Hindawi
Security and Communication Networks
Volume 2021, Article ID 6615899, 29 pages
https://doi.org/10.1155/2021/6615899

Review Article

Use of Security Logs for Data Leak Detection: A Systematic Literature Review

Ricardo Ávila,[1] **Raphaël Khoury**,[1] **Richard Khoury**,[2] **and Fábio Petrillo**[1]

[1]*Département d'informatique et de Mathématique, Université du Québec à Chicoutimi, Québec, Canada*
[2]*Department of Computer Science and Software Engineering, Université Laval, Québec, Canada*

Correspondence should be addressed to Ricardo Ávila; ricardo.lims@gmail.com

Received 26 October 2020; Revised 19 January 2021; Accepted 19 February 2021; Published 11 March 2021

Academic Editor: Flavio Lombardi

Security logs are widely used to monitor data, networks, and computer activities. By analyzing them, security experts can pick out anomalies that reveal the presence of cyber attacks or information leaks and stop them quickly before serious damage occurs. This paper presents a systematic literature review on the use of security logs for data leak detection. Our findings are fourfold: (i) we propose a new classification of information leaks, which uses the GDPR principles; (ii) we identify the twenty most widely used publicly available datasets in threat detection; (iii) we describe twenty types of attacks present in public datasets; and (iv) we describe thirty algorithms used for data leak detection. The selected papers point to many opportunities that can be investigated by researchers interested in contributing to this area of research.

1. Introduction

Cybercriminals seek to access, modify, or delete confidential information for financial gain, fame, personal revenge, or to disrupt organizational services [1]. These attackers exploit software vulnerabilities, the high workload and inexperience of employees, and the heterogeneity of security solutions implemented in an organization to carry out their attacks. In this context, organizations must develop strategies that allow them to be resilient in the face of malicious attacks. Security mechanisms create layers of protection and generate event logs that can be analyzed to detect and react to possible intrusions. By studying these logs, security analysts can detect and respond to attacks as they occur, rather than forensic investigation weeks or months after an incident.

Security logs are thus a crucial tool in the detection of attacks and information leaks. However, using them comes with several important challenges. It requires attention to detail in order to pick out anomalous elements in a long list of events. The massive size of modern security logs makes it necessary to analyze a very high volume of data in a short amount of time. The heterogeneity of devices and systems in a current corporate computer ecosystem, and thus the heterogeneity of logs they generate, also contributes to log analysis complexity.

In this paper, we present a Systematic Literature Review (SLR) of 33 published, peer-reviewed studies on the use of security logs to detect information leaks. We mapped the state-of-the-art topic and its implications for future research, aiming to create approaches to detect and react to attacks. More specifically, this review makes four key contributions:

(1) A description of different types of information leaks that uses the recommendations of GDPR

(2) The identification and description of the 20 most commonly used publicly available benchmark datasets of security logs

(3) The identification and description of 20 types of attacks that can be detected through an analysis of security logs

(4) The identification and description of 30 algorithms used for data leak detection

The remainder of this study is organized as follows: in Section 2, we describe the protocol used in this systematic literature review. In Sections 3, 4, 5, 6, and 7, we present the

main results, observations, and recommendations. Threats to validity are discussed in Sections 8. Related works are presented in Section 9. Section 10 closes the study with a conclusion and a discussion of future work.

2. Systematic Review Method

An SLR is an impartial and objective research methodology aimed at answering a set of research questions by analyzing all pertinent literature in a specific area. This study identifies, classifies, and evaluates the current state-of-the-art use of security logs from different perspectives.

This study follows the guidelines made available by Kitchenham and Charters [2] and Petersen et al. [3]. The main steps of our review protocol are as follows: (i) identify research questions; (ii) devise a search strategy; (iii) define inclusion and exclusion criteria; and (iv) select studies to review; and (v) extract and integrate data. This study was conducted from May to September 2020.

2.1. Research Questions. The primary purpose of this work is to define the state-of-the-art on the use logs in information security. Specifically, we aim to answer the following research questions (RQs):

(1) RQ1: what are the different types of information leaks? The goal is to identify the different types of information leaks, proposing a new classification based on the General Data Protection Regulation (GDPR). The results are presented in Section 3—(RQ1).

(2) RQ2a: what are the different types of logs that are used in practice and in the academic literature? Find out the types of logs, formats, and main public datasets used to detect anomalies, providing a brief description of each of these datasets. The results are presented in Section 4—(RQ2A).

(3) RQ2b: can datasets be useful to detect data leaks after following GDPR guidelines? Recognize the main characteristics that a dataset must have to be valuable in the information leak identification process. The results are presented in Section 5—(RQ2B).

(4) RQ3: what are the different types of attacks that exist? Identify the types of attacks reported in the literature, describing the characteristics of each one. The results are presented in Section 6—(RQ3).

(5) RQ4: what are the main algorithms used in the analysis of logs to detect information leaks? The focus was to identify algorithms utilized to detect information leaks and highlight their main characteristics. The results are presented in Section 7—(RQ4).

2.2. Search Strategy. The search terms are constructed using a three steps process: (i) we list key concepts addressing the different elements of the research questions, (ii) we enrich the list by adding synonyms for each of the keywords that were found, and (iii) we build a search query based on the combination of key terms and their synonyms, using the OR and AND operators. The result of this process is the following search query:

("event log" OR "information leaks" OR "data leaks") AND ("algorithm * detection" OR "security information")

After ensuring that the keywords selected returned the highest number of studies related to our review, we finalized the search string. The string keywords were matched only with the title, abstract, and keywords of the papers in four academic search engines (ACM, ScienceDirect, IEEE Xplore, and Scopus). We then proceeded with the paper inclusion and exclusion process. The inclusion (IC) and exclusion (EC) criteria adopted in this mapping fulfilled the following requirements:

(1) Inclusion criteria (IC)

  (i) IC1: papers within the context of information leaks or cybersecurity.
  (ii) IC2: reports of empirical studies or reviews of information leaks.
  (iii) IC3: papers must be peer-reviewed.

(2) Exclusion criteria (EC)

  (i) EC1: studies where security logs are only used as an example.
  (ii) EC2: books, web sites, technical reports, pamphlets, dissertations, and white papers.
  (iii) EC3: papers not written in English.
  (iv) EC4: duplicate papers or studies that repeat previously-published research.

In addition to the use of search engines, we use the snowballing technique [4] to search for additional relevant literature. Snowballing is a technique used to discover relevant papers based on a study's references (backward snowballing) and on works that cite the study (forward snowballing) [5].

2.3. Data Extraction Process. Our data extraction process consists of seven steps. Each step is applied on the papers remaining after applying the previous steps.

(1) Step 1: we apply our search query on our selected academic search engines and obtain an initial list of sources.

(2) Step 2: we remove all duplicates, based on the paper's title and authors.

(3) Step 3: we remove all studies that were not peer-reviewed.

(4) Step 4: we analyze the titles, abstracts, and keywords of each paper and eliminate any that meet the exclusion criteria.

(5) Step 5: we read the introduction and conclusion of each paper and remove any that meet the exclusion criteria.

(6) Step 6: we read each paper in its entirety and remove any that meet the exclusion criteria.

(7) Step 7: we apply the forward and backward snowballing techniques to obtain additional sources. Steps 2 to 7 are then applied for each additional paper added at this step.

Figure 1 indicates the number of papers retained from each source after applying each step. We record the details of the data extraction process in a spreadsheet, which we make available to interested researchers (https://bit.ly/ExtractionPapersDataLeaks). This spreadsheet lists every paper considered, and those that were discarded are identified alongside their exclusion criteria.

2.4. Collected Studies. As shown in Figure 1, the different steps of the selection process and the studies selected in each digital database are described below.

Automatic Search. We executed our search strategy on four search engines and retrieved a total of 174 studies.

Duplication Removal. Scopus indexes studies from several digital databases, including ACM and IEEE Xplore; thus, it was expected that duplicate studies could be present among the 174 selected articles. In this step, duplicate studies were removed, reducing the number of our studies to a total of 167.

Removal of Non-Peer-Reviewed Studies. We removed all studies that are not peer-reviewed as well as challenges, editorials, position papers, showcases, panel discussion, and keynotes. This step reduced the number studies to 163.

Title, Abstract, and Keyword-Based Selection. We read the title, abstract, and keywords of the studies to decide whether or not each of the retrieved papers were relevant to our SLR. In those cases, when we were not able to reach a definite verdict at this step, we preserved the study for the next step of the selection process. This step reduced the number of studies to 54.

Introduction and Conclusion-Based Selection. We read the introduction and conclusion of each of the 54 studies to ensure that they were related to our SLR. During this step, we discarded 30 papers, leaving us with 24 studies.

Full-Text Selection. We thoroughly read each of the 24 selected studies in this step. A total of 17 studies were selected for inclusion based on reading the entire text.

Snowballing. We used the snowballing technique [4, 5] to find, identify, select, and take samples studies of the 17 selected papers. We found 79 potentially related studies. After applying the inclusion and exclusion criteria, we selected 16 studies and added them to our corpus. This brought the final number of papers for this SLR to 33. In the remainder of this paper, these studies are referred to as the selected studies and are prefixed with an S in the Appendix.

Our SLR covers studies published before July 15th, 2020, when the search was performed. We observed that the number of studies selected from 2002 to 2014 was lower than those from 2015 to 2020. This growth in the number of studies in cybersecurity documents is in line with the predictions of Cardenas et al. [6], who foresaw an increase in the use of machine learning and big data technologies in the field of information security.

This increase also can be explained by the following two reasons: (i) cyberattacks are increasingly carried out by organized groups rather than by isolated individuals and (ii) traditional software technologies are incapable of processing the high volume, high velocity, and heterogeneous data generated by security events. These factors have led to the adoption of machine learning and big data technologies by cybersecurity researchers.

3. RQ1: What Are the Different Types of Information Leaks?

Data leakage (or information leakage) is the unauthorized transfer of data (or information) from inside an organization to an external destination or container (e.g., flash drive, and CD). Data leaks may be an electronic or physical method intentionally or maliciously by an insider or outsider in the company. To identify suspicious activity, security analysts must continuously monitor security logs in different systems and resources, searching for evidence of information leaks in high volumes of data. According to Khan et al. (S2), information leaks can be initiated through an external or internal source and are usually the consequence of exploiting vulnerabilities.

Among the selected studies, there is no uniformity regarding the categorization of different types of information leakage (S2, S3, S11, S16, S21). Furthermore, a general, all-purpose classification of information leaks that uses the General Data Protection Regulation (GDPR) (https://gdpr-info.eu) guidelines was not found in the literature.

We thus propose a novel classification that is concordant with the GDPR guidelines on companies' use of personal data. According to GDPR, "personal data" refers to any information relating to an identified or identifiable natural person directly or indirectly by referencing an identifier such as a name, an identification number, and location data. Knowing that personal data belong to the individual, companies must create mechanisms to protect the data against attacks by third parties, anonymize it, or even delete them upon request.

Compliance with the GDPR guidelines is essential to create an actionable classification of different information leaks. Consequently, the seven GDPR principles are listed and explained below [7]:

(1) Lawfulness, fairness, and transparency—personal data shall be processed lawfully, fairly, and transparently with the data subject.

(2) Purpose limitation—personal data shall be collected for specified, explicit, and legitimate purposes and not further processed in an incompatible manner with those purposes.

(3) Data minimization—personal data shall be adequate, relevant, and limited to what is necessary concerning the purposes for which they are processed.

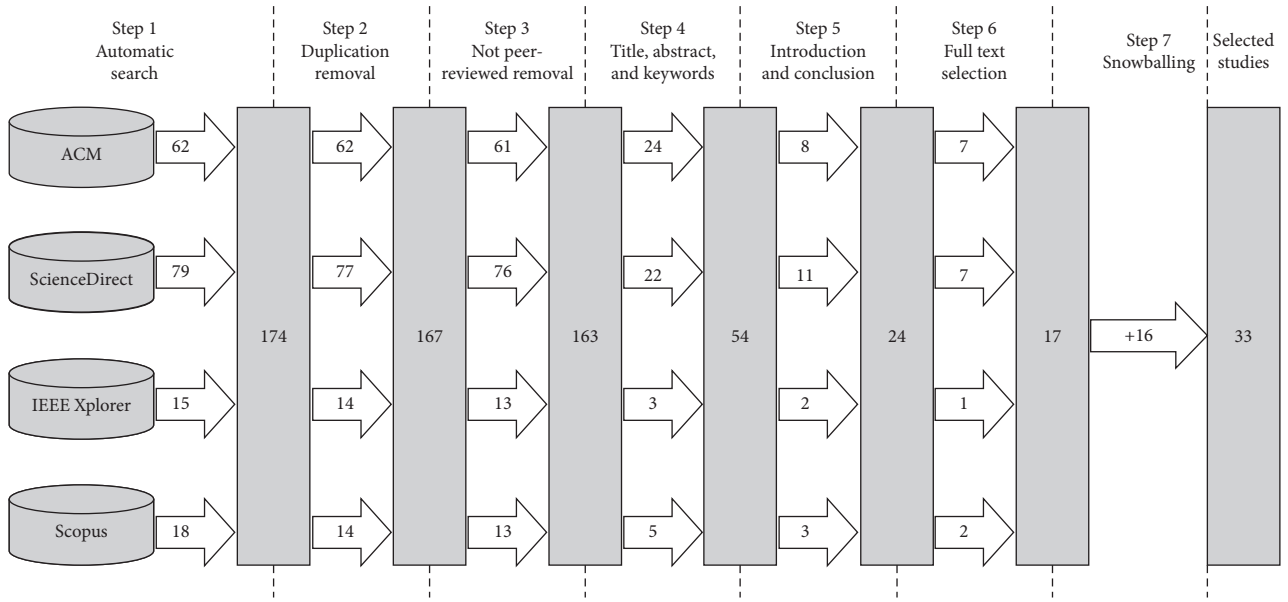

Figure 1: Steps of studies selection process.

