's vectors to the typical group interest to be identified as a typical user.
AbnormalTh – the abnormal threshold denotes the percentage of queue vector that should be abnormal to identify the user as such (atypical to the group).
MaxTimeBetweenAccess – maximum time allowed between two access vectors.
TerrorSimTH – the terror similarity threshold defines the minimal similarity of a user vector to the terrorist group interest that makes her suspected to belong to the terrorists group.
Av_i – a new access vector.
IPi – IP of the computer that downloaded the Av_i vector from the Web.
Ti – timestamp of the Av_i interception.

Output:
Alert

Step 1: Create new queue if needed.
If IPi is not in **IPs,** then Queues[IPi] ← Create a queue with size
MaxQueueSize.

Step 2: Find the smallest distance between Avi and one of the normal centroids.
MaxSim ← 0
For each vector in typicalCommonInterests
TempSim ← CalcSim(vector , Avi)
If TempSim > MaxSim Then MaxSim ← TempSim
UsersQueues[IPi].Push (MaxSim, Ti)

Step 3:
Counter ← 0
Flag ← 0
For each pair in UsersQueues[IPi]
If UsersQueues[IPi].Back.T - UsersQueues[IPi].
Front.T >
MaxTimeBetweenAccess, then UsersQueues[IPi].
Pop(UsersQueues[Ipi].Back);
if UsersQueues[IPi].size < MinQueueDetection,
then exit.
For each MaxSim in UsersQueues[IPi] if MaxSim < SimTh, then Counter++
If (counter/UsersQueues[IPi].size >= AlbnormalTh), then flag ←1; identify user as atypical.

Step 4 (optional): If a user is abnormal **e**xamine if a user belongs to the terrorist group.
If flag = 1
MaxSim ← 0
For each vector in terrorCommonInterests
TempSim ← CalcSim(vector , Avi)
If TempSim > MaxSim, then MaxSim ← TempSim
If MaxSim> *TerrorSimTH,* then issue an Alarm

Evaluating ATDS Performance

An experimental version of ATDS was implemented using Microsoft Visual C++ and designed in a modular architecture. The computational resources and the complexity of the methodology behind ATDS required a careful design to enable real-time online sniffing. The system was deployed on an Intel Pentium 4 2.4 GHz server with 512 MB RAM, which was installed at the Computation Center of Ben-Gurion University (BGU), Israel. The normal group was defined to include only the students of one department at BGU (information systems engineering [ISE]). Thus, the server was configured to monitor 38 public stations in the teaching labs of the ISE department. The stations comprised Pentium 4, 2.4 GHz with 512 MB of RAM, 1Gbps fast Ethernet Intel adapter and Windows XP professional operating system. The access to those stations is allowed only to the ISE students who may be considered as a relatively homogenous group, as they have similar backgrounds and study according to the same teaching curriculum. To measure the false-positive rate of the system on users who may have different interests from the normal group, but have nothing to do with a terrorist group, we have also monitored the Web traffic of a computing lab at the Faculty of Social Sciences and Humanities in the same university. That lab included 13 public stations. The simulation runs that evaluate ADTS are described in the following sections.

Data Preparation for the Simulations

We prepared two training sets for the learning mode, which contained normal and terrorist Web pages, and three test sets for the detection mode, which included accesses to typical (*normal*), atypical (*abnormal nonterrorist*), and *terrorist* content to represent typical, atypical, and terrorist users, respectively. As explained below, the terrorist content originated from three distinct terrorist organizations.

For the learning mode, we collected 170,000 student accesses to the Web in the ISE teaching labs during 1 month of regular classes. Because most students are using these labs at least 4 days every week and their information needs tend to be relatively stable (academics, social networking, entertainment, etc.), we assume that 1 month of data should be

DOI: 10.1002/asi

sufficient for covering their normal interests. All students were aware of their access activity monitoring and agreed to have their accesses collected anonymously. Non-English pages (mainly, in Hebrew) were excluded from the training set, because ATDS is not currently aimed at detecting access to terrorist pages in Hebrew.[1] After the exclusion, the training set included 13,276 pages. The HTTP filter, HTML reconstruction, text extraction and convert text to vector modules were applied to the 13,276 pages resulting in a 13,276 * 38,776 matrix (i.e., 38,776 distinct terms representing the 13,276 pages). These terms form the features of the vectors in the clustering process.

