IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 8, 2019, accepted July 24, 2019, date of publication August 6, 2019, date of current version August 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933197

Sec-Lib: Protecting Scholarly Digital Libraries From Infected Papers Using Active Machine Learning Framework

NIR NISSIM[1,2], AVIAD COHEN[1,3], JIAN WU[4], ANDREA LANZI[5], LIOR ROKACH[1,3], YUVAL ELOVICI[1,3], AND LEE GILES[6]

[1]Malware Lab, Cyber Security Research Center (CSRC), Ben-Gurion University, Beersheba 84105, Israel
[2]Department of Industrial Engineering and Management, Ben-Gurion University, Beersheba 84105, Israel
[3]Department of Software and Information Systems Engineering, Ben-Gurion University, Beersheba 84105, Israel
[4]Computer Science Department, Old Dominion University, Norfolk, VA 23529, USA
[5]Computer Science Department, University of Milan, 20122 Milan, Italy
[6]Computer Science and Engineering Department, Pennsylvania State University, State College, PA 16801, USA

Corresponding author: Nir Nissim (nirni.n@gmail.com)

ABSTRACT Researchers from academia and the corporate-sector rely on scholarly digital libraries to access articles. Attackers take advantage of innocent users who consider the articles' files safe and thus open PDF-files with little concern. In addition, researchers consider scholarly libraries a reliable, trusted, and untainted corpus of papers. For these reasons, scholarly digital libraries are an attractive-target and inadvertently support the proliferation of cyber-attacks launched via malicious PDF-files. In this study, we present related vulnerabilities and malware distribution approaches that exploit the vulnerabilities of scholarly digital libraries. We evaluated over two-million scholarly papers in the CiteSeerX library and found the library to be contaminated with a surprisingly large number (0.3-2%) of malicious PDF documents (over 55% were crawled from the IPs of US-universities). We developed a two layered detection framework aimed at enhancing the detection of malicious PDF documents, Sec-Lib, which offers a security solution for large digital libraries. Sec-Lib includes a deterministic layer for detecting known malware, and a machine learning based layer for detecting unknown malware. Our evaluation showed that scholarly digital libraries can detect 96.9% of malware with Sec-Lib, while minimizing the number of PDF-files requiring labeling, and thus reducing the manual inspection efforts of security-experts by 98%.

INDEX TERMS Scholarly, digital, library, paper, PDF documents, malware, malicious documents, distribution.

I. INTRODUCTION

The number of scholarly documents (English language) accessible on the Web is enormous, estimated at over 114 million PDF documents [5], of which over 27 million (∼24%) can be easily accessed without payment or subscription [5]; since then, the estimated number of scholarly documents on the Web raised significantly. These documents are freely available in part because researchers publish draft versions of their papers on their professional home pages (often within the domains of universities), before the final versions are published by the publishers.

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

Researchers also publish their research on their home pages to increase exposure, reach researchers around the world, and gain citations and recognition for their work [6], [7]. In order to assist researchers, many scholarly digital libraries and search engines collect and index the author's version. Thus, the papers can be easily downloaded worldwide. This free collection of scholarly documents is a valuable resource for most researchers and academics who may not have a comprehensive subscription to all publishers' content.

Figure 1 presents a snapshot of search results for a searched paper using Google Scholar. At the bottom of the page, one can access all 15 versions of the paper, already indexed by Google Scholar, simply by clicking on the blue

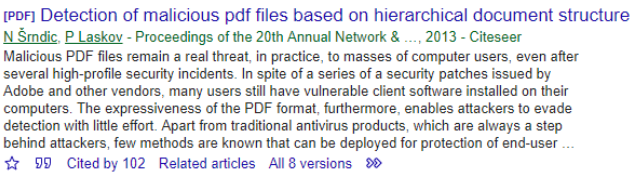

FIGURE 1. Google Scholar's search results for a given academic paper, including 14 additional versions of the paper.

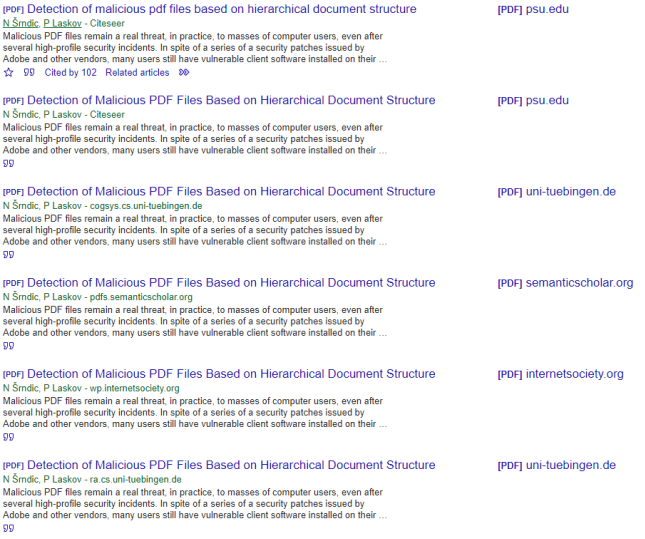

FIGURE 2. Some of the additional versions of the searched paper, including those available for free.

"All 8 versions" link; in this way, free and convenient versions are literally at the user's fingertips, as seen in Figure 2. Note that there are several versions from different sources (e.g., psu.edu, uni-tuebingen.de, semanticscholar.org).

Researchers heavily use scholarly digital libraries to access and download scholarly documents [33]. For example, according to a survey by EBLIDA,[1] there are a total of 5,974 academic libraries in Europe; however, this number is far from complete given that it is based on information provided by only 25 countries participating in the survey. Nevertheless, the number of registered users of these libraries is 39,328,294. As Europe represents only part of the world's research activity, the global usage of scholarly digital libraries is much higher.

Universities are considered to be highly reputable institutions that primarily focus on research, the goal of which is to contribute new and valuable knowledge to the world. Therefore, they are considered a trusted content source with no malicious intent. Correspondingly, the websites of their academics and researchers (which often reside on the institution's network domain) are also thought to contain only trusted content, free of malicious PDF documents. In a circular fashion, academic digital libraries tend to harvest these allegedly trusted sites without hesitation or fear, and do not even scan them to detect malicious content.[2] In addition, their reputation as sources of trusted scholarly documents makes digital libraries an attractive platform from which to take advantage of and distribute malicious PDF documents. Attackers are aware of this chain of trust and use social engineering techniques in which they take advantage of the heavy use and blind trust of researchers in scholarly digital libraries and the papers (PDF documents) they download from them; once one researcher within an organization is infected, it can quickly become a major cyber security incident for the entire organization's computational system [32]. Researchers' Web pages have become a target that can be used to launch attacks.[3] In addition, researchers, professors, and research students are naturally attractive candidates for attack, because, due to the nature of their work, they have access to confidential and sensitive information, such as nuclear knowledge, medical records, aviation, and educational records and materials (e.g., student data, exams, etc.). Moreover, some researchers collaborate with governmental agencies and industry, which allows them access to national and confidential information from governments (such as computational criminology), national institutions, and companies' intellectual property, while other researchers collaborate with healthcare institutions and hospitals, and are connected to their networks and computerized systems [34].

Previous studies have presented many methods of improving the detection of malicious PDF documents [1], [2]. These studies focused on detection techniques based on analyzing the malicious PDF documents when they have already been downloaded to the host machine. To the best of our knowledge, no study has addressed the issue at an earlier stage, before downloading, a stage at which it might be possible to prevent malicious PDF documents from being mass distributed through legitimate channels and exiting platforms, and thus, markedly improve the detection of malicious PDF documents, including those found on popular, well-known, and extensively used sources of PDF documents, such as scholarly digital libraries. These libraries can be intentionally used as a free and very successful platform for distributing PDF malware quickly and easily to a desired group of victims with access to valuable information. An academic paper arouses little suspicion, particularly if an attacker wants to distribute a new zero-day attack quickly in the shape of a benign PDF document. Zero-day attacks[4] utilizes new attack techniques or new vulnerabilities[5] that are difficult to detect, particularly by the anti-virus tools commonly used by organizations such as universities and academic digital libraries for scanning PDF documents. Thus, these libraries can easily be used as a new and convenient platform for

[1] http://www.eblida.org/activities/kic/academic-libraries-statistics.html

[2] According to the CiteSeerX team, some of whom are authors of this paper.

[3] http://www.nytimes.com/2013/07/17/education/barrage-of-cyberattacks-challenges-campus-culture.html?pagewanted=all&_r=0.

[4] http://www.bullguard.com/bullguard-security-center/pc-security/computer-threats/what-are-zero-day-attacks.aspx

[5] http://www.pctools.com/security-news/zero-day-vulnerability/

distributing zero-day attacks. One should take into consideration that these libraries are a very dynamic environment in which a large number of papers are added daily; thus, even advanced detection solutions for PDF malware based on machine learning algorithms [1] will lose their generalization capabilities regarding the detection of unknown PDF malware, since both for benign and malicious PDF documents the data and patterns concealed in them change rapidly and often. Consequently, an efficient and frequent update process is required in order to maintain and improve the detection of PDF malware in light of the daily additions of new PDF documents into these scholarly digital libraries.

Smutz and Stavrou [52] suggested a variant of the query by committee [55] method to improve the detection of PDF classifier. Their method included mutual agreement analysis which was found effective at identifying specific samples to be added to the training set, resulting in significant improvement in the classifier's performance compare to when random sampling method was used. However, this method requires maintaining an ensemble of different classifiers, a requirement that can be dismissed by using methods of another active learning approach that enables the use of only one classifier.

Keeping pace with newly created documents was demonstrated in recent years by Nissim *et al.* [11], [15], [20], using active learning (AL) methods [38], which have been integrated and used in the solution we present in this study. The contributions of our paper are fourfold:

1) We are the first to demonstrate the vulnerability of large public databases such as scholarly digital libraries, as well as the first to present several simple approaches and techniques that can be used to compromise these libraries and utilize them for malicious purposes.
2) We are the first to reveal the malicious use of scholarly digital libraries and also the first to estimate the extent of this phenomenon. Using current anti-viruses, after indexing we were able to assess which papers contained malware. To support our findings and emphasize the severity of the phenomenon, we also performed further analysis one year after our initial evaluation of papers that were collected by the CiteSeerX digital library over a period of eight years.
(3) We are the first to investigate very large databases, such as scholarly digital libraries that are traditionally thought to be secure and harmless, and showed how they can be maliciously used as a platform for malware distribution and leveraged for targeted cyber security attacks. We also evaluated the impact of the presence of malicious documents in a scholarly digital library.
(4) In this study we also present and implement Sec-Lib, a comprehensive and adaptive detection framework aimed at enhancing the security of very large databases such as scholarly digital libraries. Sec-Lib encompasses both deterministic and advanced machine learning approaches (e.g., active learning for updating based on the frequent changes in these libraries) in order to provide a comprehensive solution for the detection of malicious papers (PDF malware) in scholarly digital libraries. While the machine learning approaches used in Sec-Lib are based on our previous work [11], we have extended and improved our original approach, by adding deterministic detection layers to Sec-Lib and rigorously and empirically evaluating Sec-Lib, using real data originating from a scholarly digital library and demonstrating that it can substantially reduce the number of malicious PDFs included in these libraries.

II. BACKGROUND

As indicated previously, the Web contains more than 114 million scholarly documents [5], and this number represents a significant attack surface for adversaries who want to take advantage of the fact that scholarly digital libraries are considered trusted and their content (PDF documents) is downloaded by many users worldwide. In order to grasp the potential harm that can be caused by the presence of malicious PDF documents in a scholarly digital library, we briefly describe targeted attacks conducted via scholarly digital libraries using malicious PDF documents. Then, we present the possible attacks that can be launched by a malicious PDF document mistakenly considered a benign scholarly document, and the techniques used to achieve this. In so doing, we aim to raise the awareness of scholarly digital libraries, as well as innocent researchers and readers, regarding the power of a malicious PDF document, so that they will increase their vigilance against such attacks and employ the best security means possible.

A. TARGETED ATTACKS VIA SCHOLARLY DIGITAL LIBRARIES USING MALICIOUS PDF DOCUMENTS

Sophisticated attackers interested in sensitive and novel knowledge about a specific domain, such as nuclear energy, can launch a targeted attack by inserting an attractive, yet malicious, paper that addresses nuclear energy into digital libraries, engaging and tempting researchers to download the paper. It is noteworthy that the attacker does not need to be a co-author of the paper. Our investigation showed that most scholarly digital libraries (such as Google Scholar) crawl academic websites and index the papers they find, disregarding any mismatches between the author's affiliation and the website that stores the paper. Thus, an attacker can take a popular paper written by someone else, inject malicious code into it, and upload it to several websites. When the victim opens the malicious PDF document, a malicious code will be executed in the victim's computer. This malicious code may allow the attacker to exfiltrate data from the victim's machine and send it to a remote server controlled by the attacker.

This type of attack is within the realm of reality, for the previously mentioned reasons, and also because users consider non-executable files safer than executables, and thus are less suspicious of PDF documents, especially when downloaded from popular and trusted scholarly sources. Unfortunately, non-executable files such as PDF documents are as dangerous as executable files, since their readers can contain

vulnerabilities that, when exploited, can allow an attacker to execute malicious actions on the victim's computer. Symantec's Internet Security Threat Report (ISTR[6]) 2016 indicates that malicious PDF documents are used for targeted attacks, especially via malicious emails. Note that since that time, the number of targeted attacks on Adobe Reader has almost doubled. In the following section, we elaborate on several of the most common techniques and attacks involving the use of malicious PDF documents.

B. POSSIBLE ATTACK TECHNIQUES USING PDF DOCUMENTS

Before explaining how scholarly digital libraries can be easily used as a platform to leverage and distribute attacks worldwide, we present some of the many ways PDF documents can be used maliciously when created or manipulated by an attacker.

1) JAVASCRIPT CODE

PDF documents may contain embedded JavaScript code or code retrieved from URIs [22], including 3D content, form validation, and mathematical calculations. Typically, a malicious JavaScript code in a PDF document attempts to exploit a vulnerability in the PDF viewer in order to divert the normal execution flow to the embedded malicious code. This is achieved by a heap spraying[7] attack. JavaScript also allows the download of an executable file that may contain malicious content. Alternatively, JavaScript code can access websites, whether malicious or benign.

2) CODE OBFUSCATION AND FILTERS

Code obfuscation is used legitimately to prevent reverse engineering of proprietary applications. However, it can also be used by attackers to hide malicious content. Filters are used in PDFs to compress data for encoding and reducing file size, and are frequently used by attackers to conceal malicious content. Available filters and their primary purposes are discussed by Baccas and Kittilsen [23], [24].

3) EMBEDDED FILES

A PDF document can contain other file types, such as HTML, JavaScript, SWF, XLSX, EXE, or even another PDF document, which can be used to embed malicious files that are frequently obfuscated. When special techniques are applied, the embedded file can be opened without alerting the user. Maiorca *et al.* [25] presented a novel evasion technique called ''reverse mimicry,'' which was designed to evade state-of-the-art malicious PDF detectors based on their logical structure[8] [21]. Mimicry attacks inject malicious content into a benign PDF while maintaining its benign structure. This method can be automated easily and does not require knowledge of the structural features used in the maliciousness detector.