(4) Accuracy—personal data shall be accurate and, where necessary, kept up to date.

(5) Storage limitations—personal data shall be kept in a form that permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed.

(6) Integrity and confidentiality—personal data shall be processed in a manner that ensures appropriates security of the personal data, including protection against unauthorized or unlawful processing and accidental loss, destruction, or damage, using appropriate technical or organizational measures.

(7) Accountability—the controller shall be responsible for and demonstrate compliance with the GDPR when requested by the authorities.

The classification presented by Seo and Kim [8] is based on the types of threats that occur in organizations. The authors propose a quantitative model based on internal and external threats. The study uses realistic internal threats through the definition, classification, and correlation/association analysis of various human-machine records of acts associated with security breaches that occur in an organization. The authors' proposal does not meet the recommendations of the GDPR, and, even though identifying personal data as threatening information, it focuses much more on the types of threats than on the kind of information that is exfiltrated.

Other studies have different classifications of data leak threats. A study by Alneyadi et al. [9] classifies data leaks based on their causes and considers whether the confidential information breach was carried out intentionally or not. On the other hand, a study by Shabtai et al. [10] follows a different approach based on the origin of the threat that caused the data leak: internal or external threats. These approaches by type of threat can cover all aspects related to incidents and vulnerability detection. However, they do not take into account data privacy and GDPR recommendations.

Instead, we propose the following classification of information leaks in four types, based on the type of data that has been leaked: personal, company, trade secrets, and analytics. Table 1 presents a summary of our information leaks classification according to the GDPR principles.

The most common type of information leak is of *personal information* from users, customers, and employees (S2, S11). This type of leak can occur either due to the individual's failure or the company's failure. It includes identifying information (name, address, social security numbers, and password), medical or health information (electronic medical record, medications used, and health plan), online activity information (IP address, navigation habits, payment history, and usage details), or financial records (credit card numbers, CVV code, and expiration dates). From the perspective of the GDPR, this is the most sensitive type of information, and companies must take particular care in protecting themselves against leaks of this data.

Another type of data that can be leaked is *company information* (S2). In this case, the leak may include strategic information (business plan, strategic map, and roadmaps), internal communications [11] (memos, emails, and confidential documents), or metrics about the company's operations (statistical data, inventory, sales projection, and other data collected about the company's daily operations). Companies' internal emails can include private customer information, making it challenging to classify it as personal or company information. An acceptable information security policy that specifies each document's privacy concerns and secrecy levels must be implemented to mitigate this problem. To create this information security policy, companies can implement the recommendations of ISO/IEC 27001:2013—Information Security Management Systems (https://www.iso.org/standard/54534.html).

Table 1: Information leaks and GDPR principles.

Information leaks	Important information assets	Threats	GDPR principles to follow
Personal information	Identifying medical or health online activity financial records	Loss, modification, disclosure, damage	1 to 7
Company information	Internal communication metrics of operations	Loss, modification, disclosure, damage, interruption	1 to 7 when using personal data
Trade secrets	Plans, formulas, and design source code business planning	Loss, modification, disclosure, damage, interruption	—
Analytical information	Model-based demographic behavior	Loss, modification, disclosure, damage, interruption	1 to 7 when using personal data

The third type of information leak concerns the loss of *trade secrets* (S16). Generally, it consists of critical and sensitive information that, if exposed, can allow competitors to gain an advantage, thus incurring financial losses for the company and even lead to bankruptcy (S3, S21). Trade secrets include plans, formulas and designs (creation of new products and services), and source code or business planning (market strategy, contacts with suppliers, and customers). The loss of trade secrets may negatively affect a company, but it is less sensitive from the perspective of the GDPR, since the latter focuses on personal data.

Finally, *analytical information* consists of data from different sources that synthesize trends, patterns, and trajectories of the overall business environment and are used as the basis for decision-making (S2). Analytical information can be demographic (age, gender, and location), behavioral (personality attributes and detailed navigation information on a website), or model-based (constructed characteristics based on various details collected). As this analytical information can use personal data and understanding that data does not belong to the company, they must be anonymized or even deleted when requested, as explained previously. Like was the case with company information, analytical information can sometimes contain personal information, and if this is the case, then the information must be awarded the highest degree of protection.

The main advantage of our proposal to classify information leaks is that it follows the GDPR principles when it comes to personal information, namely, that from the moment that companies acquire personal data and use it as part of their decision-making processes, the strictness of the confidentiality policy governing the data must be increased. Also, control mechanisms must be created to facilitate the anonymization and deletion of personal information. Therefore, following this classification makes it easier to ensure compliance with the GDPR.

4. RQ2A: What Are the Different Types of Logs That Are Used in Practice and in the Academic Literature?

According to the studies we selected, network traffic is captured either in packet-based or flow-based format, usually mirroring ports on network devices during a specified period. Several studies (e.g., (S13, S18)) detect anomalies using public datasets to understand their characteristics. Alongside public datasets, some studies report that they have started collecting security event logs from the internal network.

Collecting and filtering security alerts to create a dataset is challenging, and the process includes several assessments and validation steps (S9). New types of attacks and more complex programs and network structures frequently appear, requiring updates to the dataset.

Table 2 summarizes the different types of security logs used in the literature from multiple sources. These types of security logs are the source for the creation of public datasets. Security logs are crucial to the detection of anomalies and information leaks. However, the collection, analysis, and management of logs raise several complex challenges. It requires attention to details, the ability to analyze a high volume of data intractable time and often the ability to collate multiple different logs from diverse sources such as individual computers, networking equipment, and mobile devices. This process of collecting data from different sources makes it difficult to update and maintain detection models, opening malicious agents' vulnerabilities.

Security log data is an essential resource for continuous monitoring, providing organizations with a complete and real view of their infrastructure. Companies need to understand that this type of routine is not an expense but one way to reduce the costs and subsequent losses caused by security breaches and increase the quality of service.

On the other hand, security logs can be made publicly available in datasets' format to researchers and professionals interested in understanding the leakage of information, thus adopting solutions that are more appropriate to their needs. These datasets, their description, and access location are listed in Table 3, which provides an overview of all commonly used datasets.

Our SLR identified 20 benchmark datasets publicly available containing data in different formats such as traffic captures, security logs, or other types of execution traces. We have organized the description of these datasets according to the following properties:

(1) Year of creation—the year of creation of the dataset.

(2) Type of traffic—traffic can either be real (captured within a productive network environment) or emulated (captured within a testbed or emulated network environment). Furthermore, it can generate in networks of different size (small, enterprise, or university) or from a honeypot (computational

Table 2: Different types of security logs.

Types of security log	Source	References
Application log	Web applications, database	S7, S12, S16, S23, S26
Audit log	Security audit log	S12
Network log	DNS, DHCP, Firewall, IDS, Proxy	S4, S6, S7, S12, S16, S29
Security log	Event logging and monitoring services	S7
Setup log	Ex.: Log files from Msiexec.exe	S16
System log	Syslog, log and event manager	S12, S15, S16
Virtual machine logs	Microsoft Hyper-V, Oracle VM Virtualbox, VMWare	S7
Web-server log	Apache Web Server, IIS, Jboss, Tomcat	S7, S9, S10, S16

Table 3: General characteristics of datasets.

Dataset	Type of traffic/size	Format	Data volume	Labeled	Anonymity	Attack class	References
DARPA-99	Emulated/small network	Logs	9 GB packets	Yes	None	DOS, U2R, R2L, PROBE	S18, S20, S22, S24, S28, S32, S33
KDD-99	Emulated/small network	Others	5 m points	Yes	None	DOS, U2R, R2L, PROBE	S13, S18, S20, S28, S32, S33
DARPA-98	Emulated/small network	Logs	38 MB packets	Yes	None	DOS, U2R, R2L, PROBE	S20, S24, S28, S31, S33
UNSW-NB15	Emulated/small network	Others	2 m points	Yes	None	Fuzzers, DoS, backdoors, exploits	S17, S27, S33
NVD	Real/system calls	CSV, XML, text, HTML	139 k points	Yes	None	Buffer overflows, trojans, others	S5, S8, S21
CAIDA	Emulated/ enterprise network	Logs	179 TB packets	No	Yes	DoS, DDoS, others	S18, S28
CTU-13	Emulated/ university network	Uni. & bi.flow	81 m flows	Yes	Yes	Botnets	S17, S33
NSL-KDD	Emulated/small network	Others	150 k points	Yes	None	DOS, U2R, R2L, PROBE	S27, S28
DARPA-00	Emulated/small network	Logs	385 m packets	Yes	None	DOS, U2R, R2L, PROBE	S17, S20
ISCX-12	Emulated/small network	bi.flow	2 m flows	Yes	None	DOS, fuzzers, backdoors, others	S27, S28
ADFA	Emulated/system calls	Logs	403 MB packets	Yes	None	Exfiltration, DDoS, others	S28, S33
CSIC-10	Emulated/small network	CSV, logs	99 MB packets	Yes	None	SQL injection, cross-site scripting	S33
DEF CON	Real/small network	Uni. And bi. Flow, logs, others	10 GB packets	No	None	Botnet, various	S28
PKDD-07	Emulated/small network	XML, CSV	1.9 GB packets	Yes	None	Cross site scripting, SQL injection	S33
ENRON	Real/enterprise network	Text	0.5 m messages	No	Yes	n.s.	S30
ICS	Emulated/small network	ARFF, CSV	1.4 GB packets	Yes	Yes	DoS	S28
ISOT	Emulated/small network	Packet	11 GB packets	Yes	None	DoS, botnet	S33
KYOTO	Real/honeypots	Others	93 m points	Yes	Yes	DoS, U2R, R2L, PROBE	S28
LBNL/ICSI	Real/enterprise network	Packet	11 GB packets	No	Yes	Brute-force, botnets, DoS	S18
PREDICT	Real/enterprise network	Logs	25 GB packets	Yes	None	Botnet, DoS, DDoS, others	S28

n.s. = not specified, m = million, k = thousand, uni. = unidirectional, bi. = bidirectional, TB = terabyte, GB = gigabyte, MB = megabyte.

resources dedicated to being probed, attacked, or compromised, in an environment that allows the registration and control of these activities [12]).

(3) Format—datasets are available in a variety of formats including (i) network packets, which contain network traffic (for example, pcap); (ii) flow-based, which contain unilateral or bilateral metadata about network connections (for example, NetFlow); (iii) logs collected from different sources, such as the types of security logs presented in Table 2; (iv) CSV, XML, Text, ARFF, or HTML files; and (v) other formats, which contain additional attributes to enrich the data.

(4) Data volume—the number of packets/flows/points/lines/messages, or the size in TB (terabyte), gigabyte (GB), and megabyte (MB).

(5) Labeled—this property indicates whether the datasets are labeled or not. The types of labels vary according to dataset.

(6) Anonymity—this property indicates if the data are anonymized or not. For privacy reasons, some datasets are only available in anonymized form. The data anonymization process keeps the source anonymous, protecting private or confidential information, such as user names, IP addresses, and social security numbers, among others.

(7) Attack class—indicates the types of attacks present in the dataset. A description of each attack type is detailed in Section 6.

4.1. DARPA-99. This dataset (https://kdd.ics.uci.edu) contains emulated network traffic and the entire contents of the data packets written in the tcpdump (data-network packet analyzer program that runs into a command-line interface) format. The dataset consists of three weeks of data for which no attacks were reported in the first and third weeks. The second week includes a number of simulated attacks. The goal was to provide examples of how to report attacks after they are detected. This dataset is not anonymized and contains logs from a small network, where each sample is classified either as normal (no attacks) or as exhibiting one of the following types of attacks:

(1) Remote-to-local (R2L)—a type of attack that is executed in order to access a specific network address remotely illegally.

(2) User-to-root (U2R)—occurs when the attacker performs a privilege escalation attack.

(3) Denial of service (DoS)—an attack type that seeks to make a machine or network resource unavailable to legitimate users.

(4) Probe—an attack that scans the network to collect the information on computers in order to identify any vulnerability present therein.