We have randomly selected 952 vectors to be removed from the normal matrix and used as a part of the validation set representing accesses to normal content. Thus, our final training set contained 12,324 normal vectors only. The abnormal nonterrorist part of the validation set included 217 vectors of Web pages viewed by the users of the humanities computing lab. In addition, we downloaded a set of 582 terror-related pages from Jihadi Web sites in English (mostly associated with Azzam Publications and Al Qaeda) for the simulation of accesses to the terrorist content. The HTTP filter, HTML reconstruction, text extraction and convert text to vector modules were applied to these pages, resulting in a set of vectors representing terrorist-generated sites. The resulting 582 vectors were used for generating the terrorist content profile. To simulate access to terrorist content in the validation set, we have downloaded 913 documents from a Hezbollah Web site called Moqawama (http://www.moqawama.org/english/) and 91 documents from a Hamas Web site called Palestine Info (www.palestine-info.co.uk/am/publish/). The terrorist documents in the training and the validation sets were collected from different terrorist organizations (Al-Qaeda vs. Hezbollah and Hamas), because in a real-world situation, we may expect the terrorist users to access new–previously unknown to the system–propaganda Web sites. Obviously, training the system on all terrorist Web sites in the world or even all Jihadi Web sites would be a completely infeasible task, especially because such sites are renewing constantly.

Evaluation Objectives and Measures

We evaluated ATDS performance by plotting receiver operating characteristic (ROC) curves using the following measures (based on Sequeira & Zaki, 2002):

True positive (TP) (also known as detection rate or completeness or hit rate) is the percentage of terrorist detection in case of terrorist (positive) activity.

False positive (FP) is the percentage of false alarms when nonterrorist (negative) activity is taking place.

[1] According to Intelligence and Terrorism Information Center (2007), there is currently only one known Jihadi Web site in Hebrew – Hezbollah Web site called Moqavemat. Consequently, we do not consider detection of access to terrorist Web sites in Hebrew as an important feature of ATDS.

ROC graphs graphically represent the tradeoff between the TP and FP rates for every possible cutoff. Equivalently, a ROC curve represents the tradeoff between sensitivity and specificity. In a ROC graph, the X-axis represents the FP rate, whereas the Y-axis represents the TP alarm rate. A point on the graph represents the FP/TP for a specific similarity threshold *SimTh*.

The experiments were aimed at examining the effect of the following system parameters on the detection performance:

1. *SimTh* – similarity threshold for comparing the user vectors to the typical group interest.
2. *Cn* – number of clusters representing the typical common interests.
3. *MaxQueueSize* – the size of the queue of Web pages accessed by a single IP.
4. *AbnormalTh* – the abnormal threshold denoting the percentage of abnormal vectors in a queue vector.
5. *TerrorSimTH* – the terror similarity threshold for comparing the user vectors to the terrorist group interest.

In addition, we have also studied the effect of the navigation behavior of an atypical user on the detection performance.

The Experiments

Effect of the number of clusters Cn on the ATDS performance. To examine the feasibility of ATDS and examine the effect of the number of clusters *Cn* on TP and FP rates, we ran ATDS in the learning mode three times, each with a different number of clusters, 50, 100, and 200, while setting the MaxQueueSize to 1 (for all three runs). We then simulated accesses to single Web pages by normal, abnormal nonterrorist, and terrorist users. To simulate normal users, we accessed pages typical for the monitored group (ISE students), whereas the simulation of abnormal nonterrorist users included access to pages viewed in the humanities lab. Terrorist accesses were simulated using terrorist-generated Web pages downloaded from Hezbollah and Hamas propaganda Web sites mentioned above. The detection module was applied during these simulations to examine whether the system is able to detect terrorist users accessing a variety of terrorist Web sites.