4) FORM SUBMISSION AND URI ATTACKS

Hamon [4] presented practical techniques that can be used by attackers to execute malicious code from a PDF document. The author showed that security mechanisms, such as the Protected Mode of Adobe Reader X or the URL Security Zone Manager of Internet Explorer, can easily be disabled by changing the corresponding registry key. Moreover, a URI[9] address can be used (instead of a URL), directing the user to any type of file located remotely, including executables.

It should be noted that Adobe Reader version X, released in 2011, included a new sandbox isolated environment, Protected Mode Adobe Reader (PMAR), that ensures that malicious code operations cannot affect the operating system. Nevertheless, most organizations (including universities) do not keep up with the newest versions of PDF readers, and thus, are exposed to many of the well-known attacks.

C. ANALYZING VULNERABILITIES OF POPULAR SCHOLARLY DIGITAL LIBRARIES

Now, we briefly present the most popular libraries, their market share, and their uniqueness, and then explain the vulnerabilities that exist within them. In addition, we present new vulnerabilities that we utilized to demonstrate potential attacks. We present three libraries in which we found a vulnerability.

1) GOOGLE SCHOLAR

Google Scholar[10] is a free public Web search engine for scholarly literature. It indexes nearly 100 million scholarly documents and is considered the largest scholarly digital library, encompassing 87% of the total number of the existing scholarly documents [5]. Current articles are indexed and can be found when searched. A user clicking on an article that appears on the results page of Google Scholar is usually directed to the article's Web page on the publisher's official website. In addition to articles on the publisher's website, other versions of the papers, from other places on the Web are also indexed (e.g., papers from a researcher's Web page on an academic institution's website).

In order to demonstrate contamination of a digital library such as Google Scholar, we used the Web page of a researcher at a known university (details are not provided for privacy reasons). The articles on the researcher's Web page were indexed by Google Scholar previously and can be accessed by clicking the ''All X versions'' link under the relevant article in Google Scholar, as shown in Figure 1. With no connection to the researchers' names appearing in Figure 1, in our case, after we had obtained the permission of the researcher

[6]https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

[7]Heap Spraying - A technique used in exploits to assist random code execution.

[8]PDF logical structure is a hierarchy of structural elements, each represented by a dictionary (see the PDF file structure section).

[9]**URI** – ''a compact string of characters for identifying an abstract or physical resource,'' RFC2396. A URI is an extension of a URL, used for identifying any Web resource (not limited to Web pages).

[10]https://scholar.google.co.il/

mentioned above, we downloaded the most popular paper (a PDF document) from his web page and injected a malicious JavaScript into it using a PDF editing program called *PDF-Fill*[11] (such that the malicious JavaScript code is launched when the new article's file is opened). Then, we replaced the benign paper with this new malicious version of the paper on the researcher's website. Now, the malicious paper is available for downloading through Google Scholar using the original indexing information that was neither changed nor updated in light of the replacement of the paper behind the published URL. The vulnerability in Google Scholar lies in the indexing mechanism, which checks only the title and author's name and pays no attention to whether a new file was uploaded with the same title and author's name.

As far as we could determine, Google Scholar does not verify that the uploaded paper is related to the researcher's home page. Thus, a malicious PDF document that carries the same title and authors of a popular paper can easily be created and placed on other Web pages unconnected to a researcher's home page within a university. These malicious papers can easily be promoted with an acceptable payment to Google for a promoted link. In this way, the attacker uses several elements of existing tools and services to launch his/her attack. First, the attacker takes advantage of the popularity of a particular paper, second he/she uses the fact that Google Scholar is a trusted source of information, and third the attacker exploits a vulnerability in the Google Scholar indexing mechanism. Consequently, the attacker achieves his/her attack goals by redirecting the download traffic to the malicious version of the paper.

2) CiteSeerX

CiteSeerX[12] is a growing scientific literature digital library and search engine that focuses primarily on literature in the areas of computer and information science. It is unique in that it collects papers solely from researchers' home pages from the domains of universities and physically stores the papers themselves and creates links to the papers. The result is that the library contains over four million academic papers in PDF format, and its total size is estimated at about 3.8 terabyte.

According to the way in which CiteSeerX collects academic documents, we identified several methods by which a malicious PDF paper could be indexed and stored by a popular digital library. A malicious paper could be uploaded to a researcher's website directly. This can happen unintentionally if the paper was infected by a malware resident on the researcher's computer before it was placed on the website. Alternatively, the paper could be contaminated using a free, malicious PDF creator that injects malicious code into the edited papers. Another likely scenario is that the researcher's page could be hacked, with the attacker replacing a benign paper with a malicious one. In each of these examples, a malicious paper finds its way to the researcher's home page within an academic institution's trusted domain, making it available for uploading by CiteSeerX, as well as to the general public worldwide.

3) SOCIAL NETWORK BASED SCHOLARLY DIGITAL LIBRARIES

ResearchGate[13] (founded in 2008) is a social networking site for scientists and researchers, enabling them to share papers, communicate, and find collaborators. Today, it has more than six million members. ResearchGate is also considered an academic digital library as its members can upload and share papers with other members.

Academia.edu[14] (launched in September 2008) is a platform for academics for sharing research papers, monitoring their impact, and following researchers in a particular field. Over 63 million academics have signed up to Academia.edu, adding ∼21 million papers. Academia.edu attracts over 19 million unique visitors a month.

ResearchGate and Academia.edu are examples of scholarly academic digital libraries affiliated with social networks for researchers whose purpose it is to share data, papers, and knowledge with other researchers.

To utilize scholarly digital libraries for malware distribution, an attacker can create a fictitious profile of a famous researcher through Academia.edu. The attacker can then upload several of the researcher's well-known papers in order to boost the profile's credibility and gain the trust of colleagues. After several weeks, when the profile is active and papers have been downloaded from the profile, the attacker can easily upload a malicious version of the same papers in order to perform an attack. A malicious PDF document (a non-zero-day malicious PDF document) that should be recognized by an anti-virus tool will not be rejected when uploaded, since such scholarly digital libraries have limited security mechanisms. In this way, social relationships and trust can be used in order to leverage to a social network-based library for the distribution of a PDF malware.

4) ADDITIONAL EXISTING SCHOLARLY DIGITAL LIBRARIES

The following are additional existing scholarly digital libraries that we have not yet checked for vulnerabilities; however, we assume that vulnerabilities exist and should be further investigated.

Microsoft Academic Search[15] (MAS) is a free public Web search engine for academic papers and literature, developed by Microsoft Research for the purpose of algorithm research on object-level vertical search, data mining, entity linking, and data visualization. Microsoft Academic Search consists of almost 50 million scholarly documents and is considered one of the top alternatives to Google Scholar [5].

[11] http://www.pdfill.com/
[12] http://csxstatic.ist.psu.edu/about
[13] http://www.researchgate.net/
[14] https://www.academia.edu/
[15] http://academic.research.microsoft.com/

Web of Science[16] is an online subscription-based scientific citation indexing service maintained by Thomson Reuters that provides comprehensive citation search. It consists of nearly 50 million scholarly documents and is, along with MAS, among the largest academic digital libraries after Google Scholar [5]. One should note that Web of Science does not index the PDF documents, as Google Scholar does.

PubMed[17] is a free search engine that primarily accesses the MEDLINE database of references and abstracts on life sciences and biomedical topics. The United States National Library of Medicine at the National Institutes of Health maintains the database as part of the Entrez system of information retrieval. PubMed comprises over 24 million citations of biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publishers' websites.

arXiv is an automated electronic repository and distribution server for research articles, consisting of electronic preprints of scientific papers in the fields of mathematics, physics, astronomy, computer science, quantitative biology, statistics, and finance, which can be accessed online. Almost all scientific papers within arXiv are self-archived, meaning that they were uploaded by the users themselves. Table 1 summarizes some interesting details about the scholarly digital libraries mentioned in this section, as well some libraries from the Darknet.

TABLE 1. Summary of Scholarly digital libraries' details regarding to their crawling, indexing and redirecting approaches to the scholarly documents.

Scholarly digital libraries	Upload by User (Preprint)	Crawling Publisher	Crawling Authors Homepage (Preprint)	Indexing the PDF content	Store the PDF	Link to the original PDF	Number of Scholarly documents (In Millions)
Google Scholar	No	Yes	Yes	Yes	No	Yes	99.3
Microsoft Academic	No	Yes	Yes	Yes	No	Yes	50
Web of Science	No	Yes	No	No	No	Yes	50
CiteseerX	Yes	No	Yes	Yes	Yes	Yes	4.2
PubMed	No	Yes	No	Yes	No	Yes	24
Research Gate	Yes	No	No	Yes	Yes	No	Unknown
Academia.edu	Yes	No	No	Yes	Yes	No	5
arXiv	Yes	No	No	Yes	Yes	No	1
http://libgen.org/scimag (Darknet)	Yes	Yes	No	No	Yes	No	36
http://sci-hub.org/ (DarkNet)	No	Yes	No	No	Yes	No	Unknown
http://booksc.org/ (DarkNet)	No	Yes	No	No	Yes	No	18

The largest libraries, Google Scholar, MAS, and Web of Science, do not rely on papers uploaded by users as they collect (crawl) papers from the publishers and do not store them. Note that there are several closed grouped libraries within the Darknet, such as Libgen, Sci-hub and Booksc, and we assume that specifically in these closed libraries the probability and percentage of malicious papers is higher than in the known and wide-open libraries. This assumption should be scrutinized in future research.

[16]https://apps.webofknowledge.com/

[17]http://www.ncbi.nlm.nih.gov/pubmed

III. SCANNING CiteSeerX FOR PDF MALWARE

As part of this collaborative study with the CiteSeerX team, we scanned and analyzed the CiteSeerX digital library as our dataset. Our goal was to determine whether this platform had already been used, either intentionally by an attacker or unintentionally by an innocent researcher, to distribute malicious PDF documents, and in so doing, to measure the extent of harm that can be caused by such a scenario. When we began scanning, the CiteSeerX library contained 4,044,118 academic papers in PDF file format that were collected until the time this study began. The papers originated from more than 188 countries and most continents and were written by 1.3 million different authors from 4,963 universities.

We used the VirusTotal[18] service to scan the entire CiteSeerX library for malicious PDF documents. VirusTotal, a subsidiary of Google, is a free online service that provides comprehensive analysis of files and websites (URLs) by a set of ∼63 anti-virus engines and website scanners. VirusTotal allows a user to submit suspicious files for analysis. After the analysis, VirusTotal provides a report that lists suspicious files identified by each of the anti-virus engines. Note that we considered a PDF document malicious, if at least five different anti-viruses identified it as a malicious file.

A. SCANNING RESULTS OF CiteSeerX IN 2015

In this section we present the results and provide an analysis of the results regarding the process of scanning of the PDF documents within the CiteSeerX library. We provide an analysis of aspects of both crawling and downloading the malicious papers, on the basis of a worldwide breakdown. Some of these scanning results were presented in our preliminary paper published recently [19], however in the current work we explain it more comprehensively, and also present our novel detection methodology, Sec-Lib, and its evaluation in terms of the detection of PDF malware in scholarly digital libraries.

1) CRAWLED MALICIOUS PAPERS

Of the 4,044,118 PDF files that were submitted for analysis from the CiteSeerX library, only 2,586,820 were actually scanned by VirusTotal due to our license and network bandwidth limitations. Of these files, 753 (∼0.3%) were found and classified as malicious by VirusTotal's anti-virus engines. Figure 3 presents the breakdown of the threats identified.

The threat categories were provided by the identifying anti-virus engine. As can be seen in Figure 3, 72% of the malicious files were identified a vulnerability exploitation.[19] Usually, a vulnerability in the PDF document format is exploited utilizing an embedded JavaScript code.[20] 9.5% of the malicious files were classified as a Trojan, a malicious program that when executed performs covert actions that have not been permitted by the user. 7.5% of the malicious files

[18]https://www.virustotal.com/

[19]http://searchsecurity.techtarget.com/definition/exploit

[20]http://blogs.technet.com/b/mmpc/archive/2013/04/29/the-rise-in-the-exploitation-of-old-pdf-vulnerabilities.aspx

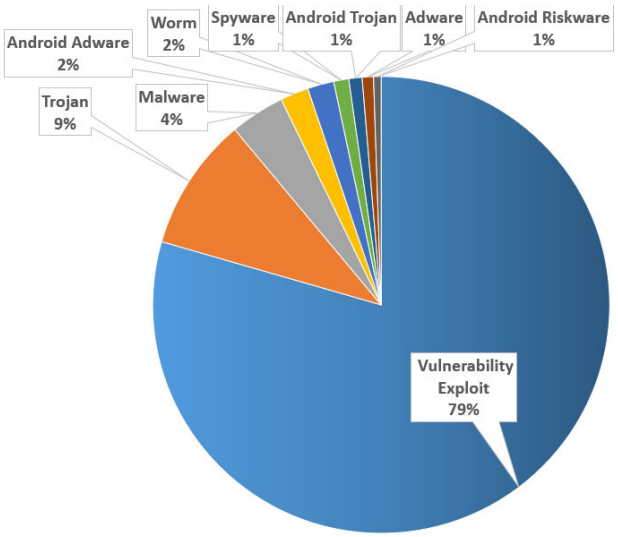

FIGURE 3. Breakdown of the threats identified among the 753 malicious PDF documents found by VirusTotal on the CiteSeerX library.

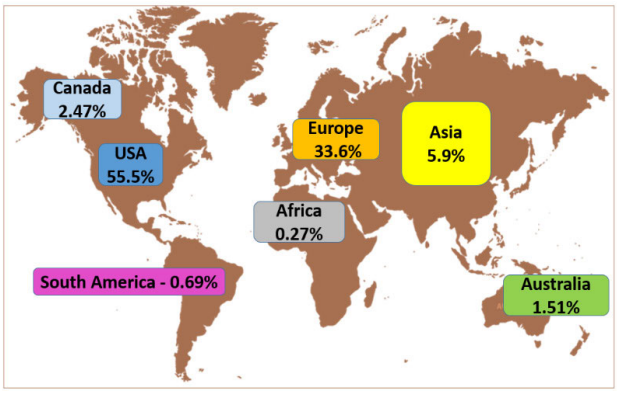

FIGURE 4. Distribution of the malicious scholarly documents based on the geographical location from which they were crawled by CiteSeerX scholarly digital library.

contained JavaScript code that was recognized as malicious. JavaScript code can be identified as malicious although it does not exploit any vulnerability and is considered malicious when the code signature is known to represent a malicious code. 3.9% of the malicious files were classified as malware, which means that malicious software (e.g., Exe, PDF, etc.) was found embedded in them. 3.4% of the malicious files contained a threat (Adware,[21] Trojan, or Riskware[22]) targeting the Android operating system widely used on mobile devices. 1.9% of the malicious files contained a computer worm,[23] which is a malicious program that can propagate by autonomously copying itself from one machine to another. A small percentage of files (1.1%) were classified as Spyware,[24] which is a malicious computer program aimed at collecting personal information from the victim's computer. Although it does not damage the victim's computer, it can cause damage to the victim by stealing sensitive information. Adware is a program aimed at supporting advertising and operates without the user's permission. An additional 5,775 files were identified as malicious by the Fortinet anti-virus, because they contained a suspicious threat called "HTML/Redirector.BK!tr." These files might be malicious, since they may direct the user to malicious destinations such as websites, IP addresses, and servers. Deeper analysis is required to reach a final decision; however, in case these files would be found to be malicious, and the percentage of malicious PDF documents in the CiteSeerX library will increase from 0.3% to 2%, this situation will emphasizes the phenomenon of scholarly digital library contamination we describe in this paper.