This dataset is the most cited among the selected studies with seven references and used in several different contexts. Notably, Shiravi et al. (S18) use it when proposing an approach to create synthetic datasets suitable for various situations and capable of effectively decreasing the need for public datasets. According to their study, open datasets are heavily anonymized, obscuring the relationships between entities and renders them less useful to researchers. The authors also claim that DARPA-99 tends to be out of date with more recent or sophisticated attacks. The dataset proposed by the authors uses real traffic data, unlike the emulated traffic of DARPA-99.

The studies of Buczak and Guven (S20) and Ahmed et al. (S28) cite datasets (KDD-99, DARPA-98, and DARPA-00), in which the DARPA-99 was used to create efficient data leak detection models and also to explain their performances. The paper reviews the literature on the topic of machine learning and data mining methods used for cybersecurity. According to the authors, the most significant gap observed is the unavailability of labeled data, suggesting that the community should invest more in collecting and labeling publicly available data.

Pietraszek (S22) uses the DARPA-99 dataset to assess an attack detection classifier's performance, comparing the results with the alerts generated in real-time by Snort. Cheng et al. (S24) also used this dataset to validate their approach to classifying intrusions in the data. Finally, Shah et al. (S32) and Yavanoglu and Aydos (S33) both use the DARPA-99 dataset to validate machine learning techniques to identify attacks. In general, these studies point to DARPA-99 as a valuable data set for evaluating each proposed approach, besides suggesting that capturing real-world traffic can add background knowledge and improve classification accuracy.

4.2. KDD-99. This dataset was created for the Third International Data Discovery and Data Mining Tools Competition, part of the KDD-99, the Fifth International Conference on Knowledge Discovery and Data Mining. The dataset is available on the website of the University of California (https://kdd.ics.uci.edu). The participants were tasked with creating a network intrusion detector, a predictive model capable of distinguishing between benign or malicious connections. This database contains a set of data to be audited, including a wide variety of simulated intrusions in a controlled network environment: DoS attacks, remote-to-local (R2L), user-to-root (U2R), and probe. The data are not anonymized.

The KDD-99 dataset was cited by six studies in our sample literature set. Ullah and Babar (S13) report that KDD-99 is one of the most widely used datasets for analyzing security event data and protect organizational networks, computers, and data from cyberattacks. The authors claim that there is no standard for comparing the accuracy of machine learning algorithms on the same dataset or different subsets of the same dataset.

The other studies (S18, S20, S28, S32, and S33) also mention the use of KDD-99 to make comparisons with other datasets and validate their approach for intrusion detection. These studies indicate KDD-99 as an outdated dataset, with attack scenarios that hardly occur today. The authors also suggest the tcpdump traffic collector, used to generate the

dataset, can frequently become overloaded and discard packets when the traffic load is heavy.

4.3. DARPA-98. This dataset (https://www.ll.mit.edu/r-d/datasets) was developed by the MIT Lincoln Laboratory and is one of the oldest datasets still in use in cybersecurity. The dataset consists of a sample of artificial attack injections delivered to the Air Force Research Laboratory (AFRL) and used to evaluate internal systems' security. These data were first made available in February 1998 and has been updated with new data in 1999 and 2000. The dataset is nonanonymized, contains logs from a small network, and records the occurrence of several types of attacks, including DoS attacks, remote-to-local (R2L) attacks, user-to-root (U2R) attacks, and probe.

Five studies cited this dataset. Liao and Vemuri (S31) proposed a new classifier algorithm of intrusion detection and used DARPA-98 to verify the effectiveness of their approach to detecting intrusive program behavior. According to the authors, the dataset generated with simulated network traffic does not matter, as the thirty-eight types of attacks and various realistic intrusion scenarios were conducted in a real network environment.

All other studies (S20, S24, S28, and S33) used the DARPA-98 to compare their results with different datasets, verifying the effectiveness of their approach. According to these studies, the fact that so many papers use DARPA-98 is related to how arduous and time-consuming it is to acquire a representative dataset. Once a dataset is available, researchers start to use it.

4.4. UNSW-NB-15. This dataset was created by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS), and consists of real and synthetic access activities classified as normal or attack behaviors. The dataset is available for download at the University of New South Wales (UNSW) web page (https://www.unsw.adfa.edu.au). The dataset is nonanonymized, contains logs from a small network, and exhibits nine types of attacks:

(1) Analysis—it includes different attacks of port, spam, scan, and HTML files penetrations.

(2) Backdoors—attack in which a system security mechanism is bypassed furtively to access a computer or its data.

(3) Denial of service (DoS)—a malicious attempt to make a resource or server unavailable to users.

(4) Exploits—type of attack where the attacker knows of a security problem within an OS or a piece of software and uses that knowledge by exploiting vulnerabilities.

(5) Fuzzers—attempting to cause a program or a network suspended by seeding randomly created data.

(6) Generic—a technique that works against all-ciphers, without considering the structure of the block-cipher.

(7) Reconnaissance—includes different types of attacks that can simulate attacks that collect information.

(8) Shellcode—one small piece of code used as the payload in the scanning of computer program vulnerability.

(9) Worms—attacker replicates itself to spread to other computers, relying on security failures on the target computer to access it.

Three studies in our sample set cited this dataset. Shin et al. (S17) proposed a feature extraction module that was validated using the UNSW-NB-15 dataset. The proposed module allows users to describe the features as four data types: nominal (N), integer (I), floating point (F), and Bool (B). Likewise, flow features are classified into five different categories: basic flow information, flow identification, payload contents information, time-related information, and additional features. According to the authors, the evaluation of the proposed framework showed good performance, with F-measure of 0.9380 as the maximum, which observed using a modified apriori algorithm.

A study by Idhammad et al. (S27) proposed a semi-supervised DDoS detection approach based on entropy estimation that used UNSW-NB-15, NSL-KDD, and ISCX-12 to validate their strategy. The authors affirm that experiment results, in terms of accuracy and false-positive rate, are satisfactory in DDoS detection methods, even using datasets with emulated network traffic data.

The NSW-NB-15 is one of the datasets used by Yavanoglu and Aydos (S33) to validate the use of machine learning techniques to detect attacks. This study also claims that UNSW-NB-15, compared to existing datasets, has several attack families that reflect modern attacks. According to selected studies in this SLR, this is one of the most recent public datasets, along with NVD and ICS.

4.5. NVD. The National Vulnerability Database (NVD) is another important dataset. Even though it is not a dataset with security logs, it is a reference library widely used by researchers and professionals in cybersecurity. It consists of a detailed description of upwards of 140.000 documented vulnerabilities, many of which are exploited by malicious adversaries in the context of cyber attacks. It is maintained by the National Institute of Standards and Technology, an agency of the United States Department of Commerce. The dataset is freely available for download (https://nvd.nist.gov/vuln/data-feeds).

Three of the selected studies cite NVD datasets. NVD datasets contain software vulnerabilities used by Peiró, Munoz, and Crespo (S8) and Peiró et al. (S21) to present a technique for detecting kernel-based information leaks through static analysis. Similarly, Österlund et al. (S5) work used NVD datasets to create a multivariant execution (MVX) approach against information leak vulnerabilities in the kernel by running multiple diversified kernel variants simultaneously on the same machine. These studies report intrinsic limitations in their approaches to identifying

vulnerability. The models only consider information leaks at a single function level, not covering information leaks involving multiple functions. Thus, each type of information leak must take different approaches to detect leaks. On the other hand, the authors recommend NVD datasets as they are updated continuously with new information leaks.

4.6. CAIDA. This dataset created by the Centre for Applied Internet Data Analysis (CAIDA) (https://www.caida.org/data) contains packets received by the UCSD Network Telescope between February 2001 and November 2008 alongside a collection of responses to spoofed traffic sent by DoS victims. This dataset is unlabeled, anonymized, and includes multiple-attack scenarios including misconfiguration, scanning of the address space by attackers or malware looking for vulnerable targets, and the automated spread of malware.

This dataset was cited by Shiravi et al. (S18) and Ahmed et al. (S28) who used the CAIDA dataset to create efficient infoleaks detection models and compared the results obtained with the models against other datasets, as DARPA-99, KDD-99, and DEF CON. According to these authors, the dataset is strongly anonymized with the payload wholly removed, resulting in less utility for researchers.

4.7. CTU-13. The CTU-13 dataset contains network traffic that was captured at the Czech Technical University in Prague (CTU) (https://www.stratosphereips.org/datasets-ctu13) in 2011. The dataset contains a large capture of real botnet traffic mixed with normal traffic and background traffic. The dataset consists of thirteen scenarios of different botnets samples (Rbot, Virut, Sogou, Neris, Menti, NSIS, and Murlo). This dataset includes a label for each instance, and its capture process was carried out in a controlled environment. Botnet traffic is usually created/captured in a controlled environment. The process to capture botnet traffic originates with the infected hosts, while the normal traffic is from verified normal hosts and the background traffic is derived from any other traffic of the network.

This dataset, which includes a large volume of attack patterns, was used to show that the approach proposed by Shin et al. (S17) identifies different attacks with high precision, with maximum F-measure of 93.80%. According to Yavanoglu and Aydos (S33), the advantage of using this dataset is that it is carefully labeled and captures processes conducted in a controlled environment. Finally, these studies indicate the importance of creating new datasets and making them publicly available.

4.8. NSL-KDD. This dataset was developed to address the KDD-99 criticism, namely, that it contains a large amount of redundancy. The number of records in the NSL-KDD training and test sets is sufficient for most of its intended usages for anomaly detection, as it contains about 150,000 data points. Its advantage is that the dataset does not include redundant or duplicate records, which can cause bias in machine learning classifiers. It was created by the Canadian Institute for Cybersecurity (CIC), based at the University of New Brunswick in Fredericton, and accessible on the web page (https://www.unb.ca/cic/datasets/nsl.html). This dataset is labeled, nonanonymized, and includes simulated intrusions in a controlled network environment: DoS attacks, privilege escalation (R2L and U2R), and probe.

The approach proposed by Idhammad et al. (S27) was tested on the NSL-KDD dataset and obtained satisfactory results, with 98.23% accuracy. The authors claim that they chose this dataset because it solves some of the intrinsic issues of its predecessors KDD-99 and DARPA-98, mainly the redundancy and duplication of records.

According to Ahmed et al. (S28), this dataset is the most suitable for solving some of the difficulties inherent to the intrusion detection problem, such as low detection rates for new attack types. This is partly because many public datasets are obsolete (some more than ten years old) and are not up to date with the new types of attacks that appear regularly.

4.9. DARPA-00. MIT Lincoln Laboratory created this intrusion detection dataset for the Wisconsin Rethink meeting and the July 2000 Hawaii PI (principal investigators) meeting, a part of the DARPA Strategic Information Assurance (SIA) program. This dataset was created with comprehensive examples of attacks and background traffic to help researchers and it is available for access on the web page (https://www.ll.mit.edu/r-d/datasets). This dataset is labeled, nonanonymized, and includes several types of attacks including DoS attacks, remote-to-local (R2L), user-to-root (U2R), and probes performed by an attacker used in multiple auditing and networking sessions. This final attack consists of five phases: (i) investigate the network; (ii) breaks into a host; (iii) exploiting vulnerability; (iv) installs trojan software; and (v) DDoS attack on an external server from the host server committed.

Shin et al. (S17) use DARPA-00 because it is one of the most widely cited by researchers. According to Buczak and Guven (S20), reusing the same dataset permits a comparison of the accuracy between different approaches.

4.10. ISCX-12. This dataset was created by Information Security Centre of Excellence (ISCX) at the University of New Brunswick (UNB) and contains captured real traffic for HTTP, SMTP, SSH, IMAP, POP3, and FTP. It is one of the most effective datasets in terms of realism, evaluation capabilities, total capture, completeness, and malicious activity. The dataset includes normal and anomalous traffic from seven days of network activity and can be downloaded from the UNB's web page (https://www.unb.ca/cic/datasets/ids.html). This dataset is labeled, nonanonymized, and includes four attack scenarios: infiltrating the network from the inside, HTTP DoS, DDoS using an IRC botnet, and a brute-force attack of SSH credentials. A description of each one of these attacks is detailed in Section 6.

The dataset UNB ISCX-12 was cited in the selected studies by Idhammad et al. (S27) and Ahmed et al. (S28),

who used them to validate the performance of the proposed method and compare it with the latest generation DDoS detection methods.

4.11. ADFA. This dataset covers both Linux and Windows operating systems and was created for the evaluation of Host-Based Intrusion Detection System (HIDS). The dataset (https://www.unsw.adfa.edu.au/unsw-canberra-cyber) includes selected attacks focused on the methods of contemporary penetration testers and hackers, where legitimate programs are operated as usual. This dataset is labeled, nonanonymized, and includes different attack scenarios including exfiltration, DDoS, user-to-root (U2R), and brute-force.

The dataset is cited in the studies of Ahmed et al. (S28) and Yavanoglu and Aydos (S33) who claim that ADFA is one of the best candidates to replace the DARPA and KDD-99 datasets as it uses a modern Linux operating system and updated exploits. This dataset also covers both Linux and Windows operating systems and is projected for evaluation by a system call-based Host Intrusion Detection Systems (HIDSs). HIDS acts in the detection and prevention of intrusions, using the behavior and history of the computer's data traffic on which it is installed.