The results have proven the feasibility of ATDS implementation. Figures 4a and 4b describe some of the simulation results showing the TP and FP rates for a terrorist user accessing a single terrorist page and a nonterrorist user accessing a single nonterrorist page as a function of the similarity thresholds *SimTh* and *TerrorSimTH*. Each ROC curve refers to a different number of clusters generated during the learning mode.

Figure 4a assumes that no terrorist-generated pages are available for learning, and the detection of suspicious users is based on the anomaly detector module only whereas Figure 4b refers to the terrorist detector applied in tandem with the anomaly detector. The area under the ROC curves is shown in Table 1, which includes an additional case of the terrorist detector applied directly to all vectors (both normal and

DOI: 10.1002/asi

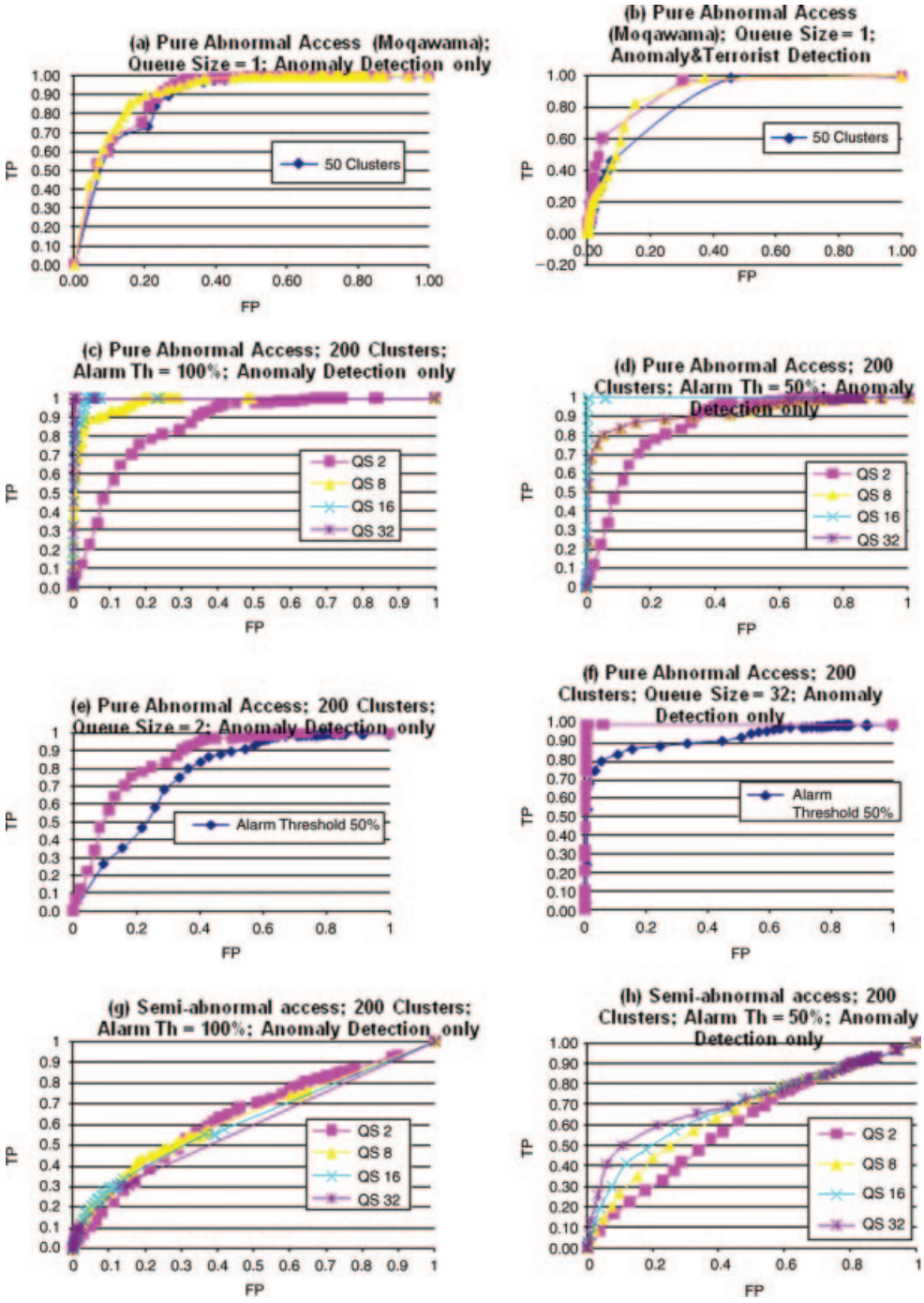

FIG. 4. Selected ROC curves.