[21]http://www.pctools.com/security-news/what-is-adware-and-spyware/
[22]http://usa.kaspersky.com/internet-security-center/threats/riskware
[23]http://www.pctools.com/security-news/what-is-a-computer-worm/
[24]http://www.microsoft.com/security/pc-security/spyware-whatis.aspx

Figure 4 presents the distribution of the malicious scholarly documents according to the geographical location from which they were crawled by the CiteSeerX scholarly digital library. More than 55% of the malicious papers in CiteSeerX were crawled from IP's belonging to US universities, whereas about 33% were crawled from IP's belonging to European universities. Asia includes several countries, e.g., China, Russia, and Korea, which on the one hand are known to have a large population of researchers and on the other have been found to be the origin of many malware samples. Because of this, we were surprised to find that only 5.9% of the malicious papers were crawled from IP's belonging to an Asian institution; we did not find any malicious papers crawled from Russia or Korea which are the origin of a great deal of malicious content, especially malicious Android applications found in application markets [10].

In Table 2, we can see the top 11 European countries in terms of the percentage of malicious scholarly documents crawled from researcher's homepages and websites associated with IP addresses belonging to these countries. While Germany was found to be the origin of 10.7% of the malicious

TABLE 2. Breakdown of the distribution of the malicious scholarly documents based on the geographical location from which they were crawled by CiteSeerX scholarly digital library (of the entire world's malicious papers).

Country	Percentage
Germany	10.70%
United Kingdom	6.04%
Holland	2.74%
France	2.61%
Austria	2.33%
Luxembourg	2.06%
Sweden	0.82%
Switzerland	0.82%
Denmark	0.69%
Italy	0.55%
Turkey	0.55%

papers in CiteSeerX among all of the countries around the world, it is the origin of more than 31% of the malicious papers in CiteSeerX among all European countries. It was followed by the United Kingdom (6.04%), Holland (2.74%), and France (2.61%). The other European countries not mentioned here were the origin of less than 0.41% of malicious scholarly documents of the entire world's malicious papers in CiteSeerX.

2) DOWNLOADED MALICIOUS PAPERS

We now present the impact and power of an attack stemming from malicious papers published in a scholarly digital library. Using CiteSeerX's database and its website's historic log files, we extracted and aggregated information regarding the download data of the malicious papers we found. We faced a 'big data' problem due to the enormous amount of data we needed to extract and process, and therefore, we extracted the downloading information for only the top 31 malicious papers identified by a large number of anti-virus engines out of the total 723 malicious papers that were found. We focused on the download statistics for the five preceding years, and therefore, we can provide conclusions regarding relevant download trends. In addition, we also used GNU Parallel[25] to boost the speed and reduce the very long running time. The data comprised 5197 successful downloads of malicious papers (during 2009-2014) which resulted from just 31 malicious papers crawled by CiteSeerX's, meaning that scholarly digital libraries have had an average 'damage coefficient' of167 in the last five years.

Note that the papers are downloaded based on the interest they create among readers in the scientific community, regardless of how malicious and dangerous they are. Therefore, the number of times papers are downloaded is not dependent on their level of maliciousness. Thus, the papers that were identified as malicious are in fact randomly selected in terms of the number of downloads; consequently, this number is not affected when other (more malicious or less malicious)randomly selected papers are chosen @perioBy calculating the damage coefficient, our goal was to present the damage rate for the most dangerous papers and emphasize the importance of the phenomenon of scholarly digital library contamination with malicious papers.

The average number of different countries that downloaded malicious papers was 16, covering all of the continents (except for Antarctica), which constitutes extremely wide coverage of the world's research population within universities and other institutions. Table 3 presents information regarding the top 20 most downloaded malicious papers (from CiteSeerX) during the last five years. The most downloaded malicious paper is on the topic of computer forensics and apparently was a malicious version of a very popular paper; it was downloaded 2213 times in 108 different countries on all continents (except for Antarctica). The popular topics among malicious papers were related to computers, such as cyber security and computer science.

[25]http://www.gnu.org/software/parallel/

TABLE 3. Top 20 most downloaded malicious scholarly documents during the last five years, their country of origin, and the number of countries in which they were downloaded.

Paper's Topic	Country of Origin	Total Number of Downloads	Number of Countries
Computer Forensics	USA	2213	108
Network Security	France	860	77
Computer Hardware	Germany	633	59
Computer Networks	Germany	480	52
Social Networks	USA	235	51
Machine Learning	Germany	123	20
Mathematics	Germany	90	12
Computer Science	USA	79	11
Software Engineering	BVI	77	4
Mathematics	USA	57	7
Computer Science	USA	57	4
Sociology	Brazil	46	10
Economics	USA	42	4
Computer Science	China	35	7
Computer Science	France	23	7
Computer Science	Holland	20	3
Astronomy	USA	20	3
Medical	USA	17	5
Physics	USA	13	4
Meteorology	Canada	13	3

Figure 5 presents the distribution of the malicious scholarly documents according to the geographical location from which they were downloaded from CiteSeerX scholarly digital libraries. More than 40% of the malicious papers in CiteSeerX were downloaded from US IPs, whereas about 28% were downloaded from Asian IPs.

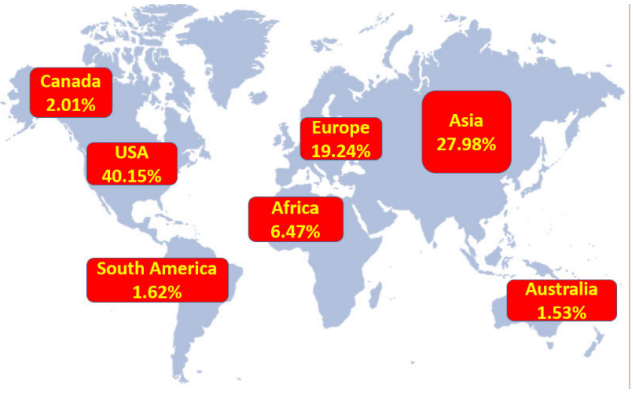

FIGURE 5. Distribution of the malicious scholarly documents according to the geographical location from which they were downloaded from CiteSeerX scholarly digital library.

Figure 4 shows that the US was the origin of more than 55% of the malicious papers in CiteSeerX. The data presented in Figure 5 and Table 4 indicates that the US was also the most popular destination of malicious papers and the location of more than 40% of the downloads of malicious papers, followed by India (9.52%), China (5.04%), and the UK (3.77%). As can be seen, using a scholarly digital library as a platform,

TABLE 4. Top countries downloading most of the malicious scholarly documents from CiteSeerX during the last five years.

Country	Percent of Downloads
United States	40.15%
India	9.52%
China	5.04%
United Kingdom	3.77%
Germany	2.87%
Philippines	2.77%
Spain	2.16%
Canada	2.01%
Iran	1.67%
Malaysia	1.23%
Italy	1.21%
France	1.15%
Australia	1.14%
Egypt	1.10%
Ethiopia	0.98%
Russia	0.89%
Korea	0.87%

an attacker can easily distribute a worldwide attack through a malicious scholarly document.

IV. Sec-Lib: A FRAMEWORK FOR MALWARE DETECTION IN SCHOLARLY DIGITAL LIBRARIES

After our discovery of malicious PDF documents in the CiteSeerX scholarly digital library, we informed CiteSeerX team about the malicious PDF documents, as well as the potential harm that could be caused by not attending to this problem. We also shared our ideas and practical solutions regarding how to address the issue with CiteSeerX. These ideas and solutions presented in this paper were used to develop the Sec-Lib framework. Sec-Lib is a comprehensive detection system aimed at enhancing the security of very large databases, such as scholarly digital libraries. Such databases share characteristics, including: 1) being frequently updated with many new files (e.g. PDF documents) on a daily basis, 2) being affected by external factors, such as the discovery of new vulnerabilities and attack techniques that can be exploited via the hosting program of such files (e.g. Adobe reader), and 3) containing new malicious versions of papers that already exist within the database. Sec-Lib integrates two security layers in order to enhance the detection of malicious PDF documents within scholarly digital libraries. The first layer includes a set of deterministic and rule based detection solutions aimed at the detection of 1) known malware, 2) known vulnerabilities, and 3) incompatibility of PDF documents. The second layer consists of several advanced machine learning based methods for the efficient detection of unknown malicious PDF documents, as well as improving the detection capabilities of the framework on a frequent basis.

Figure 6 demonstrates the Sec-Lib framework and its two layers. The dynamic database consists of the existing papers in the digital library as well as the new papers which are published daily and collected from the Web according to the library's policy. This policy might include collecting

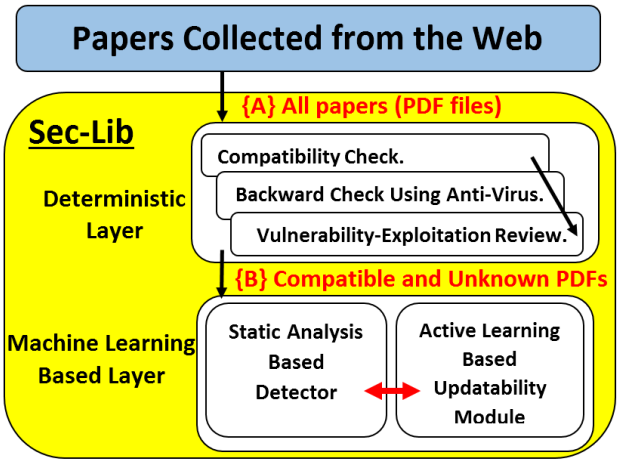

FIGURE 6. Sec-Lib framework.

papers from authors' official websites, open access journals, forums, etc. Then, in step {A} all of these PDF documents are inspected in the deterministic first layer which is particularly aimed at the detection of known PDF malware and its variants, and includes:

- A compatibility check of PDF documents to ensure that they can be properly opened by users before they are made available to library users. (96.5% of malicious PDF documents are incompatible).
- Filtering by the backward check module which filters all of the known malicious PDF documents using an anti-virus signature repository.
- An additional review of each new PDF vulnerability-exploitation identified.

Note that we don't apply the machine learning based layer (second layer) directly on all of the PDF documents for the following reasons. 1) Before implementing the machine learning algorithms, we execute anti-virus software, thereby allowing the anti-virus software to detect as many known malicious files as possible (based on its existing signatures) in the first deterministic layer. It is well-known that machine learning algorithms have had great success in detecting new variants of malware, as well as in assisting in the crafting of signatures for malware. Despite this success, their false positive and false negative rates are typically higher than those of anti-virus software (especially in light of the increased use of adversarial machine learning approaches [55]). Since a system like Sec-Lib is expected to cope with an enormous number of new PDF files on daily basis, Sec-Lib should be efficient and employ a multilayer process; therefore, we use the anti-virus in the first layer and employ the machine learning based detector in the next layer on the remaining files (those files that the anti-virus software couldn't detect as malicious due to its limited detection capabilities).

2) There is no reason to use machine learning solutions on incompatible files, both malicious and benign, as these incompatible files are not openable, and thus, will not be

transferred to the user. These two modules manage to filter out (in advance) all of the files that don't require classification by a machine learning algorithm, while focusing the machine learning-based layer on the remaining more hard to detect files. Since scholarly digital libraries consist of millions of files, a practical solution must be efficient and only focus its efforts on files that cannot be handled by the other deterministic components. Therefore, machine learning solutions are aimed at compatible and unknown PDF documents, both malicious and benign.

Once step {A} is completed, we are left with compatible PDF documents which might contain new unknown PDF malware. These files undergo deep inspection in step {B} (in the second layer) which features a machine learning based approach aimed at new unknown malware detection and includes two advanced components:

- A detector that is based on machine learning techniques and aimed at malicious PDF document detection, which efficiently leverages the statically extracted properties of the structural hierarchies in the PDF documents.
- An active learning-based module and methodology for frequent and efficient update of both the detector and anti-virus tools with new informative PDF documents, especially new malicious ones.

We now elaborate on each of the components in the deterministic first layer:

A. COMPATIBILITY CHECK OF PDF DOCUMENTS

In our previous works [2], [11] we observed that many malicious files are not compatible with the PDF document format specifications according to the Adobe PDF Reference[26] and cannot, in fact, be opened by the PDF reader and viewed by the user. When the user tries to open an incompatible file (malicious or benign), the PDF reader is not able to open it and provides an error message. If it is a malicious PDF document, the malicious operation is executed; if it is a benign file, nothing occurs. However, in both cases the file remains unopened and cannot be viewed by an innocent user. Thus, it is clear that there is no reason to deliver an incompatible file to the user, and this observation should be taken into account in academic digital libraries, which can easily identify such files and mark them as suspicious, or even block them from being published before they are ever opened by an innocent user. Incompatibilities of PDF documents are originated in cases which the crafted PDF documents do not meet the PDF document format specifications according to the Adobe PDF Reference. One example of incompatibility of PDF documents can be observed at the end of the file in the line between "startxref" and "%%EOF" lines. This line should contain a number serving as a reference (offset) to where the last cross reference table section is located in the file. In cases of incompatibility, the number that appears is incorrect. This incompatibility and many others can be easily determine using simple check of the PDF document content in light of the known specifications.

[26]http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf

B. BACKWARD CHECK USING ANTI-VIRUS TOOLS

One of the vulnerabilities we found in academic digital libraries (particularly Google Scholar) relies on the fact that once a new paper is initially uploaded and indexed, it is then assumed to be scanned to verify that it is virus free. However, in cases in which the clean paper file behind the indexed link is later replaced by a malicious version, the file is not rescanned and is now the paper version is available as a malicious file through these libraries. We suggest applying a simple check of the hash function behind each indexed file after it is first uploaded (therefore it is backward check). By comparing the original hash function to a daily hash function of each indexed file, a mismatch between the daily hash of the file and the original version acquired on the initial upload will serve as an indication that the file should be further scanned using Anti-Virus to verify that it is virus free. This idea can also be improved by adding an additional condition, so that the files for which there is a mismatch between the abovementioned hash functions, will only be scanned if the user asks to access this paper, otherwise a scanning operation will not be performed. On the one hand, this improvement will significantly reduce the amount of inspections required, while protecting the users from downloading malicious PDF papers. However, on the other hand, this idea will require an online check when the user wants to download the file, an operation that will increase the time it takes for a user to request and receive a desired paper.