4.12. CSIC-10. The dataset is available at The Spanish National Research Council (CSIC) (https://www.isi.csic.es/dataset) which is one of the most renowned institutions of the European Research Area (ERA). The dataset contains HTTP/1.1 packets, including web application penetration testing with two labels (normal and anomalous). In the dataset, there are three types of attacks: static, dynamic, and unintentional illegal requests. The dataset is nonanonymized and includes attacks such as SQL injection, buffer overflow, information gathering, files disclosure, CRLF injection, XSS, server-side include, and parameter tampering.

Yavanoglu and Aydos (S33) used this dataset to focus specifically on solutions that use HTTP protocols for connecting clients with servers. According to the authors, for convenience, the dataset is separated into three different subsets: training, anomalous, and testing. This division makes it easier to compare the results with other approaches.

4.13. DEF CON. The DEF CON dataset consists of several datasets used during a capture the flag (CTF) competition, which took place during the DEF CON conference. Teams used the dataset to defend their network while trying to break into opponents' networks. The dataset contain attack traffic and user behavior exclusively. The website is updated annually with new data from CTF competitions (https://www.defcon.org). Even though events in this dataset are considered different from the regular traffic on a network, it is used to identify attacks such as DDoS, botnet, user-to-root (U2R), and brute-force.

Ahmed et al. (S28) affirmed that the data present in this dataset differ from real network traffic because it consists mainly of intrusive traffic and is generally used for alert correlation techniques. The authors reinforce the need for the tracing of prelabeled data. As the DEF CON datasets are not labeled, they are less useful for researchers.

4.14. PKDD-07. This dataset is composed of samples that are superficially similar to real attacks but cannot succeed because they are constructed blindly and do not target the correct entities. Each sample contains at least one of the following attack types: cross-site scripting, SQL injection, LDAP injection, XPATH injection, path traversal, command execution, and server-side include (SSI) injection). Each sample is entirely independent of the others and contains a unique ID which will include three major parts: context, class, and the description of the query itself. The dataset is nonanonymized and is described in extensible markup language (XML). It was created by the Laboratory of Computer Science, Robotics and Microelectronics of Montpellier (LIRMM) and is available on the institution's web page (http://www.lirmm.fr/pkdd2007-challenge).

According to Yavanoglu and Aydos (S33), this dataset is suitable to study several different types of attacks in a heterogeneous and adversarial network environment. According to the authors, this dataset has the essential characteristics to compare the results with other approaches: dataset divided into training and tests, labeled, and not anonymized.

4.15. Enron. This dataset (https://www.cs.cmu.edu/~enron) was collected and prepared by the CALO Project (a Cognitive Assistant that Learns and Organizes). It contains data from approximately 150 users, mostly senior management of Enron Corporation, organized into folders. The dataset does not include attachments and is unlabeled, and parts of the text in some messages have been deleted to protect the privacy of employees. The dataset is valuable, and it is the only substantial collection of real emails that is in public access.

This dataset was used in the selected study by Shu et al. (S30) to detect complex data leak patterns and inexact sensitive data patterns. This detection process is paired with a comparable sampling algorithm, which allows one to compare the similarity of two separately sampled sequences. Besides, it is the only dataset with real emails made publicly available. Even though the data are anonymized in compliance with GDPR, its contribution to information leak detection is recognized.

4.16. ICS. The Oak Ridge National Laboratories (ORNL) have created this collection of nominated Industrial Control System (ICS) Cyber Attack Datasets. It contains three datasets, including standard electric transmission, disturbance, control, and cyber-attack behaviors. The dataset consists of sensors, logs from Snort, a simulated control panel, and relays. The dataset is available at the website of the Electrical & Computer Engineering (ECE) Department at Mississippi State University (MSU) (http://www.ece.msstate.edu). This dataset is labeled, anonymized, and

available in ARFF, CSV, and other formats. Ahmed et al. (S28) used this dataset to map different types of data leaks with the major types of attacks based on the analysis of the characteristics. According to the authors, labeled datasets favor the comparison of results with other approaches. Also, it is one of the newest publicly available dataset.

4.17. ISOT. The Information Security and Object Technology (ISOT) dataset (https://www.uvic.ca/engineering/ece/isot/datasets) is the combination of existing publicly available malicious and nonmalicious datasets. The malicious traffic originates from the honeynet project and consists of data from the Storm and Waledac botnets. Nonmalicious traffic was obtained from Traffic Lab Ericson Research in Hungary and from the Lawrence Berkeley National Lab (LBNL). This compilation contains general traffic from numerous types of applications besides HTTP, SMTP, and FTP for training and evaluation purposes. This dataset is labeled and nonanonymized and is available in pcap format.

Yavanoglu and Aydos (S33) affirmed that they opted for the ISOT dataset because of its volume and contains data from various botnets. ISOT is one of the largest publicly available datasets, with general traffic from multiple types of applications besides HTTP web browsing, traffic from the game World of War craft, and traffic from Bit Torrent clients.

4.18. Kyoto. This dataset (http://www.takakura.com/Kyoto_data) contains traffic data from Kyoto University's honeypots. It consists of 24 statistical features: 14 standard features extracted from KDD-99 dataset and 10 additional features which can be used to investigate what kinds of attacks happened on computer networks. The dataset is anonymized, is labeled, and contains DoS attacks, remote-to-local (R2L), user-to-root (U2R), and network probe.

Ahmed et al. (S28) used the Kyoto dataset to compare their results with other approaches. According to the authors, this dataset has the advantage of ignoring redundant features present in the KDD-99. Also, it captures the behavior of existing real networks.

4.19. LBNL/ICSI. This dataset is composed of traces recorded from a medium-sized site, publicly released in anonymized form, and spans a wide range of dimensions. The data have been anonymized in such a way as to remove any information which could identify an individual IP. It was created by the Lawrence Berkeley National Laboratory and is available on the International Computer Science Institute (ICSI) web page (http://www.icir.org/enterprise-tracing).

The ICSI dataset was used by Shiravi et al. (S18) to develop a systematic approach to the creation of datasets using real traffic of HTTP, SMTP, SSH, IMAP, POP3, and FTP. The fact that it is anonymized and unlabeled, it diminishes its importance for researchers.

4.20. PREDICT. This dataset (https://www.impactcybertrust.org/) also known as IMPACT Cyber Trust supports developers and researchers by providing regularly updated network data to cybersecurity research, including indicators on a wide variety of topics from 40 advanced and emerging countries. The dataset is non-anonymity, is labeled, and contains attack types like botnet (Ares), cross-site scripting, DoS, DDoS, heartbleed, infiltration, SSH brute-force, and SQL injection.

Ahmed et al. (S28) used the PREDICT dataset due to its regular updates and because it is considered one of the most complete in cybersecurity research. PREDICT is one of the public datasets, along with DEF CON and NVD that is updated continuously.

5. RQ2B: Can Datasets Be Useful to Detect Data Leaks after Following GDPR Guidelines?

According to the selected studies, the authors state that it would need to be correctly labeled, constantly updated, have realistic traffic data, and not be anonymized for a dataset to be considered relevant (S18, S20, S24, S28, S33). Unfortunately, as shown in Table 4, only the dataset PREDICT covers all these characteristics.

Datasets with emulated traffic do not accurately reflect what occurs in the real world. Even creating a controlled environment using regular real-world network traffic in parallel with attack scenarios, it is unlikely to generate false-positives, which ultimately reduces the accuracy of the models. On the other hand, depending on the volume of data captured, real-world traffic can present other difficulties including wrongly assigned labels and unbalanced classes. However, most researchers agree that datasets with realistic traffic are better, even if the time and workforce's investment needed to create them are more significant.

Anonymized data may complicate the analysis of network-based data sets. On the other hand, it is essential to know that some types of data must follow GDPR rules concerning the protection of users' data. Therefore, it is necessary to carefully evaluate which attributes must be discarded and which datum can be published anonymously. Datasets that are totally or partially anonymized decrease their utility to researchers.

For the dataset to be usable in supervised machine learning algorithms, the dataset must be labeled. This feature is also desirable when comparing the results of different approaches with each other. According to our SLR, the research community agrees on the importance of labeled datasets.

An important point regarding the use of public datasets is that some are out of date for today's vulnerabilities and attacks. Of course, current systems must be prepared to handle all known attack types, as attackers exploit all possible weaknesses. Besides, a common practice for attackers is to make subtle changes to the signatures of botnets, rootkits, malware, and viruses to deceive data leak detection systems.

Finally, we found that researchers have remarked on the paucity of publicly available datasets. This absence of new

Table 4: Datasets according to the principal characteristics.

-	Realistic traffic	Anonymized	Labeled	Constantly updated	Various types of attack
CAIDA	No	Yes	No	No	No
ICS	No	Yes	Yes	No	Yes
DARPA-98, DARPA-99, DARPA-00, KDD-99, UNSW-NB-15, NSL-KDD, ISCX-12, ADFA, CSIC-10, PKDD-07, ISOT	No	No	Yes	No	Yes
NVD	No	No	Yes	Yes	Yes
Enron, LBNL/ICSI	Yes	Yes	No	No	No
CTU-13, KYOTO	Yes	Yes	Yes	No	Yes
DEF CON	Yes	No	No	Yes	Yes
PREDICT	Yes	No	Yes	Yes	Yes

datasets is pointed out in a study by Sommer and Paxson [13] as one of the great open challenges for anomaly-based intrusion detection. With the increase of research in this field, additional data leak detection datasets and improvements can be expected.

6. RQ3: What Are the Different Types of Attacks That Exist?

According to Bridges et al. [1], attackers typically seek to access, modify, or delete confidential information for financial gain, fame, personal revenge, or to disrupt organizational services. Attackers exploit software vulnerabilities, the high workload and inexperience of employees, and the heterogeneity of security solutions implemented in an organization to perform their attacks.

In the current security context, organizations must develop strategies that allow them to be resilient in the face of malicious attacks. Security mechanisms create layers of protection and generate event logs that can be analyzed to detect and react to possible intrusions or malfunctions. By studying the logs, security analysts can detect and respond to attacks as they occur, rather than through the application of forensic investigation weeks or even months after an incident. A careful analysis of the security logs may reveal the attacks or even the identity of the culprit. However, the considerable size of modern security logs and the heterogeneous nature of logs originating from different systems require that such an analysis be scalable and grounded on solid theoretical footings.

While the attack can be initiated by either an internal or external attacker, internal attackers are widely seen as the most dangerous, in part because an external attacker must perform several complicated steps before executing an attack. In contrast, the internal attacker knows the systems, the vulnerabilities he can exploit, and privileged access and can be motivated by revenge or financial gain.

Leaks of personal information are often considered the most important cause for concern since they result in an invasion of privacy, unexpected and undesired by users. After obtaining personal information, the attacker can build detailed user behavior profiles and leverage this profile to extort money, guess credentials, or gain other personal advantages.

A substantial number of information leaks result from human errors; for example, a missing or stolen laptop containing unencrypted sensitive information or an employee transmitting confidential data without using an end-to-end encryption program.

The problems caused by this type of behavior can be severe, and companies must create internal mechanisms and policies that reinforce security. Investing in internal auditing and monitoring tools to detect threats are minimal requirements needed to ensure information security. This same concept is codified in information security management practices such as ISO/IEC 27001 [14] and ISO/IEC 27002 [15]. Following these best practices, institutions must develop and establish an information security process to allow and enable its proper functioning, identifying the assets it is intended to protect.

Several different types of attacks have occurred in recent years, the most common type being the DoS and DDoS (17 out of 33 studies), followed by unauthorized accesses (12 studies) and injection attacks (7 studies).

The different attack types reported in the studies in our sample set are listed and classified in Table 5 and are divided into categories and subcategories. The category "unclear or not specified" includes cases where the attack was not specified and could not be inferred without the authors' interpretation (S16, S25, S26). In what follows, we briefly describe each category of attack.

6.1. Application. Attacks in this category are designed for specific applications, the most common being Web servers, including database servers, websites, and Internet of Things (IoT) devices.

6.1.1. Code Injection. Code injection attacks are part of an entire class of attacks that rely on injecting data into a web application in order to execute or interpret malicious data in an unexpected way [16]. According to OWASP's top 10 vulnerabilities (https://owasp.org/www-project-top-ten/), injection attacks are most common and successful on the Internet due to their numerous types, large attack surface, and the complexity sometimes required to protect them. Injection failures occur when untrusted data are sent to an interpreter as part of a command or query. Hostile data from the attacker can cause the interpreter to execute unintended commands or access unauthorized data.

Code injection is one of the most common and increasingly dangerous attack types (S1, S4). These attacks

Table 5: Types of attacks.