abnormal) without using the anomaly detector module. As can be seen from Figures 4a–4b and from Table 1, the results for 50, 100, and 200 clusters exhibit similar performance with the area under the curve (AUC) going up to 89%–91% when only the anomaly detection module is used.

As shown in Table 2, the differences between the AUCs for a number of clusters were found statistically insignificant at the 99% confidence level using a nonparametric test for comparing two ROC curves described in Krzanowski and Hand (2009). On the other hand, according to Table 3, which are based on the same test, there is a statistically significant increase in the area under ROC curve when the terrorist detector module is replaced by the anomaly detector module (disregarding the number of clusters), because the validation set included pages from different terrorist Web sites than the training set.

Using the terrorist detector in addition to the anomaly detector has improved the performance by 11.5% (in case

DOI: 10.1002/asi

TABLE 1. Area under the curve for queue size = 1 with different number of clusters and different detection modes.

Number of clusters	50	100	200
Moqawama			
Anomaly detection only	0.879	0.896	0.911
Terrorist detection only	0.657	0.689	0.745
Anomaly detection + terrorist detection	0.837	0.907	0.894
Palestine Info			
Anomaly detection only	0.695	0.837	0.889
Terrorist detection only	0.817	0.619	0.683
Anomaly detection + terrorist detection	0.810	0.899	0.879

TABLE 2. Comparison of the number of clusters (p-values).

Number of clusters	50 vs. 100	200 vs. 100
Moqawama		
Anomaly detection only	0.2245	0.2627
Terrorist detection only	0.1320	0.0268
Anomaly detection + terrorist detection	0.0498	0.3916
Palestine Info		
Anomaly detection only	0.0002	0.0716
Terrorist detection only	0.0000	0.0469
Anomaly detection + terrorist detection	0.0197	0.3568

TABLE 3. Comparison of detection modes (p-values).

Number of clusters	50	100	200
Moqawama			
Anomaly detection vs. terrorist detection	0.0000	0.0000	0.0000
Anomaly + terrorist detection vs. Terrorist detection	0.0000	0.0000	0.0000
Anomaly + terrorist detection vs. Anomaly detection	0.0783	0.3917	0.3163
Palestine Info			
Terrorist detection vs. anomaly detection	0.0002	0.0000	0.0000
Anomaly + terrorist detection vs. Terrorist detection	0.4093	0.0000	0.0000
Anomaly + terrorist detection vs. anomaly detection	0.0012	0.0835	0.4182

of Palestine Info) or less, though most performance changes were not found statistically significant. The improvement happens when the terrorists' pages have some similarity to known terrorist content and can be detected as such. However, these pages are not similar enough to be detected as terrorists if only the terrorist detection module has been applied with a higher threshold to prevent false alarms. In the case of applying the two modules, false positive is prevented because the users are first detected as atypical and only then are compared with known terrorists. The system needs to be calibrated for optimal similarity thresholds for both modules, the abnormal and the terrorists, but our results do show that it is feasible to tune the system to obtain good results.