In addition, most benign documents don't usually use JavaScript, OpenAction or the embedding option of the PDF format. This is particularly true for academic papers which don't have any reason to use these option. Thus, a simple approach in the case of scholarly digital libraries is to remove any element that can be used for malicious purposes (including JavaScript code, OpenAction commands or embedded files) from the PDF paper. However, such an approach might lead to a loss of data and or functionality or cause the file to become incompatible; thus, rather than changing the file's content by removing parts of the file, we prefer detecting the malicious content of known malware using AV tools and ML algorithms for unknown malware.

C. NEW PDF MALWARE BACKWARD CHECK

While the vulnerabilities of new PDF documents are identified from time to time by virus experts, the length of the discovery period might be quite long. Meanwhile, the new vulnerability is being used and distributed in additional PDF documents. If a zero-day malware contains such new vulnerabilities, it will probably evade the widely used anti-virus tools for some time. Therefore, as new vulnerabilities are discovered and anti-virus tools are updated accordingly, we suggest a periodic re-check of new vulnerabilities in order to provide a comprehensive and backward review of a process that could

easily be automated for all of the files already exist in the scholarly digital library.

This idea can be improved upon in exactly the same manner we suggested in the previous subsection, by adding the condition of conducting this check only if the user asks to access this paper. The consequences and effects of this will be the same as was mentioned in the previous subsection, both in terms of efficiency and time delay.

We now elaborate on each component of the machine learning based second layer.

D. AN ACTIVE LEARNING BASED DETECTION METHODOLOGY FOR UNKNOWN MALWARE DETECTION

Before we present the active learning based detection methodology, one must first understand the motivation for active learning in very large databases and repositories. In addition, since active learning process is strongly related to the induction of an updated detection model, which is based on informative features, then we will also cover the features on which we were based on when we induced the detection model.

Currently, anti-virus packages are not sufficiently effective at intercepting malicious PDF documents, even in the case of highly prominent PDF threats (Tzermias *et al.* [8]). According to many studies (surveyed by Nissim *et al.* [1]), machine learning methods can effectively distinguish between malicious and benign PDF documents [1]. Yet, when applying machine learning based solutions, it is not enough to create a one-time detection model, since a natural concept drift process exists ([11], [13]), specifically in the malware domain [13]. This is due to the fact that benign files and newly created malware contain new properties and features that haven't been seen by the detection model, as well as existing features with very different values than those the detection model has been trained on (for instance, these new features may result from new elements in the file structure). In addition, the malware domain is very dynamic, since attackers are continually seeking out new ways of attacking, new vulnerabilities that can be exploited, and new targets. These changing parameters affect the file's structure which affect the features extracted from the analyzed file, and thus, significantly reduce the detection capabilities of the detection models which are not updated and remain outdated. Recent studies have successfully applied active learning based solutions for efficient malware detection (e.g. [2], [9], [11], [14], [15], [20], [35]), and concentrated on the updatability process and enhancement of the detection capabilities of the detection model, striving to improve efficiency and speed in these areas. An enhanced and updated detection model will have greater ability to detect future malware. It is therefore essential to update the detection model constantly and frequently with new files (malicious and benign) in order to maintain detection accuracy over time, especially in large repositories such as scholarly digital libraries.

Since scholarly digital libraries contain vast amounts of papers and files to analyze and examine, we suggest conducting static analysis which is fast, lightweight, and analyzes the general descriptive content in the PDF document, rather than dynamically analyzing the JavaScript code as many approaches do. These above mentioned desired elements in static analysis can be achieved by an approach that utilizes the meta-features of the content and structure of the PDF document [3]. The advantage of using meta-features such as structural paths [3] is that they are not affected by code obfuscation. The structural path feature extraction methodology was shown to be a very effective method to discriminate between malicious and benign PDFs, even for new and unknown malicious files created two months after the classification model was created.

Instead of analyzing JavaScript code or any other content, this approach makes use of essential differences in the structural properties of malicious and benign PDF documents. It parses the PDF documents and extracts their structural paths which are the paths in the file's hierarchical object tree that characterize the document's structure. Each structural path is analogous to a set of relations between the objects within the PDF document. For example, the "…/JS" path means that the PDF document contains JavaScript code. The structural paths represent the file's properties and actions, therefore they actually represent the file's genes rather than a current behavior of the file which can be postponed or delayed according to specific conditions.

Note that this structural feature extraction approach has an additional significant advantage in that it does not directly rely on any specific attack element or PDF file component (e.g., it doesn't depend on the presence of JavaScript as was proposed by [54] or embedding malicious executables). This approach identifies the discriminative features based on an analysis of the entire dataset, between the given classes. This approach will suggest using the identified features regardless of the specific attack technique; in this way, the approach will be able, with our AL advancement, to also identify newer attack techniques within the malicious PDF files.

Figure 7 provides a simple example of the conversion of a PDF document into a set of structural paths. The PDF code is treated as a tree of objects. Note that only the paths of the leaves in the structural tree are counted.

When an attacker injects malicious content into the PDF document, the file structure inevitably changes. Thus, this approach can easily discriminate between benign and malicious files. This approach has several advantages. First, it is not affected by code obfuscation, filtering, and other encryption methods used for hiding and concealing malicious code in the PDF document, since it doesn't actually analyze the embedded JavaScript code.

Second, it is robust towards mimicry and reverse mimicry attacks, since our detection framework implements and is based on the structural feature extraction methodology proposed in previous work [21]. In an experiment that included 5,000 malicious and 5,000 benign files, the authors of that

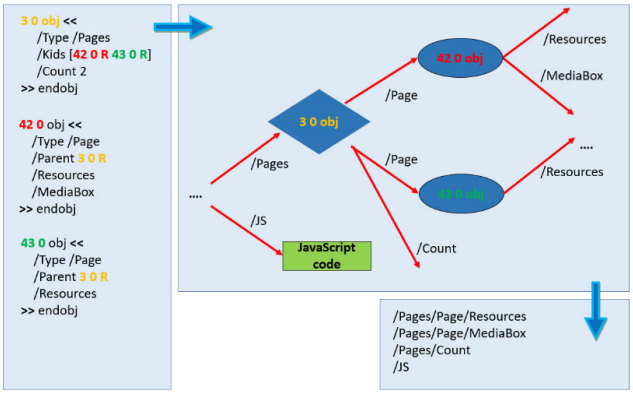

FIGURE 7. Example of the conversion of a PDF document to a set of structural paths.

study showed that this structural feature extraction methodology is resilient to mimicry attacks in 99.975% of the cases they checked. Since our framework utilizes this feature extraction methodology and even improves it with an active learning module that constantly updates the detection capabilities, our framework would be at least as resilient to mimicry attacks as the work which presented this methodology [21]. Finally, it is very fast and lightweight, since the analysis is done statically and does not require any execution of the PDF document. Because of this, analysis is conducted quite quickly at the rate of 28ms for an average file [21]. Using the PdfFileAnalyzer[27] parser we parsed the compatible PDF documents and extracted all of the unique structural paths that were found within our dataset. We applied the information gain feature selection method, and this resulted in the 100 most distinctive and prominent paths. Each of these paths was used as a feature. Each PDF document was represented as a vector of Boolean features so that the presence (1) or absence (0) of a structural path within a PDF document is represented by 1 or 0 respectively.

There are three main approaches for feature selection: *filter methods*, *wrapper methods*, and *embedded methods* [51] Filter methods use a metric to evaluate the correlation of each individual feature to the target class. The *information gain* feature selection method [50] is a type of *filter method* [51] which assigns a higher score to features that contribute more to discrimination between the classes. Information gain is based on entropy calculations. The entropy $E(S)$ characterizes the disorder of an arbitrary set of instances. The higher the entropy the greater the disorder. Equation 1 presents the formula for calculating the entropy of a set of items S (e.g., feature's values), based on C subsets of S (e.g., item classes). The information gain measures the expected reduction of entropy caused by dividing the examples according to attribute A, in which V is the set of possible values of A, as shown in Equation 2.

$$E(S) = \sum_{c \epsilon C} -\frac{|S_c|}{|S|} \cdot \log_2 \frac{|S_c|}{|S|} \tag{1}$$

$$IG(S, A) = E(S) - \sum_{v \epsilon V(A)} \frac{|S_v|}{|S|} \cdot E(S_v) \tag{2}$$

Table 5 shows the 14 most prominent features of the 100 selected features. Their rank and selection criterion are also included in the table. It is interesting to note that none of the 14 prominent features are associated with any of the elements that are known to be utilized in attacks (e.g., embedding files, JavaScript, etc.); such a fact only points to the generality of the features, in terms of their ability to distinguish between malicious and benign PDF files without consideration of the attack's technical aspects. This is a great of importance, since these features will probably also work well when new vulnerabilities or attacks are utilized or invented.

TABLE 5. The 14 most prominent features as they were ranked and selected by the information gain method.

Feature
Trailer\Root\Pages\Kids\.\Kids\.\Kids\.\Resources\Font\F27\FontDescriptor\CharSet
Trailer\Root\Pages\Kids\.\Kids\.\Kids\.\Resources\Font\F12\FontDescriptor\FontFile3
Trailer\Root\Pages\Kids\.\Resources\Font\T18
Trailer\Root\Pages\Kids\.\Resources\Font\T18\Name
Trailer\Root\Pages\Kids\.\Kids\.\Resources\Font\T5\CharProcs\i
Trailer\Root\Names\Dests\Kids\.\Names\.\D\.\Parent\Kids\.\Resources\ExtGState\GS1
Trailer\Root\Pages\Kids\.\Resources\Font\TT2\FontDescriptor\AvgWidth
Trailer\Root\Pages\Kids\.\Resources\Font\C2_2\DescendantFonts\.\FontDescriptor\Font Family
Trailer\Root\StructTreeRoot\K\.\K\.\K\K\.\K\K\.
Trailer\Root\Pages\Kids\.\Kids\.\Kids\.\Resources\XObject\Im88
Trailer\Root\StructTreeRoot\ClassMap\Title\WritingMode
Trailer\Root\Pages\Kids\.\Resources\Font\R190\Widths
Trailer\Root\Pages\Kids\.\Kids\.\Resources\Font\T8\CharProcs\e
Trailer\Root\StructTreeRoot\K\.\K\.\K\K\.\K\.\K\K

Leveraging the abovementioned structural feature extraction approach [3] using machine learning algorithms will induce the detection model. After understading the components of the induced detection model, on the following subsection we deeply explain why and how we suggest applying our active learning framework [2], [11] for enhancing the capabilities of the detection model in light of the mass creation and addition of new PDF documents to the scholarly digital libraries daily. We now describe the active learning-based framework.

E. METHODOLOGY FOR IMPROVING DETECTION OF MALICIOUS PDF DOCUMENTS.

The machine learning based layer of Sec-Lib is depicted in Figure 8. It can be seen that the methodology deals with the process of detecting and acquiring new malicious PDF documents through maintaining the updatability of the anti-virus and detection model. If the file is informative enough or is

[27] http://www.codeproject.com/Articles/450254/PDF-File-Analyzer-With-Csharp-Parsing-Classes-Vers

assessed as likely being malicious, it will be acquired for manual analysis. As Figure 8 shows, the compatible and unknown PDF documents are transported from the deterministic first layer and scrutinized within the second layer of the Sec-Lib framework {1}.

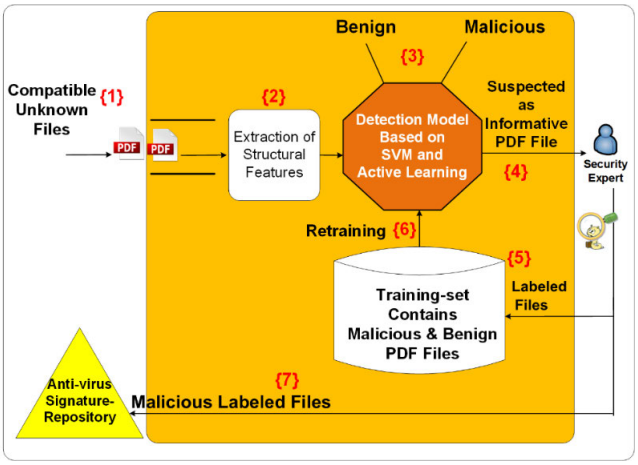

FIGURE 8. The process of detecting and maintaining the updatability of the detection model and anti-virus tool using AL methods.

Then, the prominent and relevant structural paths are extracted from the remaining PDF documents (which are compatible and unknown documents), and each of these paths was used as a feature. Each PDF document is now represented as a vector of Boolean features, so that the presence (1) or absence (0) of a structural path within a PDF document is represented by 1 or 0 respectively. Then the vectors representing the files are introduced to the detection model which is based on SVM and AL.

The detection model scrutinizes the PDF documents and provides two values for each file: a classification decision using the SVM classification algorithm and a distance calculation from the separating hyperplane {3}. A file that the AL method recognizes as informative and has indicated should be acquired is sent to an expert who manually analyzes and labels it {4}.

The goal of the manual analysis process carried out by a human expert for malware detection is to decide whether the informative file that was selected is malicious or benign, and accordingly assign a label (malicious or benign) to each file based on indicators found when the file is analyzed statically or dynamically (during run-time). In order to better understand the file's behavior and the operation the file is actually performing during its run-time, a comprehensive dynamic analysis process must be applied. In this process, the human expert executes the file in an isolated and emulated computational environment referred to as sandbox [39], [40]. Using sandbox, the security expert is able to trace the file's behavior in an environment that is identical to the computational environment that the file will be opened in by the users: the people who will really read a given paper and open the file. Such an environment includes popular operation systems, the relevant PDF reader version, network communication, etc. In this way, the security expert can identify many of the operations taking place during the file's run-time, and therefore malicious and dangerous operations associated with the inspected file can be recognized. Such operations might include 1) unneeded communication with an unknown remote server (can be used to exfiltrate sensitive and private information from the victim's computer, 2) extensive encryption operations on the victim's hard drive (can be a good indication of a ransomware [41] attack in which the attacker encrypts the victim's files and documents, which cannot be accessed until the victim pays the attacker demanding the ransom), 3) extensive usage of CPU and memory (can be a good indication of a cryptojacking attack [42] in which the attacker utilizes and exhausts the computational resources of the victim's computer in order to mine cryptocurrencies[28] on behalf of the attacker). On the other hand, running the informative file in Sandbox, is a secure way to understand the file's behavior while preventing any damage or infection to users' computers and systems. The main shortcoming of manual analysis is that the process is not straightforward, and it requires a human expert as well as computational time and resources. Therefore, our active learning-based framework helps us focus the human expert's efforts and time on the most informative files only, aiming to obtain as much new information for the detection model as possible and keeping it updated given the volume of new malicious files created on a daily basis.