Categories	Attacks	References	Datasets	Log types
Application	Code injection	S1, S4, S13, S15, S23	CAIDA	AP, WS
	Cross-site scripting	S15, S19, S20, S23, S33	CSIC-10, PKDD-07, PREDICT	WS
	Kernel	S5, S8, S21	NVD	SY
	Microarchitectural attacks	S3	—	SY
	Information flow attack	S11	—	AP, WS
Authentication	Malicious insiders	S9, S19, S30	CAIDA	AP, NT, SY, WS
	Session hijacking	S9, S19	—	AP, NT, WS
Credential	Brute-force	S2, S9, S18, S29	ADFA, ISCX-12, DEF CON, PREDICT	AP, AU, NT, VM, WS
	Keylogger	S9, S14, S19	—	AP, AU, NT, SC, VM, WS
	Man-in-the-middle	S9, S19, S23	—	AP, AU, NT, ST, VM, WS
	Phishing	S13, S19	ISCX-12	AP, NT, WS
Network	Botnets	S9, S13, S17, S18, S20, S26, S33	CAIDA, CTU-13, DEF CON, ISOT, LBNL/ICSI, PREDICT	NT, WS
	DNS attacks	S6, S7, S9, S19	CSIC-10, PKDD-07, CAIDA	NT, WS
	DoS and DDoS	S9, S10, S12, S13, S15, S17, S18, S19, S20, S22, S24, S27, S28, S29, S31, S32, S33	DARPA-99, KDD-99, DARPA-98, UNSW-NB-15, NSL-KDD, DARPA-00, ISCX-12, CAIDA, ADFA, ICS, ISOT, KYOTO, LBNL/ICSI, PREDICT	NT, WS
	Reused IP address	S9, S19	CSIC-10, PKDD-07	NT, WS
	Network sniffing	S7, S9, S19	CAIDA	NT, WS
Physical	Theft of equipment	S14	—	—
	Unauthorized accesses	S14, S15, S18, S20, S22, S24, S28, S29, S30, S31, S32, S33	—	—
Virtualization	Hypervisor Rootkit	S19	—	VM
	VM Escape	S19	—	VM
Unclear or not specified	S16, S25, S26	—	—	

AP = application log, AU = audit log, NT = network log, SC = security log, ST = setup log, SY = system log, VM = virtual machine log, WS = web server log.

typically take advantage of the fact that the server assumes that the client's request is valid and well-formed, and accordingly, does not perform sufficient validation (S23). According to Ullah and Babar (S13), the injection of erroneous data can incur disastrous consequences in terms of attack detection, for example, by allowing computer malware to disseminate. Attack is used to steal user information by inserting malicious code into the Web application as user input (S15, S20). The malicious code could be injected into SQL or NoSQL queries, operational system commands, XML parsers, SMTP headers, etc. The objective is to insert malicious code that changes the common usage of the computer program and modifies the ordinary course of execution. According to Iqbal et al. (S19), attackers insert or inject malicious code into a web page, occasioning web pages infected by malware when Internet users browse the page. This type of malware may disturb the functionality of web services and pages (S9, S17). It may also compromise the availability of Internet connections (S33).

6.1.2. Cross-Site Scripting. Cross-site scripting (XSS) is an attack-type in which an adversary exploits a vulnerability present in web applications to insert code (as JavaScript) and obtain certain types of advantage over victims. It is usually used on pages common to all users, such as a website's homepage or even pages where users can leave their comments. For the attack to occur, the page must contain a form that allows the attacker to interact with the system, such as a search field or a field for entering comments [17].

Cross-site scripting is an application layer attack technique (S33) that injects malicious scripts into a web application to gather information from a different machine (S15, S19, and S23). According to Buczak and Guven (S20),

50% of the attacks on web applications discovered in 2009 were of this type.

6.1.3. Kernel Attacks. This type of attack exploits vulnerabilities in the kernel drivers to gain privileges (S8). It is considered one of the most dangerous vulnerabilities and has become the expert adversaries' tool of choice. Several malware attacks, including Derusbi, RobbinHood, Sauron, Uroburos, and GrayFish (which exploit vulnerabilities detailed in the NVD datasets), leveraged kernel driver vulnerabilities to gain kernel privileges and effectively disable protection, compromising machines and systems.

According to a study by Österlund et al. (S5), most large-scale projects are written in insecure languages, and the kernel can contain four main classes of exploits: memory corruption, policy violation, denial of service, and information leak. Peiró, Munoz, and Crespo (S8) and Peiró et al. (S21) show that this type of vulnerability allows access to the layout and contents of the kernel memory, allowing the attacker to bypass the kernel-level protections and read sensitive kernel data.

6.1.4. Microarchitectural Attacks. This emerging type of attack exploits speculative execution by leaking secret information along miss-speculated paths via cache-based channels and others. Due to their ability to direct the execution of the control flow path, these attacks have the potential to break all confidentiality and completely ignore important hardware/software security mechanisms. The attack consists of inserting a process on the victim's machine and obtaining access to the shared cache memory. According to Taram et al. (S3), this type of attack can reveal secret information, such as cryptographic keys, key loggers, and browsing activities.

6.1.5. Information Flow Attack. This type of attack captures sensitive or confidential information that should not be exposed externally. In other words, information flow attacks occur when an adversary can deduce confidential information by observing the relation between private inputs a public outputs (S11). According to Vorobyov, Krishnan, and Stocks (S11), this type of attack is used in several contexts because it relates information flows to furtive attack scenarios where an opponent may bypass detection.

6.2. Authentication. Authentication attacks aim at gaining access to resources without obtaining the appropriate credentials. This can usually be accomplished through a privilege escalation. Privilege escalation is the reconnaissance of a programming error, vulnerability, wrong setup, or any kind of gained unauthorized access to resources that are typically limited from the application or user.

6.2.1. Malicious Insiders. Malicious insiders can be employees, ex-employees, or business partners who have genuine access to systems and data but use these privileges to destroy, steal, or sabotage [18]. This attack category does not include well-intentioned employees who accidentally put the cybersecurity of the enterprise at risk or leak data because of human error, lack of technical skills, or inexperience. In particular, malicious insiders are dangerous because they have already overcome enterprise defenses, have sensitive data in hand, and are aware of the network's weaknesses, which help them perform their misdeeds.

One of the biggest threat vectors that occur when an employee intentionally deploys malicious code to capture sensitive information (S9, S30). According to Iqbal et al. (S19), insider attacks are a major threat but do not attract much attention because companies commonly focus their attention to external threats. According to Nkosi et al. [18], only after being attacked by insiders, companies begin to pay more attention to their collaborators, implementing information security controls. Also, malicious internal attacks are generally planned in advance.

6.2.2. Session Hijacking. Session hijacking is synonymous with a stolen session, in which an attacker intercepts and takes over a legitimately established session between a user and a host [17]. The user-host relationship can apply any authenticated resource, such as a web server, or another TCP-based connection. Attackers stand between the user and the host, allowing them to monitor the user's traffic and launch specific attacks. Once a successful session hijacking occurs, the attacker can assume the legitimate user's role or monitor the traffic to inject or collect specific packages to create the desired effect. In possession of the user's session, the attacker impersonates them and performs any action that falls within the privileges of the user (S9 and S19).

6.3. Credential. Credential attacks occur when a third party is trying to gain access to systems by cracking a user's password, typically by running a software on their side. Many methods can be used to access accounts including brute-force attacks to guess passwords, as well as comparing various word combinations against a dictionary file. None of the datasets we reviewed contain attacks of this category.

6.3.1. Brute-Force. This attack occurs when an attacker tests a large amount of passwords in order to discover the victim's credentials and thus be able to access his account or system [16]. Various types of brute-force attacks exist, such as credential stuffing and the reverse brute force attack. Brute-force attacks are generally more successful in cases where weak or relatively predictable passwords are used.

This attack aims to gain authorized access through credentials that are discovered because they use a weak username and password combination (S2, S9, S18). Brute-force attacks commonly use a dictionary of common passwords and exhaustively try them all (S29). This pattern of user behavior facilitates the use of techniques such as brute-force and account hijacking.

6.3.2. Keylogger. Keyloggers are a type of surveillance technology used to monitor and record each keystroke typed on a computer's keyboard, thus retrieving the username/password to gain access to systems and other applications (S14). A keylogger can be based on hardware or software and is useful as a legitimate personal or professional information technology monitoring tool (S9, S19). On the other hand, a keylogger can also be used for criminal purposes. Typically, this type of record is made by malicious spyware used to capture sensitive information, such as passwords or financial information, which is then sent to third parties for criminal exploitation.

6.3.3. Man-in-the-Middle. This type of attack occurs when a hacker interposes himself in the communication between a user and another party, such as a bank website, login e-mail, or social networks (S9, S19), intercepting outgoing messages and impersonating one of the parties involved [16]. When an attacker participates in the communication between two parties, he can tamper with or block the information without the victims noticing it. A common example is in intercepting communication between a user and his bank to steal sensitive information. This information can then be put to use for the benefit of the hacker. In a study by Jung et al. (S23), the use of encrypted connections is suggested to increase the work factor of a potential adversary.

6.3.4. Phishing. Phishing is the act of deceiving people into sharing confidential information such as passwords and credit card numbers (S13).For example, a victim might receive an e-mail or text message that impersonates a different person or organization they trust, such as a friend, a company, or a government agency [16]. When the victim opens the e-mail or the text message, they are directed to imitating a legitimate website. From there, the victim feels safe to login with his username and password. In this way, access information is sent to thieves who use it to steal identities, steal bank accounts, and sell personal data on the black market. A study by Iqbal et al. (S19) defines phishing as a type of social engineering attack aimed at obtaining the most critical data from customers or users, such as financial records information or other personal information.

6.4. Network. Network attacks are unauthorized actions that aim to destroy, modify, or steal confidential data. There are two principal types of network attacks: (i) passive, where sensitive information is screened and monitored, potentially compromising the security of the company and their clients, and (ii) active, where data are modified or destroyed entirely.

6.4.1. Botnets. A bot is a program capable of propagating automatically, exploiting existing vulnerabilities or flaws in the configuration of software installed on a computer. Botnets are networks formed by computers infected with bots (S18). These networks can be composed of hundreds or thousands of computers. An attacker who has control over a botnet can use it to increase the power of its attacks, for example, by sending hundreds of thousands of phishing or spam emails or by launching denial of service attacks.

According to Shiravi et al. (S18), botnets are a sophisticated attack vector against workstations and networks. Initially, attackers gain access to a network by way of a Trojan, which allows them to execute code remotely (S9, S13, S17). Botnets are one of the most studied attacks and a significant amount of public data are available about them (S20, S26, S33).

6.4.2. DNS Attacks. If the attacker can alter domain name system (DNS) servers to incorrectly resolve name resolution queries and redirect users to malicious websites [16], then a DNS attack becomes possible. This way, when the victim's device queries a domain, the malicious DNS responds with the IP address of a fraudulent website, different from the original (S6, S7, S19). This type of attack's main objective is to redirect the victim to fraudulent sites, which are mostly phishing, and thus obtain users, passwords, credit cards, and all types of confidential information. This attack is also used to redirect the user to advertising sites or display adware, which generates an economic gain for the attacker. Such an attack can compromise the availability of the underlying servers (S9).

6.4.3. Denial of Service and Distributed Denial of Service (DoS and DDoS). These two categories of attack are used to prevent legitimate users from accessing a desired network or Internet service. This is usually performed by overloading the target (almost always an Internet server) with a huge amount of traffic or sending malicious requests causing the attack target to work unwantedly or fail (S9, S10, S17, S18, S19, and S33). Some Denial of Service (DoS) attacks seek to hinder specific people's access to their networks or databases, while others try to make these resources totally inaccessible. These attacks can last from minutes to hours, and in some rare situations, even for days. They can cause large financial loss for companies that become targets and have no effective strategies to combat the practice.

The attackers send several packet requests to the target to make it so overloaded that it can no longer respond to any packet request. Thus, users are no longer able to access computer data because it is unavailable and unable to respond to any request. In a Distributed Denial of Service attack (DDoS), a master computer can direct up to millions of computers, called zombies, and coordinate their behavior during the attack. Thus, all zombies act in unison and uninterruptedly request the same resource from the target (S28). Considering that web servers can only serve a limited number of users at the same time, this large number of traffic makes it impossible for the server to fulfill any request. The server may restart or even hang, depending on the resource that was victimized. According to (S12), this type of attack increased by 40% in 2018, with more than 400,000 monthly attacks that year. This attack can be detected by monitoring network traffic volume between source and destination IPs (S13, S15, S22, and S31). It is

considered the most used attack and the first option to compromise a network (S20, S24, S29, and S32) and a significant Internet threat (S27).

6.4.4. Reused IP Address. According to Guillén et al. [16], security vulnerability is present if a user leaves a networks and a subsequent user receives the same IP address. This occurs because the new users may be able to access the resources of the user to whom this IP address was previously assigned (S9, S19).

6.4.5. Network Sniffing. This is a process by which a malicious intruder can capture and analyze all of the data packets containing sensitive information passing through given network traffic (S7 and S19). Sniffers can be hardware or software installed in the system. This program captures packets that arrive at the computer's network interface and allows for analysis of them [16]. Sniffing activities aim to gain legitimate access or steal information (S9). Because some network applications transmit data in text format (HTTP, FTP, SMTP, POP3, etc.), through a sniffer, the cybercriminal can find useful and sometimes confidential information (for example, user namesl,and passwords). Intercepting usernames and passwords is very dangerous because users often use the same username and password for various features and applications.