We believe that it is impossible to draw a general conclusion about an ideal number of clusters. It depends on the training corpus and the number of vectors included in the process. However, this simulation shows that as the number of clusters may affect the system's performance, a sensitivity analysis (such as this simulation) is required to find the optimal number of clusters for a given monitored group. In addition, these simulations confirm the system's robustness because even when no terrorist-generated pages are available for training, we are able to obtain reasonable results (AUC up to 91%) with a simple anomaly detection algorithm. The obtained results also indicate that it is preferable to prepare a training corpus of terrorist-generated pages for the terrorist detector module, which may, in some cases, improve the performance of the anomaly detector module. We think that using the terrorist detector module prevents false-positive alarms that could be issued when users are atypical to the group but are not interested in terrorism. However, if terrorist data is not available, it can be useful to identify the atypical users using only the anomaly detector module and to examine these atypical users with other means. It can be far better than missing potential terrorists accessing the Internet

Effect of queue size and abnormal threshold on ATDS performance. In this simulation experiment, we examined the effect of the queue size *MaxQueueSize* and the abnormal threshold *AbnormalTh* on the detection performance of the anomaly detector in terms of TP, FP, and AUC; we, therefore, ran several experiments providing the detection algorithm with different values of these parameters. We examined the queue size with the values, 2, 8, 16, and 32, and the alarm thresholds with values of 50% and 100%. Because these simulations might be affected by the order of incoming vectors that might alter the number of abnormal pages in a user's access queue and the rate of the abnormal pages in the queue (abnormal threshold), we repeated the detection 10 times to cancel this effect. We present the results averaged for the 10 repeated simulations. As in the previous experiment, a terrorist user was simulated by accesses to abnormal (terrorist-generated) pages only (pure abnormal access). The graphs in Figures 4c and 4d show the effect of the queue size and those in Figures 4e and 4f show the effect of the alarm threshold. The AUC of each chart is presented in the left part of Table 4.

It can be seen from Figures 4b-4e and Table 4 that the detection performance improves with the increase of the queue size for both values of the abnormal threshold. Also, the graphs show that for these pure abnormal access data, the 100% abnormal threshold is better than the 50% threshold. Actually the system reached an almost ideal detection for queue size = 32 and abnormal threshold of 100%. This result of superiority of the 100% abnormal threshold over the 50% threshold cannot be generalized because it may depend on the monitoring data. However, the results suggest that with sensitivity tuning of the advanced detection algorithm's parameters, it is possible to optimize the system's performance.

DOI: 10.1002/asi

TABLE 4. Area under curve as a function of queue size (QS) and abnormal threshold (Th).

	Pure abnormal access		Semiabnormal access		
QS\Th	50%	100%	50%	100%	Average
2	0.7592	0.8586	0.6071	0.6337	0.7147
8	0.8719	0.9771	0.6598	0.6374	0.7865
16	0.9079	0.9932	0.6882	0.6190	0.8021
32	0.9209	0.9966	0.7150	0.5900	0.8056
Average	0.8649	0.9564	0.6675	0.6200	

Effect of abnormal user navigation behavior. In this experiment, we examined the sensitivity of the system detection performance to the navigational behavior of *semiabnormal* users. We, therefore, simulated users accessing both typical and atypical (terrorist-generated) content in a random order. Specifically, the simulation of the abnormal users included access to 50% normal Web pages and 50% terrorist-generated Web pages. The results are presented in Figures 4g and 4h and the right part of Table 4. We can see from Figure 4g that for the abnormal threshold of 100%, increasing the queue size does not improve the FP and TP rates and different queue sizes give similar results; whereas in Figure 4h, which represents the TP and FP rates for the alarm threshold of 50%, increasing the queue size results in better FP/TP rates. These observations are supported by the AUC results in Table 4. Our results suggest that the system should be configured with the largest queue size possible and the abnormal threshold should be set to the value that will allow detection of terrorist users with various surfing habits.

Summary and Future Research

ATDS is aimed at tracking down potential terrorists accessing terrorist-generated Web sites based on the content of information accessed by the Web users. ATDS is based on the content-based methodology for anomaly detection on the Web introduced in (Last et al., 2003a,b; Elovici et al., 2004, 2005). In this article, we presented a comprehensive evaluation of ATDS performance in an experimental environment that confirmed its feasibility and also studied the effect of calibrating ATDS parameters. We have also shown that the terrorist detector module trained on a collection of typical terrorist Web pages can improve the ATDS detection capabilities when used in tandem with the anomaly detector module.