By acquiring these informative PDF documents, we aim to frequently update the anti-virus software by focusing the expert's efforts on labeling PDF documents that are most likely to be malware or on benign PDF documents that are expected to improve the detection model. Note that informative files are defined as those that when added to the training set improve the detection model's predictive capabilities and enrich the anti-virus signature repository. Accordingly, in our context, there are two types of files that may be considered informative. The first type includes files in which the classifier has limited confidence as to their classification (the probability that they are malicious is very close to the probability that they may be benign). Acquiring them as labeled examples will probably improve the model's detection capabilities. In practical terms, these PDF documents will have new structural paths or special combinations of existing structural paths that represent their execution code (inside the binary code of the executable). Therefore these files will probably lie inside the SVM margin and consequently will be acquired by the SVM-Margin strategy that selects informative files, both malicious and benign, that are a short distance from the separating hyperplane.

The second type of informative files includes those that lie deep inside the malicious side of the SVM margin and are a maximal distance from the separating hyperplane. These PDF

[28]https://researchcenter.paloaltonetworks.com/2018/01/unit42-large-scale-monero-cryptocurrency-mining-operation-using-xmrig/

documents will be acquired by the Exploitation method (to be explained later) and are also a maximal distance from the labeled files. These informative files are then added to the training set {5} for updating and retraining the detection model {6}. The files that were labeled as malicious are also added to the anti-virus signature repository in order to enrich and maintain its updatability {7}. Updating the signature repository also requires an update to clients utilizing the anti-virus application. This second layer includes two main phases: training and detection/updating.

1) TRAINING

A detection model is trained over an initial training set that includes both malicious and benign PDF documents. After the model is tested over a stream that consists only of unknown files that were presented to it in the first trial (trials can take place every day / week / month), the initial performance of the detection model is evaluated.

2) DETECTION AND UPDATING

For every unknown PDF document in the stream, the detection model provides a classification, and the AL method provides a rank representing how informative the file is, and the methodology will consider acquiring the files based on this ranking. After being selected and receiving their true labels from the expert, the informative PDF documents are acquired by the training set, and the signature repository is updated as well, just in case the files are malicious. The detection model is retrained over the updated and extended training set which now also includes the acquired examples that are regarded as being very informative. At the end of the trial, the updated model receives a new stream of unknown files on which the updated model is once again tested and from which the updated model again acquires informative files. Note that the goal is to acquire as many malicious PDF documents as possible, since such information will maximally update the anti-virus tool that protects most organizations as well as Web services such as scholarly digital libraries.

We employed the SVM classification algorithm using the radial basis function (RBF) kernel (gamma = 3) in a supervised learning approach. We used the SVM algorithm for the following reasons: 1) SVM has been successfully used to detect worms [14], [26], classify malware into species, and detect zero-day attacks [27], 2) the trained SVM classifier is a black box that is hard for an attacker to understand [26], 3) SVM has proven to be very efficient when combined with AL methods [2], [11], [15], [20], and 4) SVM is known for its ability to handle large numbers of features which makes it suitable for handling the large number of structural paths extracted from the PDF documents [16].

Lastly, based on our preliminary experiments, we found that the SVM classifier with RBF kernel gamma = 3, outperformed all other classifier, kernels, and parameter combinations.

In our experiments we used Lib-SVM implementation [17], in order to classify the PDF files into two classes, benign and malicious, in a binary classification problem. We chose Lib-SVM, which supports the multi-class classification, so that our framework will be able to support future research and additional detection problems which might be associated with multiple classes. We integrated our detection framework within the Weka[29] machine learning environment.

We chose the RBF kernel due to the fact that a complex function is required in order to well distinguish between the malicious and benign classes of PDF documents. Attackers try to evade detection by inserting benign functionalities in the malicious PDF documents, or alternatively, try to hide the malicious elements and functionalities in the malicious file, thus making the malicious files very similar to the benign files. This results in data which is hard to classify and affects the ability of an induced model to discriminate between the benign and malicious PDF documents. The RBF kernel is able to find better separations among the complex data, and thus, is a sophisticated and subtle kernel function suitable for our use.

F. SELECTIVE SAMPLING AND ACTIVE LEARNING METHODS

Since our Sec-Lib framework and AL methodology aims to provide solutions to real problems it must be based on a sophisticated, fast, and selective high-performance sampling method. We compared our proposed AL methods to other strategies, and the four methods considered are briefly described below:

1) RANDOM SELECTION (RANDOM)

While random selection is obviously not an active learning method, it is at the ''lower bound'' of the selection methods discussed. We are unaware of an anti-virus tool that uses an active learning method for maintaining and improving its updatability. Consequently, we expect that all AL methods will perform better than a selection process based on the random acquisition of files.

2) THE SVM-SIMPLE-MARGIN AL METHOD (SVM-MARGIN)

The SVM-Simple-Margin method [18] (referred to as SVM-Margin) is directly related to the SVM (support vector machine) classifier [36], [37]. Using a kernel function, the SVM implicitly projects the training examples into a different (usually a higher dimensional) feature space denoted by F. In this space there is a set of hypotheses that are consistent with the training set, and these hypotheses create a linear separation of the training set. Among the consistent hypotheses, referred to as the version space (VS), the SVM identifies the best hypothesis with the maximum margin. To achieve a situation where the VS contains the most accurate and consistent hypothesis, the SVM-Margin AL method selects examples from the pool of unlabeled examples, reducing the number of hypotheses. This method is based on simple heuristics that depend on the relationship between the VS and

[29] https://www.cs.waikato.ac.nz/~ml/weka/

SVM with the maximum margin. Calculating the VS is complex and impractical where large datasets are concerned, and therefore, the simple heuristic is used. Examples that lie closest to the separating hyperplane (see Figure 9 in which the selected examples from both classes are colored in red and lie inside the margin) are more likely to be informative (may improve the classifier's capabilities) and therefore are acquired and labeled.

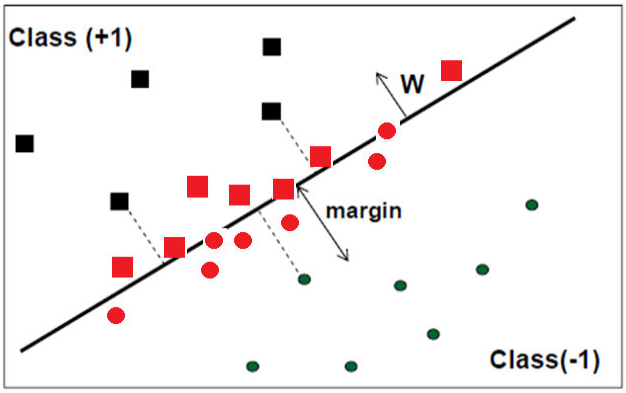

FIGURE 9. The examples (colored in red) that will be selected according to the SVM-Margin AL method's criteria.

This method, in contrast to ours, selects examples according to their distance from the separating hyperplane, only to explore and acquire the informative files without relation to their classified labels, i.e., not specifically focusing on malware instances. The SVM-Margin AL method is very fast and can be applied to real problems; yet, as its authors indicate [18], this agility is achieved, because it is based on a rough approximation and relies on assumptions that the VS is fairly symmetric and that the hyperplane's Normal (W) is centrally placed, assumptions that have been shown to fail significantly [28]. The method may query instances in which the hyperplane does not intersect the *VS*, and therefore, may not be informative. The SVM-Margin method for detecting instances of PC malware was used by Moskovitch *et al.* [29] whose preliminary results found that the method also assisted in updating the detection model but not the anti-virus application itself; however, in this study the method was only used for a one day trial. We compared its performance to our proposed AL methods for a longer period, in set of experiments which consider a daily process of detection of PDF documents and their acquisition, which reflects what happens in reality. This serves as our baseline AL method, and we expect our method to improve the new malicious PDF detection and acquisition seen in SVM-Margin.

3) EXPLOITATION: OUR PROPOSED ACTIVE LEARNING METHOD

Our method, "Exploitation" [9], is based on SVM classifier principles and is oriented towards selecting examples most likely to be malicious that lie furthest from the separating hyperplane. Thus, our method supports the goal of boosting the signature repository of the anti-virus software by acquiring as much new malware as possible. For every file X that is suspected of being malicious, Exploitation rates the distance from the separating hyperplane using Equation 3 based on the Normal of the separating hyperplane of the SVM classifier that serves as the detection model. In fact, Equation 3 calculates the distance between the vector that represents the inspected file (X) to the separating hyperplane of the detection model. The separating hyperplane of the SVM is represented by W (Equation 4). W is the Normal of the separating hyperplane and is actually a linear combination of the most important examples Xi (supporting vectors) and their labels Yi (e.g., malicious or benign), multiplied by Lagrange multipliers (α) and the kernel function K that assists in achieving linear separation in higher dimensions. Accordingly, the distance in Equation 1 is simply calculated between example X and the Normal (W).

$$Dist(X) = \left(\sum_{1}^{n} \alpha_i y_i K(x_i x)\right) \tag{3}$$

$$w = \sum_{1}^{n} \alpha_i y_i \Phi(x_i) \tag{4}$$

In Figure 10, the files that were acquired (marked with a green circle) are the files classified as malicious and have the maximum distance from the separating hyperplane. Acquiring several new malicious files that are very similar to and belong to the same virus family is considered a waste of manual analysis resources, since these files will probably be detected by the same signature. Thus, acquiring one representative file for this set of new malicious files will serve the goal of efficiently updating the signature repository. In order to enhance the signature repository as much as possible, we also check the similarity between the selected files using the kernel farthest first (KFF) method suggested by Baram *et al.* [30]

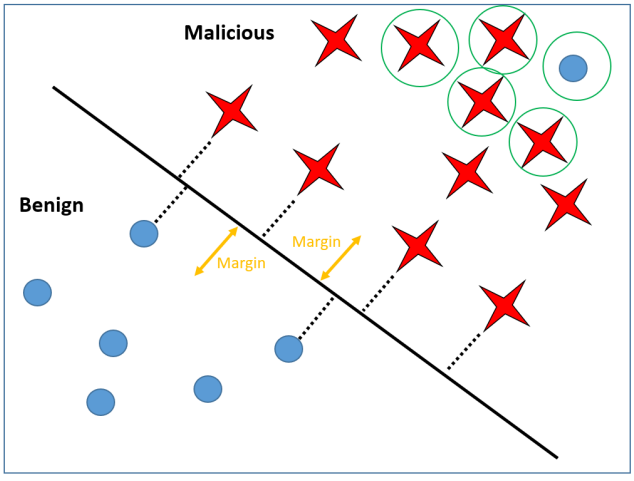

FIGURE 10. The criteria by which Exploitation acquires new unknown malicious PDF documents. These files lie the farthest from the hyperplane and are regarded as representative files.

which enables us to avoid acquiring examples that are quite similar. Consequently, only the representative files that are most likely malicious are selected. In cases in which the representative file is detected as malware as a result of the manual analysis, all variants that were not acquired will be detected the moment the anti-virus is updated. In cases in which these files are not actually variants of the same malware, they will be acquired the following day (after the detection model has been updated), as long as they are still likely to be malware.

In Figure 10 it can be observed that there are sets of relatively similar files (based on their distance in the kernel space), however, only the representative files that are most likely to be malware are acquired. The SVM classifier defines the class margins using a small set of supporting vectors (i.e., PDF documents). While the usual goal is to improve classification by uncovering (labeling) files from the margin area, our main goal is to acquire malware in order to update the anti-virus. Contrary to SVM-Margin which explores examples that lie inside the SVM margin, Exploitation explores the malicious side to discover new and unknown malicious files that are essential for the frequent update of the anti-virus signature repository, a process which occasionally also results in the discovery of benign files (files which will likely become support vectors and update the classifier). Figure 5 also presents an example of a file lying far inside the malicious side that was found to be benign. The distance calculation required for each instance in this method is fast and equal to the time it takes to classify an instance in a SVM classifier, thus it is applicable for products working in real-time.

4) COMBINATION: A COMBINED ACTIVE LEARNING METHOD

The ''Combination'' method [2] lies between the SVM-Margin and Exploitation. On the one hand, the combination method begins by acquiring examples based on SVM-Margin criteria in order to acquire the most informative files (acquiring both malicious and benign files), an exploration phase which is important in order to enable the detection model to discriminate between malicious and benign PDF documents. On the other hand, the combination method then tries to maximally update the signature repository in an exploitation phase, drawing on the Exploitation method. This means that in the early acquisition period, during the first part of the day, SVM-Margin is more dominant compared to Exploitation. As the day progresses, Exploitation becomes predominant. However, Combination is also being applied in the course of the 10-day experiment, and over a period of days, Combination will perform more Exploitation than SVM-Margin. This means that on the i^{th} day there is more Exploitation than in the $(i\text{-}1)^{th}$ day. We defined and tracked several configurations over the course of several days. Regarding the relation between SVM-Margin and Exploitation, we found that a balanced division performs better than other divisions (i.e., during the first half of the study, the method will acquire more files using SVM-Margin, while during the second half of the study, Exploitation takes the leading role in the acquisition of files). In short, this method tries to take the best from both of the previous methods.

V. EVALUATION

A. EXPERIMENTAL DATASET CREATION

In order to Evaluate Sec-Lib's machine learning based solutions, we created an experimental dataset of a 259,635 compatible malicious and benign PDF documents based on the published papers existing in the CiteSeerX scholarly digital library. We randomly selected a set of 225,591 benign PDF documents from all of the papers in the library. The selection process provided a randomly generated number (between one and zero to each file), and we simply selected only files that had a random number below a specific threshold. An additional set of 34,044 malicious PDF documents was collected, both from the malicious files we found in CiteSeerX during our first scan, and from various other sources of malicious files such as VirusTotal's repository and the Srndic and Laskov academic repository [3]. It is important to understand that according to our discovery when scanning CiteSeerX with VirusTotal, the percentage of malicious PDF documents found was around 0.3%, however this percentage can easily and significantly increase, since attackers can use the approaches we've presented to more extensively contaminate scholarly digital libraries and use them as a malware distribution platform. In addition, it is reasonable that additional approaches will be found in order to exploit digital libraries. In our experiment we wanted to create a reasonable and potentially high risk situation in which the percentage of malicious PDF documents in the scholarly digital library is higher than the percentage we have found in our scan. Although we used a higher percentage than that found in our scan, we still used a relatively low percentage of malicious PDF documents (13.2%), particularly compared to the average percentage of malicious PDF documents reported in 10 recent academic studies (38%, as we stated in a previous study [11]). The benign files were reported to be virus free by VirusTotal. The dataset was intentionally designed to be imbalanced in order to reflect the reality in the malware detection domain. It is not realistic or correct to have a balanced dataset of malicious and benign files, since in real life 50% of the content is not malicious; the percentage of malicious content is far less than that of benign content, and this is one of the challenges in the malware detection domain that we must cope with. Therefore, the test set must be based on reality and contain an imbalanced combination of malicious and benign files. In our previous study [43], we conducted a comprehensive analysis of the optimal combination of malicious and benign files, in the training set and test set. Note that the malicious files percentages within the test set must be based on reality. Our finding was that the optimal results and the best detection model is induced when the same file combination appears in both the training and test sets. Since the test set is

defined by reality, which was shown to have a low percentage of malicious files, we designed the training set similarly. In addition, this combination does not affect the classifier in terms of bias and correctness as long as the correct evaluation measurements are used. Of course, the standard accuracy measurement is not suitable for evaluating classifiers' performance which has been tested on an imbalanced dataset; therefore, more suitable evaluation measurements should be used, such as the TPR and FPR which were used in our experiments. Moreover, many previous malware detection studies addressing a variety of different malware detection challenges (e.g., malicious executables, Android malware, malicious MS Office files, computer worms, and ransomware detection) have used imbalanced datasets (with a small percentage of malicious files) [3], [14], [15], [41], [43]–[49] that reflect reality and support the correctness of our approach and the use of an imbalanced dataset. The goal of our classifier is to detect malicious files, and we accomplish this by using a binary classifier that classifies the file as malicious or benign. Thus, the detection of a malicious file is demonstrated by classifying a malicious file as malicious, and this detection rate is measured by the TPR metric; an error in classification in which a benign file is classified as malicious is measured by the FPR metric. Since the goal of the paper is to detect malicious files, we present the results of only the TPR and FPR. Presenting the results of the TNR and FNR does not add relevant knowledge about the ability of our framework to achieve the main goals which are detecting malicious files and acquiring informative files for further improvement of the detection model over time.