6.5. Physical. Physical attacks are offensive actions that aim to obtain unauthorized access to tangible assets such as infrastructure, hardware, or interconnection. This threat type usually applies to any kind of infrastructure or equipment. None of the datasets we studied contain attacks of this category.

6.5.1. Theft of Equipment. According to the Ponemon Institute [19], 63% of small and medium-sized businesses suffered a data leak incident in 2019, the main cause being the loss or theft of equipment occur when cybercriminals obtain a device, facilitating access to the internal network since the user's systems are already properly configured for use. Most people do not understand that an apparently innocent action like using a pen drive can result in corporate data theft by cybercriminals. Theft or loss of a corporate notebook that is not encrypted, for example, can compromise sensitive company data.

According to Sarkar (S14), theft of equipment could cause multiple problems including lost work hours, loss of intellectual property, loss of customer data, business disruption, sabotage, loss to individuals, or reputational damage. Based on a survey by Verizon [20], physical theft is an episode where an information asset was lost, whether through misplacement or malice. According to the same survey, 34% of attacks involved internal actors, 71% of breaches were financially motivated, and 56% of breaches took months or longer to discover.

6.5.2. Unauthorized Accesses. This attack class refers to the entry into a website, server, computer, service, or even as physical facilities of the company by an unauthorized party (S14). Before the actual incident, it is common for an attacker to perform a simulation with the objective of assessing the security of the network and identify its vulnerable points.

Based on the study by Cheng et al. (S24), this attack refers to unauthorized access or attempting access to disrupt or damage the network or steal information. The growth of networked machines and the use of the Internet in organizations increase in the number of unauthorized activities, not only by external attackers but also by internal sources, abusing their privileges for personal gain (S14, S15, and S22). Sensitive data need to be protected and cannot be exposed to unauthorized parties. There are different ways for companies to protect themselves, notably encryption, access controls, and steganography (S30 and S33). Many studies focus on creating processes for detecting unauthorized access (S18, S20, S28, S29. S31, and S32).

6.6. Virtualization. Virtualization is the abstraction of a hardware or software system that allows applications execute on top of the virtualized environment. Attackers try to obtain access to the hypervisor, which controls all the virtual machines (VMs) running in the data center or cloud. There is no specific dataset that contains security logs for virtualized environments.

6.6.1. Hypervisor Rootkit. A rootkit is a set of malicious tools that allows a cybercriminal to access privileged software areas on a machine while hiding its presence. In this way, the hypervisor rootkit can intercept communications or requests on the operating system's hardware and host [21].

According to Iqbal et al. (S19), virtualization is a new attack vector with an increasing number of rootkit tools. This attack could compromise the entire virtualized environment, and malicious clients can access the virtual machines environment. As a result, the attackers could obtain administrative privileges and get access to all the VMs.

6.6.2. VM Escape. This attack exploits a security vulnerability in hypervisors. Through this vulnerability, an attacker can execute malicious code inside a virtual machine and execute commands on another VM or even on the underlying system [21]. As a result, it may control any resource of the VM on the host system (S19).

6.7. Discussion of the Types of Attacks. As can be seen in Table 5, we identified 20 types of attacks. The attacks in the application category exploit standard interfaces to access the server and perform incorrect and harmful actions. These attacks are based on the fact that the usual server assumes acceptable client requests. This situation is especially prevalent for consumer devices with software that is generally released with little regard to the requirements of the secure development life-cycle.

The primary objective of the attacks in the authentication category is to allow unauthorized users to access the company's systems and data. Attackers exploit vulnerabilities in the mechanisms and methods used to authenticate the system to hack and gain privileges. This type of attack occurs because many users are still using the simplest username and password to secure their accounts. Other factors include deficiencies of internal controls, lack of updating and maintenance of equipment, and absence of an information security policy which leaves loopholes that attackers exploit.

The credential attack category contains various methods of breaking into a password-protected computer or server. This type of attack occurs because many computer users insist on using ordinary words as passwords, being rarely successful against systems that employ multiple-word phrases and random combinations of uppercase and lowercase letters mixed up with numerals. Attacks as brute-force attacks can sometimes be useful, but this technique can spend a long time to obtain results. A user's training is essential, being one of the most effective defenses against social engineering tactics.

Attacks in the network category try to compromise the system or render network resources unavailable to users. With the increase of software as a service offered on the Internet, the potential to explore vulnerabilities also rises. Attackers can target the entire infrastructure or focus on a selected portion, using, for example, botnets, or DNS attacks. The anonymity of the Internet also raised difficulties with respect to attack attribution. This is also the category of attack most heavily studied in the literature, and in the studies present in our sample, most large companies have deployed a variety of mechanisms to protect themselves against attacks in that category.

In the case of the physical category, the selected studies indicate that different types of losses and thefts occur, which ends up generating significant threats. Another situation is when an adversary gains unauthorized access and expose the company's confidential data. Even as companies are creating increasingly complex mechanisms to enforce their security policies, the attackers are developing new strategies to circumvent their security. Among the problems detected, we highlight (i) misuse of privileged information; (ii) absence of strict internal controls regarding the access policy to the facilities; (iii) employees who use the company's resources privately and personally; and (iv) adoption of remote work, allowing employees to carry confidential data on laptops, smartphones, and flash drivers that can be lost or stolen, compromising the data.

Finally, in the virtualization category, the attackers try to take control of the operations of hosted virtual machines (VMs), initiating the attack on other hosted VMs or another hypervisor. According to Iqbal et al. (S19), virtualized systems are more vulnerable compared to nonvirtualized. As a result, the entire virtualized environment could become compromised. Attackers can get the administrative privileges of the virtualized environment and obtain access to all the VMs. Also, the attackers can use VMs to launch spoofing attacks by changing the targeted content.

7. RQ4: What Are the Main Algorithms Used in the Analysis of Logs to Detect Information Leaks?

In the final step of this SLR, we identify all algorithms commonly used in the literature for the detection of information leaks and highlight their main characteristics. It is important to note that our study shows how each algorithm was used in each approach of the selected studies. Table 6 summarizes the algorithms used for cyber-intrusion detection. In particular, we compare the algorithms used for data leak detection, datasets used for experiments, and problem domain in studies relevant.

7.1. Adaboost. Adaptive Boosting, or simply Adaboost, is an algorithm that consists of sequentially combining several weaker models. Its operation consists of each model initialized with a standard weight, which defines its decision power in the final model [22]. As each model is training, each weak learner gains a greater weight for the observations that he correctly predicts and a lower weight for the observations in which he has a high error rate. In this way, weak learners with greater precision will have greater decision power in the final model, producing an extremely robust ensemble learning model.

According to Buczak and Guven (S20), AdaBoost is an algorithm used to address the overfitting problem (omnipresent in security and attack prevention), improving the generality of the predictive model [23]. The authors affirm that AdaBoost uses boosting to train several weak learning algorithms and match their weighted results. A weak learner is one that consistently generates better predictions than random. Also, the authors affirm that they obtain a higher detection percentage while having a lower false-positive rate. Yavanoglu and Aydos (S33) use the Adaboost algorithm on the CSIC-10 HTTP dataset to validate the correctness of their model.

7.2. Apriori. The Apriori algorithm is one of the most widely used method for mining frequent patterns in databases. The algorithm can extract sets of frequent items, and some procedures can be performed to obtain association rules from these sets [24]. The algorithm works by performing an iterative process, where each iteration basically performs two functions: (i) to generate possibly frequent candidate item sets and (ii) to define which requested item sets are really frequent.

Shinet et al. (S17) use the Apriori algorithm to create a new data leak detection method free from the drawbacks of a decoy-based authentication scheme. They measured detection accuracy on the CTU-13, and DARPA-00 datasets showed satisfactory performance, obtaining the maximum F-measure of 0.938 for the detection of Botnets using the CTU-13 dataset. In another study, Buczak and Guven (S20) use the Apriori algorithm to discover multistage attack patterns. Their work uses the DARPA-99 and DARPA-00 datasets. According to the authors, they were able to detect 93% of the attacks in 20 seconds.

Table 6: Algorithms used to analyze security logs.

Algorithm	References	Dataset	Problem domain
Adaboost	S20, S33	CSIC-10, PKDD-07	Injection attacks
Apriori	S17, S20	CAIDA, DARPA-98, DARPA-99, DARPA-00, NSL-KDD -	DoS and DDoS
Belief propagation (BF)	S6	—	DNS attacks
C4.5	S20, S26	CAIDA, CTU-13, DEF CON, ISOT, LBNL/ICSI, PREDICT	Botnet detection
Classification and regression tree (CART)	S26	—	Injection attacks
Co-clustering	S27, S28	ISOT	Botnet detection
Extreme learning machine (ELM)	S24	DARPA-99	DoS and DDoS
Fuzzy logic	S19, S20, S29, S32	DARPA-98, DARPA-99, KDD-99	DoS and DDoS
Graph-based semisupervised learning	S6	—	DNS attacks
GraphSAGE	S6	—	DNS attacks
Influence and diffusion	S6	—	DNS attacks
Iterative dichotomiser 3 (ID3)	S20	DARPA-98, DARPA-99, KDD-99	DoS and DDoS
J48	S13, S32, S33	KDD-99, CSIC-10, PKDD-07	Injection attacks
K-means	S12, S13, S26, S28, S33	CAIDA, PKDD-07	DoS and DDoS
K-nearest neighbour (KNN)	S7, S9, S26, S31, S32, S33	CAIDA, CSIC-10, PKDD-07, KDD-99	DoS and DDoS
Levenberg-Marquardt algorithm	S1	—	Code injection
Multilayer perceptron (MLP)	S1, S26	—	Injection attacks
Multivariate adaptive regression splines (MARS)	S20	DARPA-98, DARPA-99, KDD-99	Injection attacks
Naïve bayes	S13, S20, S26, S28, S33	CSIC-10, PKDD-07	Injection attacks
Ordered binary decision diagrams (OBDDs)	S4	—	Code injection
Principal component analysis (PCA)	S15, S28, S33	CSIC-10, PKDD-07	Injection attacks
Random walk with restart (RWR)	S6	—	DNS attacks
Randon forest	S13, S20, S26	CAIDA, CTU-13, DEF CON, ISOT, LBNL/ICSI, PREDICT	Botnet detection
Repeated incremental pruning to produce error reduction (RIPPER)	S22	DARPA-99	DoS and DDoS
REPTree	S13	KDD-99	DoS and DDoS
Sequence alignment algorithms	S16, S23, S30	—	Malicious insiders
SimRank	S6	—	DNS attacks
Spectral clustering	S12	—	Security visualization
Support vector machine (SVM)	S1, S13, S19, S20, S24, S26, S28, S32, S33	CAIDA, DARPA-98, DARPA-99, DARPA-00, NSL-KDD, PKDD-07, ISOT	DoS and DDoS
TF-IDF	S6	—	DNS attacks

7.3. Belief Propagation (BP). BP is a type of message-passing algorithm in which the messages are probability functions representing confidence about the value of the codeword bits. The probabilistic values can be represented as the log-likelihood ratio (LLRs), allowing calculations to be made using the sum and product [25]. While probabilities need to be multiplied, LLRs need only be added, reducing the problem's complexity.

The Belief Propagation algorithm is a graph-based inference algorithm used for threat detection (S6). It infers a node's label using knowledge about that node and other neighboring nodes by iteratively transmitting messages between all pairs of nodes in the graph. The authors affirm to have achieved 95% accuracy with the proposed approach, while the belief propagation algorithm reached 89%. The authors' approach, entitled MalRank, is a graph-based inference algorithm able to infer a node's maliciousness score based on its relationships with other entities shown in the knowledge graph, for example, shared IP addresses or hostname.

7.4. C4.5. C4.5, as other decision tree algorithms, use the "divide and conquer" method that employs a top-down search to denote the tree's possible paths concerning the example provided [26]. The selection of which attributes should be associated with the decision node is made using the information gain criterion. The gain measures how much an attribute is capable of separating a set of examples into categories. Whoever has the highest gain is selected to be included in the tree.

Buczak and Guven (S20) affirm that the C4.5 algorithm is one of the best-known methods for automatically building decision trees from a set of training data using the concept of information entropy. The authors obtained 97% accuracy in the C4.5 decision tree. However, their false-positive rates were high at 8.05%, using data collected from 18 wireless campus locations over four months. When building the decision tree, at each node of the tree, C4.5, chooses the attribute of the data that most effectively splits its set of examples into subsets (S26). The splitting criterion is the

normalized information gain (difference in entropy). The attribute with the highest normalized information gain is selected to make the decision. The C4.5 algorithm then executes recursion on the smaller subsets until all the training examples have been classified.

7.5. Classification and Regression Tree (CART). The CART algorithm divides the training examples in the spectral space into rectangular regions, also known as nodes, and assigns a class to each region. The process begins with all classes distributed throughout the spectral space and examines all the possibilities of binary divisions in each characteristic that makes up the spectral space. Subsequently, a subdivision is selected if it has the best optimization criterion [27].