An important contribution of ATDS lies in its unique application environment. The detection component is planned to operate in a real-time, wide-area network where it should be capable of simultaneously monitoring hundreds and thousands of users. Therefore, a crucial design requirement is high scalability of the data mining and the detection algorithms. ATDS is an example of applying data mining and information retrieval techniques in the international effort against the terror presence on the Internet

As for future research issues, we are developing a cross-lingual version of the system as many terrorist-generated Web sites use languages other than English (e.g., Arabic). Last, Markov, and Kandel (2006) presented preliminary results of detecting terrorist content in Arabic. We are also planning to analyze multimedia content on pages, such as logos, pictures, colors, video clips, and any other nontextual features that may identify terrorist content.

References

Abbasi, A., & Chen, H. (2005). Applying authorship analysis to extremist-group Web forum messages. IEEE Intelligent Systems, Special Issue on Artificial Intelligence for National and Homeland Security, 20(5), 67–75.

Baumes, J., Goldberg, M., Hayvanovych, M., Magdon-Ismail, M., Wallace, W., & Zaki, M. (2006). Finding hidden group structure in a stream of communications. In S. Mehrotra, D.D. Zeng, & H. Chen (Eds.), Proceedings of the IEEE Conference on Intelligence and Security Informatics (pp. 201–212). Los Alamitos, CA: IEEE.

Birnhack, M., & Elkin-Koren, N. (2003). Fighting terror online: The legal ramifications of September 11. Internal Report, The Law and Technology Center, Haifa University. Retrieved November 29, 2007, from http://law.haifa.ac.il/faculty/lec_papers/terror_info.pdf

Chen, H. (2006). Intelligence and security informatics: Information systems perspective. Decision Support Systems: Special Issue on Intelligence and Security Informatics, 41(3), 555–559.

Chen, H., Qin, J., Reid, E., Chung, W., Zhou, Y., Xi, W., et al. (2004). The dark Web portal: Collecting and analyzing the presence of domestic and international terrorist groups on the Web. In W.T. Scherer & B.L. Smith (Eds.), Proceedings of the 7th IEEE International Conference on Intelligent Transportation Systems, (pp. 106–111). Los Alamitos, CA: IEEE.

Chen, H., & Wang, F. (2005). Artificial intelligence for Homeland Security. IEEE Intelligent Systems, Special Issue on Artificial Intelligence for National and Homeland Security, 20(5), 12–16.

Das, A.S., Datar, M., Garg, A., & Rajaram, S. (2007). Google news personalization: Scalable online collaborative filtering. In Proceedings of the 16th International Conference on World Wide Web (pp. 271–280). New York: ACM Press.

De Vel, O., Liu, N., Caelli, T., & Caetano, T.S. (2006). An embedded Bayesian network hidden Markov model for digital forensics. In S. Mehrotra, D.D. Zeng, & H. Chen (Eds.), Proceedings of the IEEE Conference on Intelligence and Security Informatics (pp. 459–465). Los Alamitos, CA: IEEE.

Elovici, Y., Shapira, B., Last, M., Kandell, A., & Zaafrany, O. (2004). Using data mining techniques for detecting terror-related activities on the Web. Journal of Information Warfare, 3(1), 17–28.

Elovici, Y., Shapira, B., Last, M., Zafrany, O., Friedman, M., Schneider, M., et al. (2005). Content-based detection of terrorists browsing the Web using an Advanced Terror Detection System (ATDS). In P. Kantor, G. Muresan, F. Roberts, D. Zeng, F.-Y. Wang, H. Chen, & R. Merkle (Eds.), Proceedings of the IEEE International Conference on Intelligence and Security Informatics, Lecture Notes in Computer Science, 3495, 244–255. Springer.

Frakes, W., & Baeza-Yates, R., (Ed.). (1992). Information retrieval: Data structures & algorithms. New Jersey: Prentice Hall.

Gerstenfeld, P.B., Grant, D.R., & Chiang, C.-P. (2003). Hate online: A content analysis of extremist Internet sites. Analyses of Social Issues and Public Policy, 3(1), 29–44.

Hanani, U., Shapira, B., & Shoval, P. (2001). Information filtering: Overview of issues, research and systems. User Modeling and User-Adapted Interaction, 11(3), 203–259.