The malicious set contains several malware families such as viruses, Trojans, and backdoor attacks. Based on our preliminary experiments, we used only 100 unique structural paths (features) selected by the known information gain feature selection method. In our preliminary experiments we compared the information gain and Fisher score [31] selection methods and information gain outperformed the latter.

As part of our preliminary experiments, we also evaluated the performance of the different classifiers as a function of the number of features used, checking various amounts of features (50, 100, 150, 200, 300 – through 2000, at increments of 100 features). We note that the detection rate improves until 97 features are used; then it remains the same until 200 features are used. After that, the rate slowly decreases. We decided to select the first 100 features as they provided the maximal detection rate. The use of 200 features is associated with more computational resources (without improved performance), especially in our case in which the classifier is frequently updated using the AL methodology Sec-Lib.

Each file was represented as a vector of 100 binary features (value 1 represents the presence of a structural path in the PDF document, while 0 represents its absence).

B. EXPERIMENTAL DESIGN

The objective of our main experiment was to evaluate and compare the performance of our new AL methods to the existing selection methods, SVM-Margin and Random, on two tasks:

- Acquiring as many new unknown malicious PDF documents in scholarly digital libraries on a daily basis, in order to efficiently enrich the signature repository of the widely used anti-virus tools.
- Updating the predictive capabilities of the detection model that serves as the knowledge store of AL methods and improving its ability to efficiently detect malicious PDF documents, as well as identify the most informative new malicious PDF documents.

We evaluated the second layer of the framework (the machine learning based layer) in a simulation of 10 trials (a trial can be either a day, week, or month, depending on the needs of the digital library). In our experiments we preferred that the trials be days, for a total period of 10 days, in order to demonstrate, as much as possible, the importance and contribution of such a frequent updating process. We evaluated several acquisition methods including AL methods and random selection, and compared the performance of the detection model that was updated separately by each of the selection methods. In our acquisition experiments we used 259,635 compatible PDF documents (225,591 benign, 34,044 malicious) in our repository and created 10 randomly selected datasets with each dataset containing 10 subsets of 25,900 files representing each day's stream of new PDF documents. The 635 remaining files were used as the initial training set to induce the initial model. Note that each day's stream contained 25,900 PDF documents.

Note that the combination of malicious and benign files in each of the 10 subsets of 25,900 files and the initial training set of 635 files matches the combination of malicious and benign files in the entire dataset, since they were selected randomly from the entire dataset; in each case, $\sim$13% of the files are malicious (i.e., 34,044 out of 259,635).

First, we induced the initial model by training it over the 635 known PDF documents. We then tested it on the first day's stream.

The reason and motivation for using a relatively small initial dataset was to demonstrate the efficiency of our machine learning and active learning-based detection framework in the process of improving its detection rates over time (days/trials). One should note that in reality, having a good set of labeled files (labeled malicious or benign) requires the manual inspection of a human expert, a task which is associated with additional time and resources (as was explained earlier). Therefore, the motivation is to reduce the burden of labeling files as much as possible and to use as small an initial training set as possible. By inducing the initial detection model from a small randomly selected initial training set of 635 files that included 13.11% malicious files (83 malicious files and 570 benign files), our initial detection model had a relatively high detection rate (74.5% TPR), which shows that although we used a small training set, the structural features we extracted from the files enable us to achieve encouraging detection capabilities. Furthermore,

by performing the active learning process over a period of 10 days, we demonstrated that the initial detection model can be significantly improved by sustaining it with a small amount of well-selected informative files (malicious and benign). By doing so, we showed how our framework can contribute to reducing the amount of resources dedicated to file labeling, from the early stage of creating the initial model (based on small initial training set) and continuing throughout the experiment with its constant improvement over time.

Next, from this same stream of unknown PDF documents (files that the detection model hasn't seen or trained before), the selective sampling method selected the most informative PDF documents according to that method's criteria. One should note that when an informative file is selected for acquisition, the only one who labels it is the human expert and not any other autonomous system. The labeling process is critical for an accurate dataset, which is the basis of an accurate detection model; thus, the labeling process must be performed carefully and by a human expert (or at least under human expert supervision). Indeed, as was explained earlier, the human expert uses a variety of tools and techniques (e.g., dynamic analysis, Sandbox, etc.), which help him/her decide how to label the file. Regardless of the identified label of the file, this new informative file is acquired and added to the training set, and the new updated detection model is induced based on the extended and updated training set.

The labeled files were later acquired by the training set which was enriched with an additional K new informative files. When a file was found to be malicious, it was immediately used to update the signature repository of the anti-virus, and an update was also distributed to clients of the anti-virus software; because anti-virus software is the simplest and most widely used malware detection solution, and nearly every personal and organizational computer is protected by anti-virus software, almost everyone can be considered a client. However, anti-virus software is reliant on the frequent updates distributed by vendors which enable the anti-virus software to detect new malicious files that are created. Given this reality, our framework is of great importance and will contribute to accelerating the process of updating anti-virus software and the protection of anti-virus clients.

The process was repeated over the next nine days. The performance of the detection model was averaged for 10 runs over the 10 different datasets that were created. Each selective sampling method was checked separately on 10 different acts of file acquisition (each consisting of a different amount of PDF documents). This means that for each act of acquisition, the methods were restricted to acquiring a number of files equal to the amounts that followed, denoted as K: 10 files, 100 files, 500, 1000, 2000, 3000, 4000, 5000, 6000, and 7000).

The steps of the experiment are as follows:

1. Inducing the initial detection model from the initial available training set; i.e., training set available up to day d (the initial training set includes 635 PDF documents).

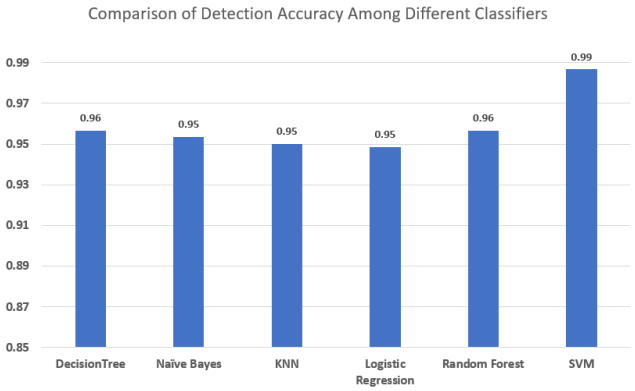

FIGURE 11. The detection accuracy achieved by each of the classifiers; SVM outperformed the other classifiers.

2. Evaluating the detection model on the stream of day $(d + 1)$ to measure its initial performance.
3. Introduction of the stream of day $(d+1)$ to the selective sampling method, which chooses the X most informative files according to its criteria and sends them to the expert for manual analysis and labeling.
4. Acquiring the informative files and adding them to the training set, as well as using their extracted signature to update the anti-virus signature repository.
5. Inducing a new and updated detection model from the updated training set (which contained the previously acquired files combined with the newly acquired files) and applying the updated model on the next day's stream (d+2).

This process repeats itself on our dataset from the first day until the tenth day.

VI. RESULTS

Before we delve into the main experiment's results, we present our results of a preliminary experiment comparing the detection accuracy of various classifiers; in this case, the classifiers' performance was evaluated utilizing the entire collection of compatible PDF file that included 293,679 files (259,635 benign and 34,044 malicious). The classifiers were evaluated throughout the standard 10-fold cross-validation process (training on nine folds representing 90% of the data and testing on the remaining fold representing 10% of the data; repeating this process 10 times, each time with other training and test folds; and finally, averaging the 10 different repetitions). The results are presented in a new figure (Figure 11). An addition preliminary subexperiment was performed, in which we compared the three different kernels (linear, polynomial, and RBF) and different tuning parameters of SVM, and we found that SVM with RBF kernel gamma = 3 slightly outperformed the others.

Note that in previous work on PDF detection based on meta and structural features [53], Smutz and Stavrou found that the Random Forest classifier outperformed all other classifiers, including SVM; their evaluation was based on

100,000 benign and 5,000 malicious PDF documents. In contrast, our preliminary experiment showed that the SVM classifier outperformed all other classifiers, including the Random Forest, in detection accuracy. Our results are based on a larger (almost three times larger) and more up-to-date PDF collection.

We rigorously evaluated the efficiency and effectiveness of our AL methodology, comparing four selective sampling methods: 1) a well-known existing AL method, SVM-Simple-Margin (SVM-Margin) based on [18]; our proposed methods 2) Exploitation, and 3) Combination; and 4) random selection (Random) as a "lower bound." Each method was checked for all 10 acquisition amounts, in which the results were the mean of 10 different folds. In order to focus the readers on the most interesting results we depicted the results of the most representative acquisition amount of 500 PDF documents which is a reasonable number of files that can be analyzed and inspected on a daily basis by the security experts of organizations like a digital library.

We now present the results of the core measure in this study, the number of new unknown malicious files that were discovered and finally acquired into the training set and signature repository of the anti-virus software. As explained above, each day the AL methodology deals with 25,900 new PDF documents, consisting of about 3400 new unknown malicious PDF documents. Statistically, the more files that are selected daily, the more malicious files that will be acquired daily. By using AL methods, we tried to improve the number of malicious files acquired by means of existing solutions. More specifically, using our methods (Exploitation and Combination) we also sought to improve the number of files acquired by SVM-Margin.

Figure 12 presents the number of malicious PDF documents obtained by acquiring the 500 files daily, by each of the four methods during the course of the 10 day experiment. Exploitation and Combination outperformed the other selection methods. Exploitation was the only method that had perfect acquisition of all malicious PDF documents from the first day, while Combination had a decrease in the second day and then perfect acquisition as well. Both of our AL methods outperformed all of the other methods, both SVM-Margin and Random.

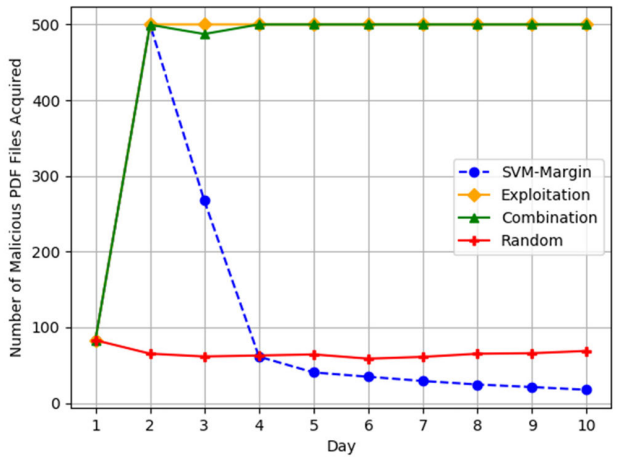

FIGURE 12. The number of malicious PDF documents acquired by the AL methodology for different methods with acquisition of 500 files daily.

On the first day, the number of new malicious PDF documents is 83, since the initial detection model was trained on an initial set of 653 labeled PDF documents that consisted of 83 malware samples. We decided on 653 files from which the initial detection model would be induced in order to have a stable detection model with sufficient detection performance from the start (74.5% TPR on the first day) that can still be improved through our active learning-based methodology.

On the tenth day, using Combination and Exploitation, 100% of the acquired files were malicious; using SVM-Margin, only 3.5% of the acquired files were malicious (17 files out of 500 which is even less than Random). This presents a significant improvement of almost 97% in unknown malware acquisition. Note that on the tenth day, using Random, only 13.8% of the acquired PDF documents were malicious (69 files out of 500). This is far less than the malware acquisition rates achieved by both Combination and Exploitation. The trend is very clear from the second day: each day, Combination and Exploitation acquired the maximal number of malicious files– a finding that demonstrates the impact of updating and preserving the capabilities of the detection model in identifying new malware samples and enriching the signature repository of the anti-virus. Moreover, the acquired malware samples are expected to also have higher quality in terms of their contribution to the detection model and the signature repository, since they are different.

As far as we could tell, the random selection trend was constant - there was no improvement in acquisition capabilities over the 10 days. While the SVM-Margin AL method showed a decrease in the number of malware samples acquired from the fifth day. The SVM-Margin acquires examples about which the detection model is less confident. Consequently, they are considered to be more informative but not necessarily malicious. As was explained previously, SVM-Margin selects new informative PDF documents inside the margin of the SVM. Over time and with the improvement of the detection model towards more malicious files, it seems that the malicious files are less informative (due to the fact that malware writers frequently try to use upgraded variants of previous malware samples). Since these new malware samples might not lie inside the margin, SVM-Margin may actually be acquiring informative benign, rather than malicious, files. However, our methods, Combination and Exploitation, are more oriented toward acquiring the most informative files and the files most likely to be malicious by obtaining informative PDF documents from the malicious side of the SVM margin. As a result, an increasing number of new malware samples are acquired; in addition, if an acquired benign file lies deep within the malicious side, it is still informative and can be used for learning purposes and to improve the next day's detection capabilities.

So far we have shown that our AL methods outperformed the SVM-Margin AL method and the Random method and improved the capabilities for acquiring new PDF malware samples and enriching the signature repository of the anti-virus software. We will now also see the detection capabilities of induced detection models. Figure 13 presents the TPR levels and their trends in the 10 day course of study. SVM-Margin outperformed other selection methods in the TPR measure, while our AL method, Combination, came close to SVM-Margin (SVM-Margin achieved about 2% better TPR rates than Combination). However, a single factor ANOVA statistical test on the TPR for SVM-Margin and Combination active learning methods, resulted in a statistically insignificant difference ($p = 0.67$), suggesting that both active learning methods, Combination and SVM-Margin, perform similarly.