The CART algorithm is used by Sjarif et al. (S26). The study reviews the techniques used in Endpoint Detection and Response (EDR) and tools, a state-of-the-art cybersecurity technology. The authors state that CART can be used when the number of models is dependent on the training set.

7.6. Co-Clustering. An alternative to the grouping task is co-clustering, which can extract differentiated information from the data matrix compared to the information extracted with grouping [28]. In the case of co-clustering, the similarity criteria are applied simultaneously to the rows and columns of the data matrices, simultaneously grouping the objects and attributes.

Idhammad et al. (S27) proposed a semisupervised DDoS detection approach based on entropy estimation using the co-clustering algorithm. The authors conducted various experiments to validate the performance of the proposed method using three public datasets, namely, the UNB ISCX-12, the NSL-KDD, and the UNSW-NB-15. They obtain results for accuracy and false-positive of 99.97% and 0.33%, 98.41% and 0.33%, and 66.28% and 0.34% on datasets UNB ISCX-12, NSL-KDD, and UNSW-NB-15, respectively. Also, according to the authors, the results, in terms of accuracy and false-positive rate, are satisfactory when compared with other state-of-the-art DDoS detection methods. According to Ahmed et al. (S28), co-clustering can be considered as a dimensionality reduction approach, and it is adequate for creating new features.

7.7. Extreme Learning Machines (ELM). Of the various types of neural networks in existence, feedforward networks are the most popular, being the single hidden layer type especially known for functioning as universal approximators [26]. Among them, the ELM stands out due to its fast training period, and its essence is the use of random synaptic weights in the hidden layer. This configuration results in a linear model for the synaptic weights of the network's output layer (output weights), which are analytically calculated using a least-squares solution. Although this network offers a good generalization capacity, the random choice of the synaptic weights of the hidden layer (input weights) can generate a nonoptimal set of weights and thresholds, causing the effect of overfitting. Another problem faced is the choice of the number of hidden neurons and the possibility of decreasing it without affecting learning effectiveness, resulting in several tests by trial and error.

A study by Cheng et al. (S24) proposed an intrusion detection system in a computer network using extreme learning machines (ELMs) to classify and detect the intrusions. The dataset used is from the DARPA-98 intrusion detection program and obtained 99.66% of accuracy after 50 trails, using 95% confidence interval. The authors affirmed that, for DDoS attack detection, the basic ELM algorithm with sigmoid additive neurons would be a good option since it has a significantly smaller training time compared to other techniques.

7.8. Fuzzy Logic. Unlike the Boolean logic that admits only the values "true" or "false", in Fuzzy Logic, a premise varies in degree of truth from 0 to 1, which leads it to be partially true and partially false. With the incorporation of the "degree of truth" concept, fuzzy sets theory extends the general set theory. The groups are qualitatively labeled (using linguistic terms such as tall, warm, active, small, close, etc.), and the elements of this set are characterized by varying the degree of belonging (value indicating the degree to which an element belongs to a set) [28].

According to Iqbal et al. (S19), the common types of Intrusion Detection System (IDS) are based on the rule-based technique and detect the intrusion behavior of the network traffic using fuzzy logic and data mining techniques. Approaches using Fuzzy Logic variations can be used to detect accidental information leaks due to human errors or application flaws (S29). It is used in many applications, mainly in data leak detection (S32). Buczak and Guven (S20) obtain results of 100% accuracy and 13% of false-positive using the KDD-99 dataset for anomaly detection. This study also reports other advantages of this algorithm such as human-comprehensible rules, easier handling of nominal attributes, and efficient classification on large datasets.

7.9. Graph-Based Semisupervised Learning. It is an algorithm based on the structure of a graph. Each sample is represented by a vertex in the weighted graph that measures the similarity between the samples. The main step in the algorithm is to build a better graph to represent the original data structure, which can be separated into two steps: (i) build a graph of all marked and unmarked samples and (ii) propagate the labeled information to the unlabeled samples through the graph [29].

The use of this algorithm requires a major adjustment to solve the problem of unlabeled data (S6). Unlabeled data can be utilized to decide the metric between data points and improve the model's performance. Unfortunately, it exhibits a high computational overhead. Additionally, according to the authors, this algorithm requires significant adjustment to support the main requirements of the proposed approach by them.

7.10. GraphSAGE. Given a partially labeled graph G, the GraphSAGE algorithm leverages the labeled nodes to predict the unlabeled nodes' labels. The GraphSage algorithm uses an embedding learning solution for each node inductively to determine the labels [30]. Specifically, each node is represented by the aggregation of its neighbors. In this way, even if a new node not seen during the training stage appears on the graph, it can still be represented correctly by your neighbors' nodes.

This algorithm uses neural networks to learn embeddings for nodes in the graph composition while getting aggregated features from a node's local neighborhood. According to Najafi et al. (S6), it requires multiple adjustments before it can be adequately used in threat detection. Some adjustments include changing the edge weights depending on the source, i.e., if a node is malicious, the edge weights connecting to that node should be increased.

7.11. Influence and Diffusion. Social networks allow broad sharing of information, messages, or opinions. The representation of these users is performed, in most cases, in the form of graphs, where each user is a node (vertex) and their interactions in the network are edges [31]. Thus, users on social networks can be represented as a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges representing how the nodes are related. Information diffusion and influence algorithms can measure influential nodes for different types of networks. This importance of a node in a network is calculated using metrics imported from graph theory.

According to Najafi et al. (S6), these algorithms are simple and need a significant setting to be used, i.e., directional and weighted edges, echo cancellation, and influence maximization. They were originally designed to study the influence and diffusion in social networks but are adaptable to the problem of information leaks detection.

7.12. Iterative Dichotomiser 3 (ID3). ID3 is one of the most well-known and simple algorithms for learning using decision trees [26]. It builds the decision tree from the root, selecting the best classifying attribute among all the dataset attributes. The best classifying attribute is selected based on a statistical evaluation of all attributes. After the choice, the data are separated according to the chosen attribute classes, generating a subdivision of the data for each descendant in the tree. The algorithm is applied recursively to each descendant until some stopping criterion is reached. This generates an acceptable decision tree, in which the algorithm never goes back to reconsider choices made previously.

According to Buczak and Guven (S20), the advantages of ID3 as decision trees are intuitive knowledge expression, high classification accuracy, and simple implementation. The main disadvantage is that for data, including categorical variables with a distinct number of levels, information gain values are biased in consequence of features with more levels. Experiments using DARPA-99 were also performed by raising the number of rules from 150 to 1581. According to the authors, this increasing speed-up of the decision tree algorithms can substantially reduce the processing time in the detection system.

7.13. J48. J48 is an algorithm that can handle both discrete and continuous attributes and categorical and missing values [26]. The treatment of continuous attributes involves considering all the values present in the training set, causing them to be sorted in an increasing way considering all the values present in the training data and, after this ordering, the value that will favor the reduction of the entropy. Also, the J48 algorithm has the advantage of not requiring accurate data.

Studies by Shah et al. (S32) and Yavanoglu and Aydos (S33) use the J48 algorithm to compare accuracy results with other machine learning techniques. Like other decision tree algorithms, J48 is simple and very powerful. It proceeds by first selecting attributes and then classifying the data, creating IF, THEN, and ELSE rules. According to a study by Ullah and Babar (S13), in terms of training time and decision time, the J48 algorithm is one of the most efficient for detecting cyberattacks targeting web applications. The authors compare six algorithms (J48, Naïve Bayes, Random Forest, SVM, Conjunctive rule, and RepTree). The RepTree was confirmed as the most efficient followed by J48.

7.14. K-Means. K-means is an algorithm that groups data based on their similarity. In k-means, it is necessary to specify the number of groups in which we want the data to be grouped [28]. The algorithm randomly assigns each observation to a cluster and finds the centroid of each cluster. The algorithm then follows two steps: (i) redistribute data points to the cluster whose centroid is closest and (ii) calculate a new centroid for each cluster. These two steps are repeated until the variation within the set can no longer be reduced. The cluster's variation is calculated as the sum of the Euclidean distance between the data points and the respective cluster centroids.

According to Najafi et al. (S12), the firewall log is one of the primary sources for visualizing intrusions, so a visualization tool has been proposed to help analysts with cluster resources to detect anomalies. The authors also affirmed, after comparing several studies, that since cybersecurity problems and challenges evolve; monolithic visualization methods cannot give a full solution. They suggest that the cybersecurity field needs configurable and interoperable visualization solutions for finding new patterns. According to Ullah and Babar (S13), Sjarif et al. (S26), and Yavanoglu and Aydos (S33), K-means is normally used to detect cyberattacks, DDoS, and flooding attacks. A study by Ahmed et al. (S28) used five variations of clustering and distance-based anomaly detection using the K-means algorithm. The best result obtained 80.15% of accuracy and 21.14% of false-positive with the NSL-KDD dataset.

7.15. K-Nearest Neighbour (KNN). The KNN algorithm is based on instance learning. This means that the training data is stored, and the classification of a new item is carried out by

comparing the similarities of the item to be classified with those of the test data [26]. The implementation of the KNN algorithm can be performed through the Euclidean distance, where the smaller the Euclidean distance between two instances, the more similarities these instances will have. Therefore, the lower the value of the Euclidean distance, the greater the efficiency in classifying a new instance, which will receive the nearest neighbor's classification as a classification.

According to Tall et al. (S7), the KNN algorithm has the characteristic of applying locality and sensitivity to cluster log or threat data which facilitates the identification of outliers. For example, nearest-neighbor analysis to clustering of activity within an IP address range, verification of workflow patterns related to time frames to identify unusual off-hour activity, or recognition of traffic size outliers may provide indicators of data leaks. Cotroneo et al. (S9) detect anomalies by means of a KNN classifier by measuring the frequency of the system calls. Other selected studies which used the KNN algorithm include Sjarif et al. (S26), and Shah et al. (S32, S33) reports it being well applicable to the domain of intrusion detection. A study by Liao and Vemuri (S31), using the DARPA-98 dataset, obtained 77.3% accuracy and a 0.59% false-positivity. According to this study, although the KNN algorithm has a lower attack detection rate, its performance is higher than other machine learning methods; even this result may not hold against a more sophisticated dataset.

7.16. Levenberg-Marquardt Algorithm. The Levenberg-Marquardt algorithm is an iterative technique that finds the minimum value of a function expressed as the sum of the squares of real values of nonlinear functions. Its application is necessary due to its ability to accelerate the convergence process. The acceleration of the training is based on the determination of the second-order derivatives of the quadratic error concerning the weights, differing from the traditional backpropagation algorithm that considers the first-order derivatives [32].

Eliseev and Gurina (S1) affirm that any methods of supervised learning can be applied for neural and opted to use the Levenberg-Marquardt algorithm. They proposed an embedded intrusion detection system which can be used in industrial automation systems (M2M) and Internet of Things (IoT). The study experiment was conducted using network traffic collection from real servers, investigating the HTTP protocol. The tests obtained 98.62% accuracy with the correct classification of normal traffic.

7.17. Multilayer Perceptron (MLP). A perceptron is a linear classifier, that is, it is an algorithm that classifies the input separating two categories with a straight line. An MLP is an artificial neural network composed of more than one perceptron [26]. They are composed of an input layer to receive the signal, an output layer that decides or predicts the input. Between these two, an arbitrary number of hidden layers are the true computational mechanism of MLP. MLPs with a hidden layer can approximate any continuous function.

Due to the equality between input and output after training, the multilayer perceptron (MLP) is capable of spotting similar inputs making almost the same output and be used for data leak detection (S1). For the experiment, the authors collected real server traffic over HTTP and HTTPS protocols intending to reduce the delay between the appearance of uncharacteristic behavior for the server and the discovery of this fact. The study by Sjarifm et al. (S26) presents a review of the main machine learning techniques used to detect attacks, and the MLP is one of the most applied. The internal structure of MLP allows to remind during training the most general features of training vectors. Unfortunately, neural network training is a resource-intensive process.

7.18. Multivariate Adaptive Regression Spline (MARS). This approach is a statistical method that tries to approximate complex relationships by a series of linear regressions at different intervals of the independent variable's limits or the independent variable's sub regions of space [33]. It is very flexible since it can adapt to any functional form. MARS builds a relationship from a group of coefficients (base functions) that are entirely determined by the regression data. For approximation, MARS uses a truncated two-sided function, also known as the iterative forward-backward approach.

The MARS algorithm is used by Buczak and Guven (S20) to create an intrusion detection system utilizing a subset of the DARPA-98 dataset. According to the authors, accuracy of 99.71%, 99.85%, 99.97%, 76%, and 100% was reported for Normal traffic, Probe or Scan, DoS, U2R, and R2L categories, respectively.

7.19. Naïve Bayes. It is a classification technique based on Bayes' theorem with the assumption of conditional independence among the predictors. In simple terms, a Naïve Bayes classifier assumes that the presence of a particular characteristic in a class is not related to the presence of any other characteristics [26]. After completion of the model, we obtain a list of probabilities that can be used to make the classification of new data based on its characteristics.