Harding, B. (2006). 29 Charged in Madrid Train Bombings. Retrieved November 29, 2007, from http://news.scotsman.com/topics.cfm?tid=1094&id=556932006

DOI: 10.1002/asi

Ingram, M. (2001). Internet privacy threatened following terrorist attacks on US. Retrieved November 29, 2007, from http://www.wsws.org/articles/2001/sep2001/isp-s24.shtml

Intelligence and Terrorism Information Center. (2007). The Internet as a battleground used by the terrorist organizations: How Hezbollah and Hamas exploit the Internet in the battle for hearts and minds, and how to combat them. Retrieved March 22, 2008 from http://www.terrorism-info.org.il/malam_multimedia/English/eng_n/pdf/int_e250707.pdf

Karypis, G., Cluto. (2002). A Clustering Toolkit, Release 2.0, University of Minnesota, Karypis Lab. Retrieved November 29, 2007, from http://glaros.dtc.umn.edu/gkhome/views/cluto

Kelley, J., (2002). Terror Groups behind Web encryption. Retrieved November 29, 2007, from http://www.usatoday.com/tech/news/2001-02-05-binladen.htm.

Krzanowski, W.J., & Hand, D.J. (2009). ROC curves for continuous data. Boca-Raton, FL, Chapman & Hall/CRC Monographs on Statistics & Applied Probability.

Last, M., Elovici, Y., Shapira, B., Zaafrany, O., & Kandel, A. (2003a). Using data mining for detecting terror-related activities on the Web. In Proceedings of the Second European Conference on Information Warfare and Security (ECIW '03) (pp. 271–280). Retrieved November 16, 2009, from http://books.google.com/books?id=zrg3cMbSWjwC&dq=proceedings+of+the+2nd++European+Conference+ on+Information +Warfare+and+S ecurity&printsec=frontcover&source=bl&ots=1hXQqljc3u&sig= acRTjvv0hCVzArgmhDN_gxi7qUw&hl=en&ei=kLcBS6yhO5SflAeVwKyMCw&sa= X&oi=book_result&ct=result&resnum=1&ved=0CAgQ6AEwAA#v=onepage&q=&f=false

Last, M., Elovici, Y., Shapira, B., Zaafrany, O., & Kandel, A. (2003b). Content-based methodology for anomaly detection on the Web. In E. Menasalvas, J. Segovia, & P.S. Szczepaniak (Eds). Lecture Notes in Artificial Intelligence, Advances in Web Intelligence, 2663, (pp. 113–123). Springer-Verlag.

Last, M., Markov, A., & Kandel, A. (2006). Multi-lingual detection of terrorist content on the Web. In H. Chen, F. Wang, C.C. Yang, D. Zeng, M. Chau, & K. Chang (Eds.), Proceedings of the PAKDD'06 International Workshop on Intelligence and Security Informatics. Lecture Notes in Computer Science, 3917, 16–30.

Leckie, T., & Yasinsac, A. (2004). Metadata for anomaly-based security protocol attack deduction. IEEE Transactions on Knowledge and Data Engineering, 16(9), 1157–1168.

Lemos, R. (2002). What are the real risks of cyberterrorism? Retrieved November 29, 2007 from http://zdnetcom.com/2100-1105-955293.html

Lieberman, J., & Collins, S. (2008). Violent Islamist extremism, The Internet, and the homegrown terrorist threat. United States Senate Committee on Homeland Security and Governmental Affairs. Retrieved November 14, 2009, from http://hsgac.senate.gov/public/_files/IslamistReport.pdf

Lyall, S. (2006). London bombers tied to Internet, not Al Qaeda. April 11, 2006. Retrieved November 29, 2007 from http://www.nytimes.com/2006/04/11/world/europe/11london.html

Mao, Y., Chen, K., Wang, D., Zheng, W., & Deng, X. (2001). MOT: Memory online tracing of Web information system. In T. Ozsu, H.-J. Schek, K. Tanaka, Y. Zhang, & Y. Kambayashi (Eds.), Proceedings of the Second International Conference on Web Information Systems Engineering (WISE'01) (pp. 271–277). Los Alamitos, CA: IEEE.