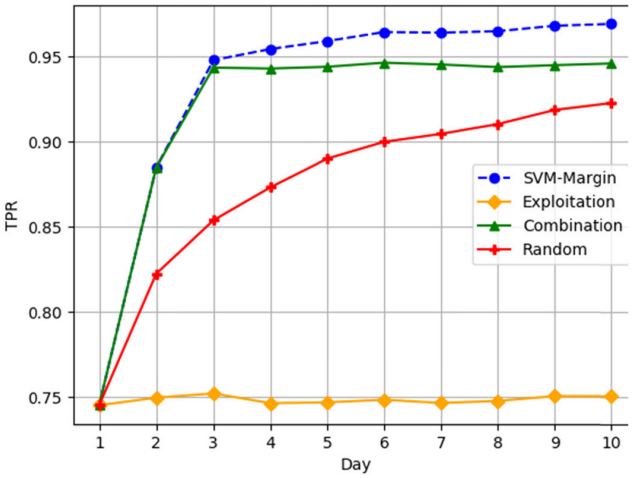

FIGURE 13. The AL methodology's TPR over the 10 days for different methods with the acquisition of 500 PDF documents daily.

In addition, the performance of the detection model improves as more files are acquired daily, so that in the tenth day of the experiment, the results indicate that by only acquiring a small and well-selected set of informative files (1.9% of the stream), the detection model can achieve TPR levels (96.9% with SVM-Margin and 94.7% with Combination) that are quite close to those achieved by acquiring the whole stream (98.1%) which requires labeling of almost the entire dataset. Using Sec-Lib and AL methods, we achieved almost the same TPR levels, while using less than 2% of the total 259,635 PDF documents in our experimental dataset after 10 days of efficient acquisition (our dataset was built so it will be representative of the entire scholarly digital library).

Note that Exploitation didn't manage to update the detection model well, and this may be due to the fact that the malicious files don't have enough new information to enhance the method's detection capabilities, and indeed a considerable number of benign files also must be acquired as with the Combination method. The largest gap between AL methods (SVM-Margin and Combination) and Random can be observed on the third day and as it was demonstrated by a gap of almost 10% of detection rates, which means that AL methods better identified informative PDF documents to update the detection model and justify the process, while Random doesn't even manage to achieve these rates during the 10 days, after acquiring a total of 5,000 PDF documents (500 on each day).

The FPR rates were very low (almost 0%) and were quite similar among all the selection methods.

Finally, to support the effectiveness of our system, we compared it to widely used anti-virus tools in the task of detecting malicious PDF files from our collection. For a fair comparison, and in order to emphasize the efficacy of our proposed solution, we didn't use the SVM detector from our preliminary experiment that was trained on 90% of the data and achieved a 99% detection rate (Figure 13). Instead we used the SVM detector that was created through the process of active learning, which better reflects the reality. This SVM detector was created and updated along 10th days, acquiring amount of new PDF files that can still be inspected and labeled by human experts. Note that after 10 days the induced classifier was trained on just a small portion of the data (less than 2% or 5,635 files – 635 files in the initial set + 500 files daily over a period of 10 days = 5,635). As can be seen in Figure 14 where the detection rate of Sec-Lib is compared to the detection rate of the 20 leading anti-virus tools, Sec-Lib outperformed all of the others. Sec-Lib's 96.9% detection rate was achieved using an efficient active learning process in which, due to the intelligent selection of the most informative

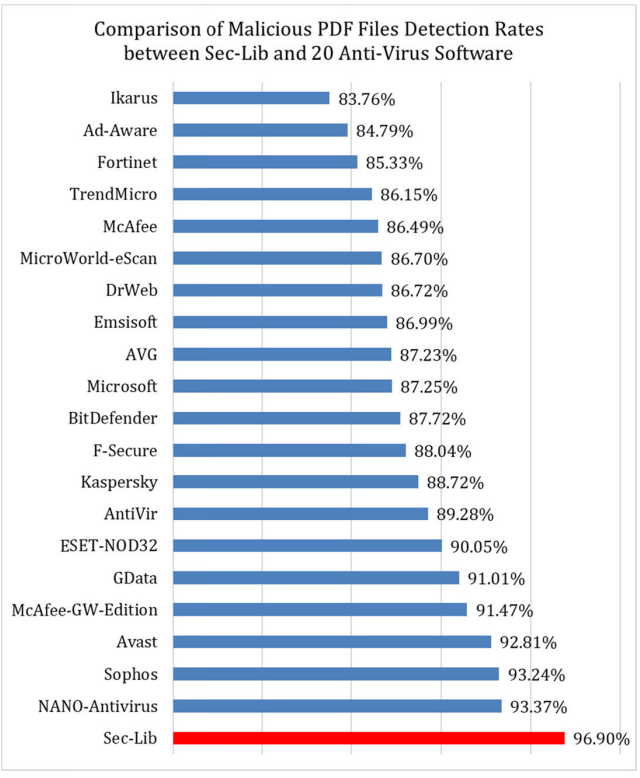

FIGURE 14. Sec-Lib detection rate compared to the 20 best anti-virus tools; Sec-Lib outperformed all of them.

PDF files each day, less than 2% of the data was sent to a human expert for labeling. This performance demonstrates Sec-Lib's ability to update and maintain the model's detection rate and to reduce labeling efforts, costs, and time.

Around one year after we shared our discoveries and detection methods with CiteSeerX, we conducted an additional scan in order to learn whether the malicious files were removed from CiteSeerX, and whether our security solutions had any long-term implications on the percentage of malicious files in CiteSeerX. During our recent scan of CiteSeerX, we managed to scan 3,874,336 PDF documents using VirusTotal. The scanning resulted in the identification of just 1,145 malicious files (representing 0.02% of the total files) compared to the rate of 0.3% that was observed in the first scan. This comparison demonstrates a significant improvement and a reduction of 15 times less malware a year after discovered in the second scan, after implementing the Sec-Lib framework in the scholarly digital library. Note that the malicious PDF documents found in the second scan do not overlap with those found on the first scan – a finding that indicates that the malicious files found on the first scan 2015 were completely removed from the CiteSeerX library.

VII. LIMITATIONS AND POSSIBLE ATTACKS ON Sec-Lib

Adding and deleting attacks, two attacks targeting active learning methods in which attackers contaminate unlabeled data prior to its sampling by the active learner module, were presented by Zhao *et al.* [56]. Their evaluation of the attacks using an intrusion detection dataset demonstrated that these attacks affect AL methods' performance and result in a dramatic drop in detection accuracy, which was shown to decrease by up to 34%. In the context of an attack on Sec-Lib, their attack model could be used to create evasive and malicious PDF files that can avoid detection, simply by taking advantage of the manner in which Sec-Lib acquires informative files. For example, the attacker could influence the AL process and bias the classifiers by utilizing malware samples with new structural paths which the classifier would acquire based on their novelty. Having accomplished this, the attacker can create an evasive PDF file which is based on the malware samples but does not include the specific structural paths, thereby creating an evasive PDF file.

Sec-Lib is resilient to such attacks for two reasons. The first reason is due to the fact that Sec-Lib's AL process is not based on a specific digital library on the Internet, but is rather sustained by many libraries containing many files. Given this, such attacks would need to flood significant portions of multiple digital libraries in order to poison the Sec-Lib framework and bias the classifier. Such flooding by an attacker is both infeasible and time-consuming, allowing anti-virus services the time to distribute a patch against it (note that anti-virus software is part of Sec-Lib's deterministic layer). The second reason stems from the fact that our framework tries to select the most informative PDF files and attempts to enlarge the signature repository that way, as opposed to choosing files that are similar to previously acquired files. In the case of the example described above, Sec-Lib's AL methods would only acquire a few representative structural paths (as opposed to a full set of malicious PDF files with similar structural paths). Thus, the framework would be resilient to such attacks, and its detection capabilities would remain unaffected.

However, as was shown by Maiorca *et al.* [55], the evolving adversarial learning trend has become popular among attackers, particularly those employing PDF malware, and more methods aimed at confusing and evading ML based detectors are being proposed. Thus, we presume that the arms race between attackers attempting to avoid detection and entities trying to defend against such attacks will continue for years.

VIII. DISCUSSION AND CONCLUSION

This study revealed the phenomenon of contamination of scholarly digital libraries by malicious PDF documents and showed how these libraries can easily be used to launch and distribute targeted cyber-attacks aimed at a specific group of researchers, universities, institutions, and countries. As far as we know, there are no reliable reports of the accurate percentage of malicious PDF documents on the Web, and therefore, we cannot determine whether scholarly digital libraries are more or less contaminated than the Web itself. Our study does point to the ease with which large public databases can become contaminated and the role they can play in the spread of malware.

In this study, we evaluated more than two million scholarly papers in the CiteSeerX library in our first scan of the database and found it to be contaminated with a surprisingly large number (0.3%-2%) of malicious PDF documents belonging to a variety of different virus families, 72% of which exploit vulnerabilities in PDF readers (Figure 3). These malicious documents were uploaded from 46 countries, covering most continents. US universities were found to be the origin of over 55% of the malicious papers in CiteSeerX. More than 41% of these malicious scholarly papers were downloaded in the US during the last five years. On average, each malicious paper was downloaded 167 times over a period of five years by researchers worldwide. As we have shown, vulnerabilities exist in scholarly digital libraries, and an attacker needs only to place a malicious version of a popular paper on an attractive topic (e.g., cyber security) in a scholarly digital library to utilize the damage coefficient we found (167 downloads in five years) and spread damage around the world or launch a targeted attack aimed at of group of victims interested in the topics the paper presents. In fact, we show how existing scholarly digital libraries can easily be leveraged as a distribution platform for targeted as well as global attacks.

As a practical means of securing very large databases such as scholarly digital libraries, we designed and developed Sec-Lib which is a comprehensive detection framework aimed at enhancing the detection of malicious PDF documents. Sec-Lib integrates two security layers. The first layer is aimed at the detection of known malware and includes a set of deterministic and rule based detection solutions.

The second layer, aimed at unknown malware detection, consists of several advanced machine learning based methods, such as an SVM classifier trained on structural features of PDF documents, as well as active learning methods for enhancing the detection capabilities over time. Results showed that Sec-Lib efficiently detected unknown malware with high TPR levels (96.9% with SVM-Margin and 94.7% with Combination) and almost no false positives, using only 2% of the dataset for training, thus reducing the manual inspection efforts of security experts by 98%. These results have potential economic implications and demonstrate the efficiency of the Sec-Lib framework in maintaining and improving the updatability of the detection model and ultimately, the anti-virus tool. These of high detection rates achieved through a minimal acquisition of PDF documents for inspection, demonstrate the benefits obtained by performing this process on a daily basis. After considering the ideas and methods presented in the Sec-Lib framework within CiteSeerX scholarly digital library, we observed a reduction of 15 times less PDF malware in this library, a finding that demonstrates our frameworks ability to provide meaningful improvement in the security of very large databases such as scholarly digital libraries.

IX. FUTURE WORK

In future work, we suggest evaluating the malicious PDF presence in additional digital libraries such as MAS, Web of Science, and PubMed, as well as investigating them for vulnerabilities. We also suggest investigating the rate of contamination of digital libraries within the Darknet, such as Libgen, Sci-Hub, and Booksc.

REFERENCES

[1] N. Nissim, A. Cohen, C. Glezer, and Y. Elovici, "Detection of malicious PDF files and directions for enhancements: A state-of-the art survey," *Comput. Secur.*, vol. 48, pp. 246–266, Feb. 2015. doi: 10.1016/j.cose.2014.10.014.

[2] N. Nissim, A. Cohen, R. Moskovitch, O. Bar-Ad, M. Edry, A. Shabtai, and Y. Elovici, "ALPD: Active learning framework for enhancing the detection of malicious PDF files," in *Proc. JISIC*, Sep. 2014, pp. 91–98.

[3] N. Šrndić and P. Laskov, "Detection of malicious PDF files based on hierarchical document structure," presented at the 20th Annu. Netw. Distrib. Syst. Secur. Symp., 2013.

[4] V. Hamon, "Malicious URI resolving in PDF documents," *J. Comput. Virol. Hacking Techn.*, vol. 9, no. 2, pp. 65–76, 2013.

[5] M. Khabsa and C. L. Giles, "The number of scholarly documents on the public Web," *PLoS ONE*, vol. 9, no. 5, 2014, Art. no. e93949.

[6] Y. Gargouri, C. Hajjem, V. Larivière, Y. Gingras, L. Carr, T. Brody, and S. Harnad, "Self-selected or mandated, open access increases citation impact for higher quality research," *PLoS ONE*, vol. 5, no. 10, Oct. 2010, Art. no. e13636.

[7] S. Lawrence, "Free online availability substantially increases a paper's impact," *Nature*, vol. 411, p. 521, May 2001.

[8] Z. Tzermias, G. Sykiotakis, M. Polychronakis, and E. P. Markatos, "Combining static and dynamic analysis for the detection of malicious documents," presented at the 4th Eur. Workshop Syst. Secur. 2011.

[9] N. Nissim, R. Moskovitch, L. Rokach, and Y. Elovici, "Novel active learning methods for enhanced PC malware detection in windows OS," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5843–5857, Mar. 2014. doi: 10.1016/j.eswa.2014.02.053.

[10] A. Apvrille and T. Strazzere, "Reducing the window of opportunity for Android malware Gotta catch 'em all," *J. Comput. Virol.*, vol. 8, nos. 1–2, pp. 61–71, 2012.

[11] N. Nissim, A. Cohen, R. Moskovitch, A. Shabtai, M. Edri, O. Bar-Ad, and Y. Elovici, "Keeping pace with the creation of new malicious PDF files using an active-learning based detection framework," *Secur. Inform.*, vol. 5, p. 1, Dec. 2016.

[12] I. Žliobaitė, "Learning under concept drift: An overview," 2010, *arXiv:1010.4784*. [Online]. Available: https://arxiv.org/abs/1010.4784

[13] A. Singh, A. Walenstein, and A. Lakhotia, "Tracking concept drift in malware families," in *Proc. 5th ACM Workshop Secur. Artif. Intell.*, Oct. 2012, pp. 81–92.

[14] N. Nissim, R. Moskovitch, L. Rokach, and Y. Elovici, "Detecting unknown computer worm activity via support vector machines and active learning," *Pattern Anal. Appl.*, vol. 15, no. 4, pp. 459–475, 2012.

[15] N. Nissim, R. Moskovitch, O. Barad, L. Rokach, and Y. Elovici, "ALDROID: Efficient update of Android anti-virus software using designated active learning methods," *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 795–833, Dec. 2016.

[16] T. Joachims, "Making large-scale SVM learning practical," SFB 475: Komplexitätsreduktion Multivariaten Datenstrukturen, Univ. Dortmund, Dortmund, Germany, Tech. Rep. 1998, 28, 1998.

[17] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 7.

[18] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, Nov. 2001.

[19] N. Nissim, A. Cohen, J. Wu, A. Lanzi, L. Rokach, Y. Elovici, and L. Giles, "Scholarly digital libraries as a platform for malware distribution," in *Proc. Singapore Cyber-Secur. Conf. (SG-CRC)*, 2017, pp. 1–153.

[20] N. Nissim, A. Cohen, and Y. Elovici, "ALDOCX: Detection of unknown malicious microsoft office documents using designated active learning methods based on new structural feature extraction methodology," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 631–646, Mar. 2017. doi: 10.1109/TIFS.2016.2631905.

[21] N. Šrndić and P. Laskov, "Detection of malicious PDF documents based on hierarchical document structure," in *Proc. 20th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2013, pp. 1–10.

[22] P. Laskov and N. Šrndić, "Static detection of malicious JavaScript-bearing PDF documents," presented at the 27th Annu. Comput. Secur. Appl. Conf. 2011, pp. 373–382.

[23] P. Baccas, "Finding rules for heuristic detection of malicious PDFs: With analysis of embedded exploit code," in *Proc. Virus Bull. Conf.*, 2010, pp. 1–7.

[24] J. Kittilsen, "Detecting malicious PDF documents," M.S. thesis, 2011.

[25] D. Maiorca, I. Corona, and G. Giacinto, "Looking at the bag is not enough to find the bomb: An evasion of structural methods for malicious PDF files detection," presented at the 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur., 2013, pp. 119–130.

[26] X. Wang, W. Yu, A. Champion, X. Fu, and D. Xuan, "Detecting worms via mining dynamic program execution," in *Proc. 3rd Int. Conf. Secur. Privacy Commun. Netw. Workshops (SecureComm)*, Sep. 2007, pp. 412–421.

[27] Z. Chen, M. Roussopoulos, Z. Liang, Y. Zhang, Z. Chen, and A. Delis, "Malware characteristics and threats on the Internet ecosystem," *J. Syst. Softw.*, vol. 85, no. 7, pp. 1650–1672, Jul. 2012.

[28] R. Herbrich, T. Graepel, and C. Campbell, "Bayes point machines," *J. Mach. Learn. Res.*, vol. 1, pp. 245–279, Jan. 2001.

[29] R. Moskovitch, N. Nissim, and Y. Elovici, "Malicious code detection using active learning," in *Privacy, Security, and Trust in KDD*. Berlin, Germany: Springer, 2009, pp. 74–91.

[30] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *J. Mach. Learn. Res.*, vol. 5, pp. 255–291, Dec. 2004.

[31] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and C. D. Bloomfield, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[32] N. Sebescen and J. Vitak, "Securing the human: Employee security vulnerability risk in organizational settings," *J. Assoc. Inf. Sci. Technol.*, vol. 68, no. 9, pp. 2237–2247, 2016.

[33] S. Makri, A. Blandford, J. Gow, J. Rimmer, C. Warwick, and G. Buchanan, "A library or just another information resource? A case study of users' mental models of traditional and digital libraries," *J. Assoc. Inf. Sci. Technol.*, vol. 58, no. 3, pp. 433–445, 2007.
[34] M. J. Harvey and M. G. Harvey, "Privacy and security issues for mobile health platforms," *J. Assoc. Inf. Sci. Technol.*, vol. 65, no. 7, pp. 1305–1318, Jul. 2014.
[35] R. Moskovitch, N. Nissim, and Y. Elovici, "Malicious code detection and acquisition using active learning," in *Proc. IEEE Intell. Secur. Inform.*, May 2007, p. 371.
[36] V. N. Vapnik, *Estimation of Dependences Based on Empirical Data*, vol. 41. Berlin, Germany: Springer, 1982.
[37] T. Joachims, "Making large-scale SVM learning practical," Tech. Rep., 1999.
[38] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proc. 17th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.* New York, NY, USA: Springer-Verlag, 1994, pp. 3–12.
[39] T. Mandl, U. Bayer, and F. Nentwich, "ANUBIS ANalyzing unknown BInarieS the automatic way," in *Proc. Virus Bull. Conf.*, vol. 1, 2009, p. 2.
[40] C. Willems, T. Holz, and F. Freiling, "Toward automated dynamic malware analysis using CWSandbox," *IEEE Secur. Privacy*, vol. 5, no. 2, pp. 32–39, Mar./Apr. 2007.
[41] A. Cohen and N. Nissim, "Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory," *Expert Syst. Appl.*, vol. 102, pp. 158–178, Jul. 2018
[42] K. Sigler, "Crypto-jacking: How cyber-criminals are exploiting the cryptocurrency boom," *Comput. Fraud Secur.*, vol. 2018, no. 9, pp. 12–14, Sep. 2018.
[43] R. Moskovitch, D. Stopel, C. Feher, N. Nissim, and Y. Elovici, "Unknown malcode detection via text categorization and the imbalance problem," in *Proc. IEEE Int. Conf. Intell. Secur. Inform.*, Jun. 2008, pp. 156–161.
[44] A. Cohen, N. Nissim, L. Rokach, and Y. Elovici, "SFEM: Structural feature extraction methodology for the detection of malicious office documents using machine learning methods," *Expert Syst. Appl.*, vol. 63, pp. 324–343, Nov. 2016.
[45] R. Moskovitch, D. Stopel, C. Feher, N. Nissim, N. Japkowicz, and Y. Elovici, "Unknown malcode detection and the imbalance problem," *J. Comput. Virol.*, vol. 5, no. 4, p. 295, 2009.
[46] N. Nissim, Y. Lapidot, A. Cohen, and Y. Elovici, "Trusted system-calls analysis methodology aimed at detection of compromised virtual machines using sequential mining," *Knowl.-Based Syst.*, vol. 153, pp. 147–175, Aug. 2018.
[47] C. Smutz and A. Stavrou, "Malicious PDF detection using metadata and structural features," in *Proc. 28th Annu. Comput. Secur. Appl. Conf.*, 2012, pp. 239–248.
[48] Z. Tzermias, G. Sykiotakis, M. Polychronakis, and E. P. Markatos, "Combining static and dynamic analysis for the detection of malicious documents," in *Proc. 4th Eur. Workshop Syst. Secur.*, 2011, p. 4.
[49] X. Lu, J. Zhuge, R. Wang, Y. Cao, and Y. Chen, "De-obfuscation and detection of malicious PDF files with high accuracy," in *Proc. 46th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2013, pp. 4890–4899.
[50] O. Henchiri and N. Japkowiczm, "A feature selection and evaluation scheme for computer virus detection," in *Proc. 6th Int. Conf. Data Mining (ICDM)*, Dec. 2006, pp. 891–895.
[51] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.
[52] C. Smutz and A. Stavrou, "When a tree falls: Using diversity in ensemble classifiers to identify evasion in malware detectors," in *Proc. NDSS*, San Diego, CA, USA, Feb. 2016.
[53] D. Liu, H. Wang, and A. Stavrou, "Detecting malicious javascript in PDF through document instrumentation," in *Proc. IEEE/IFIP 44th Annu. Int. Conf. Dependable Syst. Netw.*, Atlanta, GA, USA, Jun. 2014, pp. 100–111. doi: 10.1109/DSN.2014.92.
[54] Y. Freund, H. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Mach. Learn.*, vol. 28, pp. 133–168, Aug. 1997.
[55] D. Maiorca, B. Biggio, and G. Giacinto, "Towards adversarial malware detection: Lessons learned from PDF-based attacks," 2019, *arXiv:1811.00830*. [Online]. Available: https://arxiv.org/abs/1811.00830
[56] W. Zhao, J. Long, J. Yin, Z. Cai, and G. Xia, "Sampling attack against active learning in adversarial environment," in *Modeling Decisions for Artificial Intelligence*. Berlin, Germany: Springer, 2012, pp. 222–233.

NIR NISSIM received the Ph.D. degree (Hons.) from the Department of Information Systems Engineering, Ben-Gurion University (BGU), in 2016. He is currently a Researcher and the Head of the Malware Lab, Cyber Security Research Center, BGU. He is also a Lecturer of cyber security and machine learning topics with the Information Systems Engineering Department, BGU, and the Industrial Engineering and Management Department, Tel Aviv University. He is the Head of the ICSML program, which is an international cyber-security and machine learning academic, and professional program for international students. He published several noteworthy papers dealing with the development of a generic active learning framework aimed at the detection and acquisition of various types of malware in a variety of platforms. He is recognized as an expert in information systems security and machine learning solutions and has been leading several large-scale research projects in the field, including collaborative projects between academia and industry. In addition to his contributions to the cyber security domain, he is also interested in the bioinformatics domain and has published a number of papers regarding the efficient classification of condition severity. His main research interests include mobile and computer security, machine learning, and data mining. He received a prestigious prize in recognition of his ranking as the Faculty of Engineering Sciences' top doctoral student, in 2016. During his Ph.D. research, he received several best paper awards in top ranked scientific international conferences and awards of excellence at BGU, and he was one of a few doctoral students at BGU to win an exclusive doctoral cyber security scholarship granted by the Israeli Cyber Security Bureau.

AVIAD COHEN is pursuing the Ph.D. degree with the Department of Information Systems Engineering, BGU. He is currently a Researcher with the Malware Lab. His research is aimed at the development of a framework and the detection of malicious email, which takes a novel holistic (addressing all email components) approach to the analysis of suspicious email. He is a coauthor of several papers dealing with the analysis and detection of malicious PDF and Office documents. His main research interests include computer and cyber security, machine learning, and big data analytics.

JIAN WU graduated from the Department of Astronomy and Astrophysics, Pennsylvania State University, in August 2011. He received the bachelor's degree in physics and astronomy from the University of Science and Technology of China (USTC), in 2004. He joined the CiteSeerX team led by Dr. C. Lee Giles. He is the Tech Leader of the CiteSeerX project. He led a small team to scale the CiteSeerX collection from 3 to 10 million academic documents. He was an Assistant Teaching Professor with the College of Information Sciences and Technology (IST), Pennsylvania State University. He is currently an Assistant Professor with the Computer Science Department, Old Dominion University, Norfolk, VA. He has published nearly 30 20 peer-reviewed papers in ACM, IEEE, and AAAI conferences, receiving one best application paper award and two best paper nominations. He also processed and analyzed astronomical big data earlier in his career and published seven journal papers in the *Astrophysical Journal* (ApJ), *the Astronomical Journal* (AJ), and *Monthly Notices of the Royal Astronomical Society* (MNRAS). He was the Co-PI of NASA and NSF proposals. He has mentored at least 20 students towards their bachelor's or master's theses and teaches two undergraduate level courses.

ANDREA LANZI is currently an Assistant Professor with the Computer Science Department, Universita' degli studi di Milano, Italy, where he is leading a security Lab, called LaSER (systems and network security lab). From 2009 to 2013, he was a Senior Researcher with the S3 lab, Eurecom Graduate School, Sophia Antipolis, French Riviera, where he has been a part of the iSeclab group. From 2007 to 2009, he was a Ph.D. Visiting Scholar with the GTISC Security Lab, led by the Prof. Wenke Lee, Georgia Tech University, USA. He is interested in several aspects of cyber security. In particular, his main area of research deals with host intrusion detection systems (HIDS), memory errors exploitation, reverse engineering, and malware and forensic analysis. In recent years, he has mainly studied the application of emulation/virtualization and compiler techniques for malware analysis and detection in Android context. In addition, he has been working on analyzing large-scale security malware datasets to investigate the behavior of current cyber threats.

LIOR ROKACH is a Data Scientist and a Professor with the Department of Software and Information Systems Engineering, where he is currently the Chair. He has established the machine learning lab and the big data lab, which promote innovative adaptations of machine learning and data science methods to create the next generation of intelligent systems. His research interests include the areas of data science, machine learning, big data, deep learning and data mining, and their applications to: recommender systems, cyber security, information retrieval, information extraction, and medical informatics. He currently serves as an Editorial Board Member of ACM *TIST* and an Area Editor of *Information Fusion* (Elsevier).

YUVAL ELOVICI received the B.Sc. and M.Sc. degrees in computer and electrical engineering from BGU and the Ph.D. degree in information systems from Tel Aviv University. He was the Head of the software engineering program at BGU for two half years. For the last 11 years, he has led the cooperation between BGU and Deutsche Telekom. He also consults professionally in the area of cyber security and is the Co-Founder of Morphisec, a startup company that develops innovative cyber security mechanisms that relate to moving target defense. He is currently the Director of the Telekom Innovation Laboratories, Ben-Gurion University of the Negev (BGU), the Head of the Cyber Security Research Center (CSRC), BGU, the Research Director of iTrust, SUTD, and a Professor with the Department of Information Systems Engineering, BGU. He has published articles in leading peer reviewed journals and conferences. He has coauthored books on social network security and information leakage detection and prevention. His primary research interests include computer and network security, cyber security, Web intelligence, information warfare, social network analysis, and machine learning.

LEE GILES is currently the David Reese Professor with the College of Information Sciences and Technology, Pennsylvania State University, University Park, PA. He is also a graduate college Professor of computer science and engineering, a courtesy Professor of supply chain and information systems, and the Director of the Intelligent Systems Research Laboratory. He became a Teaching and Learning Technology Fellow and the Interim Associate Dean of Research for IST. He directs the Next Generation CiteSeer and CiteSeerx project and codirects the ChemxSeer project at Pennsylvania State University. He has been associated with Columbia University, the University of Maryland, University of Pennsylvania, Princeton University, and the University of Trento. His research is or has been supported by NSF, NASA, DARPA, Microsoft, FAST Search and Transfer, Ford, IBM, Internet Archive, Lockheed-Martin, Alcatel/Lucent, NEC, Raytheon, Smithsonian, US Department of Treasury, and Yahoo. He has consulted for or been on advisory boards of NEC, FAST Search and Transfer, PJM, KXEN, Databrary, US Department of Treasury, and the US Department of Defense. He and his collaborators, including current and former graduate students, have published over 400 journal and conference papers, book chapters, edited books, and proceedings. His work is very well cited, according to Google Scholar, and his is one of the top 100 h-indexes in computer science and information retrieval. His recent work on scholarly big data and access has appeared, in 2014, in PLOSOne and was in Nature, Science, and other news. His 2006 coauthored paper in Science proposes a cyberinfrastructure for the historial sciences. His coauthored paper, in 2004, in the Proceedings of the National Academy of Sciences created an automatic acknowledgement indexing methodology and showed that various funding agencies and individuals in computer and information science are much more acknowledged than others. In 2002, he coauthored the paper Winners Don't Take All published in the Proceedings of the National Academy of Sciences on how the topic based web does not follow a power law distribution. In 1998, he coauthored a paper published in Science on the size and search engine coverage of the Web that was well cited in the popular press and, in 1999, a well-received follow-up paper in Nature. Several of his papers have received or been nominated for best paper awards and have been reprinted in edited collections. His research has been highlighted in many places, including the Society for Industrial and Applied Mathematics (SIAM) news, *Wired* Magazine, the Wall Street Journal, the Washington Post, the New York Times, Nature news, and Science news. He plays an active professional role in scientific and technical and communities. He serves on many related conference program committees and has helped organize many related meetings and workshops. He has given many invited and keynote talks and seminars. He has been or is an Advisor and Reviewer to USA and other government and university research programs. He has served on the Editorial Boards of the IEEE Intelligent Systems, the IEEE Transactions on Knowledge and Data Engineering, *Machine Learning Journal*, *Computational Intelligence and Applications*, the IEEE Transactions on Neural Networks, *Journal of Computational Intelligence in Finance*, *Journal of Parallel and Distributed Computing*, *Neural Networks*, *Neural Computation*, and *Academic Press*.