MSNBC News Services. (2006). Pentagon surfing 5,000 Jihadist Web Sites. Retrieved March 22, 2008 from http://www.msnbc.msn.com/id/12634238/

Mobasher, M., Cooley, R., & Srivastava, J. (2002). Automatic personalization based on Web usage mining. Communications of the ACM, 43(8), 142–151.

Office of Homeland Security, The White House, USA. (2002). The first national strategy for homeland security. Retrieved November 14, 2009, from http://www.dhs.gov/xlibrary/assets/nat_strat_hls.pdf

Perkowitz, M., & Etzioni, O. (2000). Towards adaptive Web sites. Artificial Intelligence, 118(1–2), 245–275.

Popp, R., Armour, T., Senator, T., & Numrych, K. (2004). Countering terrorism information technology. Communications of the ACM, 47(3), 36–43.

Provos, N., & Honeyman, P. (2002). Detecting steganographic content on the Internet Proceedings of the Network and Distributed Systems Securi Symposium.

Qin, J., Zhou, Y., Reid, E., Lai, G., & Chen, H. (2006). Unraveling international terrorist groups' exploitation of the Web: technical sophistication, media richness, and Web interactivity. In H. Chen, F. Wang, C.C. Yang, D. Zeng, M. Chau, & K. Chang (Eds.), Proceedings of the PAKDD'06 International Workshop on Intelligence and Security Informatics. Lecture Notes in Computer Science, 3917, 4–15.

Salton, G., & Buckley, C. (1998). Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), 513–523.

Sequeira, K., & Zaki, M. (2002). ADMIT: Anomaly-based data mining for intrusions. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD '02) (pp. 386–395). New York: ACM Press.

Shahabi, C., Zarkesh, A., Adibi, J., & Shah, V. (1997). Knowledge discovery from users Web-page navigation. Proceedings of the IEEE 7th International Workshop on Research Issues in Data Engineering, (pp. 20–30).

Shapira, B., Elovici, Y., Last, M., & Kandel, A. (2008). Enhancement to the Advanced Terrorists Detection System (ATDS). In P. Kantor and B. Shapira (Eds). Security Informatics and Terrorism—Patrolling the Web (pp. 71–81). Fairfax, VA: IOS Press.

Symonenko, S., Liddy, E.D., Yilmazel, O., DelZoppo, R., Brown, E., & Downey, M. (2004). Semantic analysis for monitoring insider threats. In H. Chen, R. Moore, D.D. Zeng, & J. Leavitt (Eds.): Intelligence and Security Informatics, Proceedings of the Second Symposium on Intelligence and Security Informatics (ISI 2004). Lecture Notes in Computer Science, 3073, 492–500.

Turney, P.D. (2000). Learning algorithms for keyphrase extraction. Information Retrieval, 2(4), 303–336.

Extractor. (2003). The world of relevant information in the palm of your hand. Retrieved from http://www.extractor.com

Wagner, A.R. (2005). Terrorism and the Internet: Use and abuse. In M. Last & A. Kandel (Eds.), Fighting terror in cyberspace. World Scientific, Series in Machine Perception and Artificial Intelligence, 65, pp. 1–25.

Wu, H., Gordon, M., DeMaagd, K., & Fan, W. (2006). Mining Web navigations for intelligence. Decision Support Systems, 41(3), 574–591.

Zakin, O., Levi, M., Elovici, Y., Rokach, L., Shafrir, N., Sinter, G., et al. (2007). Identifying computers hidden behind a network address translator (nat) using machine learning techniques. In Proceedings of the Sixth European Conference on Information Warfare and Security (ECIW '07) (pp. 335–340). Retrieved November 16, 2009, from http://www.academic-conferences.org/pdfs/eciw07-booklet.pdf

Zhou, Y., Reid, E., Qin, J., Chen H., & Lai, G. (2005). U.S. domestic extremist groups on the Web: Link and content analysis. IEEE intelligent systems, 20(5), 44–51.