Using the KDD-99 dataset, a study by Ullah and Babar (S13) affirms that the Naïve Bayes classifier obtained the best training time compared to other machine learning algorithms such as SVM and Random Forest. Using the KDD-99 dataset, the algorithm Naïve Bayes presented the best training time with 79.5 s, while the Support Vector Machine (SVM) algorithm produced the worst with 479.12 s. As for accuracy, the results were 91% for Naïve Bayes and 78% for SVM, respectively. For Buczak and Guven (S20), the advantage of the Naïve Bayes is that it is an online algorithm, and its training can be finished in linear time. Because datasets and security logs tend to have many attributes, Naïve Bayes can handle an arbitrary quantity of continuous or categorical independent attributes. Naïve Bayes can decrease a high-dimensional density estimation task to a one-dimensional estimate, supposing that attributes are independent. In general, Naïve Bayes classifiers have considerably outperformed even highly advanced classification techniques (S26). Just a small

quantity of training data is needed to estimate specific parameters (S28, S33).

7.20. Ordered Binary Decision Diagrams (OBDDs). OBDDs are forms of the canonical representation of Boolean formulas. They are built through algorithms that transform a Boolean function into a binary decision tree that is then optimized to eliminate redundancy [34]. Consequently, OBDDs are more compact and can be handled more efficiently. Each state is encoded by a designation of Boolean values for the set of state variables associated with the system. Thus, a transition relationship can be expressed as a Boolean formula in terms of two sets of variables: encoding the old state and the other encoding the new one. A binary decision diagram then represents this.

Yang et al. (S4) proposed an approach to perform submatch extraction. It has been used with real network traces, enterprise event logs, and synthetic traces to show excellent performance when patterns are combined with capturing groups in regular expressions [35]. Finite automata are natural representations for regular expressions. According to the authors, the experiments are faster than Google's RE2 (http://swtch.com/rsc/regexp/) and PCRE (http://www.pcre.org) approaches by one to two orders of magnitude using the SNORT dataset. Still, they do not show the precision results, only showing the execution time (cycles/bytes) and memory consumption. Since the SNORT dataset is not publicly available, it is not part of the datasets described in Table 3.

7.21. Principal Component Analysis (PCA). PCA is a multivariate analysis technique that can be used to analyze interrelationships between many variables and explain these variables in terms of their inherent dimensions (components). The goal is to find a way to condense the information contained in several original variables into a smaller set of statistical variables (components) with a minimal loss of information. It is one of the most used methods for reducing dimensionality [36].

Landauer et al. (S15) use PCA to identify anomalies in high-dimensional message count vectors and additionally consider state variables for filling the matrix. The authors created a set of evaluation criteria and investigated 59 approaches for detecting specific types of anomalies in the log data. They used the PCA algorithm to classify, identify relations, and plot the results. According Ahmed et al. (S28) and Yavanoglu and Aydos (S33), data leak detection using PCA has the benefits: (i) free from any assumption of statistical distribution; (ii) able to reduce the dimension of the data without losing any vital information; and (iii) has minimal computational complexity which makes it an apt choice for real-time data leak detection.

7.22. Random Walk with Restart (RWR). The algorithm provides the proximity between two nodes on the graph [37]. The first solution evaluated is randomly chosen. In the next iteration, three factors determine the next solution to be evaluated: the size of the step or displacement, the dimension of the problem, and the direction of the step.

This graph-based algorithm has been used for threat detection in Najafi et al.'s (S6) works. The RWR algorithm is implemented to simulate random walkers executing steps into a graph while having a small probability of transporting to a random node, rather than following an out-edge. The authors obtained 95% accuracy with the proposed approach, entitled MalRank, a graph-based inference algorithm designed to deduce a node's maliciousness score based on its relationships with other entities. The authors did not compare the result of the proposed approach with this algorithm.

7.23. Random Forest. Random forests (RF) consist of a collection of decision trees. Each tree is constructed from a random vector's values, which is sampled independently and with uniform distribution for all trees in the forest [26]. The objective is to combine N decision trees' individual classifications, each constructed with F attributes, in a single label. The number of classification trees to be built and the number of dimensions randomly chosen per node (number of attributes) strongly impact an RF classifier's accuracy and computational effort. Random forests are computationally very effective, in addition to avoiding over fitting and being less sensitive to noise.

As reported by Ullah and Babar (S13), using the NSL-KDD dataset, optimal accuracy is achieved with Random Forest, with 82.3%, while Support Vector Machine (SVM) obtained the worst accuracy rate with 37.8%. If the training data are scarce, according to a study by Buczak and Guven (S20), the Random Forest algorithm might have an advantage because it is a classifier method that combines the decision trees and ensemble learning. A study by Sjarif et al. (S26) shows some benefits of the Random Forest algorithm: (i) measured as highly accurate and robust because of the number of decision trees associated in the process; (ii) does not have the overfitting problem because it takes the average of all the predictions, which cancels out the biases; and (iii) can work with missing values.

7.24. Repeated Incremental Pruning to Produce Error Reduction (RIPPER). RIPPER is an algorithm based on decision rules, which analyzes the set of instances and generates rules in the logical format, which commonly combine attributes to more appropriately capture the classes of instances [38]. Its operation is based first on the choice of the majority class as a standard of comparison. In a second phase, the algorithm tries to discover the rules to detect minority classes, optimizing the set of initial rules to decrease errors and make the process more selective. This algorithm is especially suitable for inducing models from data sets with an unbalanced class distribution. It also works well with noisy data sets because it uses a validation set to avoid overfitting the model.

Pietraszek (S22) affirms that the RIPPER algorithm is a fast and effective rule learner. RIPPER presents two main advantages when employed for intrusion detection: (i) it exhibits good general accuracy and concise conditions and

(ii) it is efficiency with noisy datasets. The author obtained 0.10% of false-positive using the DARPA-99 dataset to the classification in batch mode.

7.25. RepTree. RepTree consists of a fast decision tree learning algorithm based on the calculation of information gain using the entropy method to minimize the error resulting from the variance during the classification process. The model uses gain/variance information to build a regression tree and prune to reduce errors, taking into account the back fitting method [39].

Ullah and Babar (S13) reported the testing of six machine learning algorithms to compare their training time and decision time of each. The algorithms are J48, Naïve Bayes, Random Forest, SVM, Conjunctive rule, and RepTree. RepTree was confirmed as the most efficient followed by J48.

7.26. Sequence Alignment Algorithms. The sequence alignment algorithms measure the similarity between each pair of symbols in the compared sequences using a scoring system, which can incorporate information about the domain of knowledge during the comparison [40]. These algorithms belong to the group of approximate matching algorithms. That is, they allow some differences between the expected sequence and the observed sequence. The sequences' alignment is guided by a scoring system, which can add semantic aspects to the search.

Studies by Jung et al. (S23) and Ekelhart et al. (S16) used sequence alignment algorithms to better match semantic needs and to be able to deal with large datasets to data leak detection efficiently. These two studies claim that the alignment algorithms can detect data leaks but only present the vulnerabilities of applications that expose personal information and, consequently, violate the GDPR guidelines. In a study by Shu et al. (S30), sequence alignment techniques were used to detect complex information leaks at long and inexact sensitive data patterns. The approach devised by the authors, named AlignDLD, is capable of detecting all modified e-mail leaks of the ENRON dataset, with a threshold of 0.2, i.e., it results in 100% recall.

7.27. SimRank. The SimRank algorithm is a graph-based metric that can be applied across multiple domains. It is a recursive definition resulting from the similarity of the neighbors of two vertices i and j. i and j are more likely to relate if their neighbors have characteristics in common. This concept of a vertex's neighborhood comprises all others that can be reached from the initial vertex (that is, there is a path between them) [41].

In a graphical structural context, two objects are seen as similar if associated with similar objects, a concept that can intuitively be applied to measure the influence. According to Najafi et al. (S6), this metric has high computational complexity, making it impossible to calculate the knowledge graph.

7.28. Spectral Clustering. The Spectral Clustering algorithm is a generalization of standard clustering methods, designed to work in situations where clusters are not just circular or elliptical in shape. The main idea is to build a matrix that represents the local neighborhood relations on the observations. There are several ways to define the similarity matrix and its graph representing local behavior, with K-nearest-neighbor being the most popular. It is also necessary to calculate the Laplacian matrix, which can be non-normalized or normalized [42].

According to Najafi et al. (S12), spectral clustering is used to scale to large IP address spaces visually and is appropriated to handle IPv6 because it has much larger IP spaces than IPv4. The authors also affirm that IPv6 is a 128 Bit IP Address, unlike IPv4 is a 32-Bit IP Address, the algorithm clustering maintains performance visually scale to large IP address spaces. The study does not present results, only reporting that the algorithm is used in other visualization approaches, and helps analysts interpret firewall log events.

7.29. Support Vector Machine (SVM). In its simplest form, a linear form SVM combines a hyperplane so that there is a margin separating a set of positive and negative examples in a space with many dimensions [26]. Given that there may be infinite choices for the margin that separates these examples, the goal is to maximize the distance from that margin to the nearest negative and positive examples. It is impossible to linearly separate all cases from a data set in some cases, so no hyperplane divides all positive and positive points. In this case, a penalty applies to an example that fails to position itself on its correct margin.

Eliseev and Gurina (S1) developed a method using the SVM algorithm, which was capable of considering the individuality of correlation functions for different requests processed by a server. The authors used collected traffic over the HTTP and HTTPS protocols to evaluate the proposed approach but did not show results. However, the authors affirmed they implemented a lightweight attack detection model. According to Ullah and Babar (S13), Iqbal et al. (S19), Buczak and Guven (S20), Cheng et al. (S24), and Sjarif et al. (S26), SVM generates more accurate results but is computationally expensive in terms of accuracy and performance. This computational cost occurs because the traditional SVM involves solving a quadratic program, which is a quadratic order of the training sample size. A study by Ahmed et al. (S28) shows different approaches using the SVM algorithm to run as an unsupervised learning algorithm where it tries to divide the entire set of training data from its origin. In contrast, the regular supervised SVM attempts to separate two classes of data in feature space by a hyperplane. Finally, the results of the studies of Shah et al. (S32) and Yavanoglu and Aydos (S33) show that SVM is able to detect anomalies with acceptable accuracy by searching a small number of nodes.

7.30. Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF is a weight often used in information retrieval and text mining. This weight is a measure used to assess how important a word is for a document in a text collection (the corpus) [43].

The weight increases proportionally to the number of times a word appears in the document but decreases proportionally to the frequency of the word in the corpus.

Najafi et al. (S6) implemented a large-scale graph-based inference algorithm designed to infer maliciousness using the associations presented in the knowledge graph. They compared his results with Zhang et al. [43] who used the term frequency/inverse document frequency (TF-IDF) algorithm to tackle malicious URL detection. The authors did not compare the result of the proposed approach with this algorithm.

7.31. Discussion. According to Table 6, our SLR has identified thirty algorithms that have been successfully used to detect information leaks. There is no single algorithm capable of detecting all types of attacks, so hybrid approaches are commonly used to detect anomalies and threats.

When developing an information leak detection model, the researcher or professional may want to use a wide variety of algorithms. In this sense, the last column of Table 6 reports the problem domains in which the algorithm was used. The algorithms are not exclusive to each of the presented domains, and they can be applied in other domains. According to the public datasets used in the selected studies, the algorithms showed good general detection capabilities, which means that they offer balanced coverage against known attacks' most possible manifestations.

Finally, in our analysis, we observed that most algorithms depend on actions, such as determining the number of KNNs, selecting attributes, normalizing the data, or creating graphs. Thus, to mitigate this problem, in some scenarios for detecting different types of attacks, it is possible to adopt ensemble methods that use multiple learning algorithms to obtain better predictive performance. Other techniques, such as cross-validation and approaches to making a balanced dataset from an unbalanced one using undersampling and oversampling techniques, can be applied since most public datasets are highly unbalanced.

8. Threats to Validity

We assessed the quality of our SLR using the checklist of Petersen et al. [3]. We performed 17 of the 26 actions in the checklist and obtained a score of 65.38%. Our rating is higher than most reported SLR, with a median of 33% and a maximum value of 48%. However, threats to validity are inevitable. In what follows, we report the main threats to the validity of our study and how we address them.

8.1. External Validity. In order to find studies representative of state of the art on the use of security logs for data leak detection, we performed searches in four academic search engines (ACM, ScienceDirect, IEEE Xplore, and Scopus) and performed both backward and forward snowballing on the selected studies. We considered only peer-reviewed papers and excluded the so-called grey literature (e.g., white papers, editorials, etc.). Furthermore, we applied well-defined and previously validated inclusion and exclusion criteria. This decision may have caused us to miss some relevant studies. We chose to use these four databases since we considered them the most relevant ones to the topic of this SLR. Furthermore, our results provided evidence that can guide further investigations.

8.2. Internal Validity. Early in this study, the inclusion and exclusion criteria were defined according to the researchers' judgment. The first selected studies formed the basis for identifying the most commonly used keywords about information leaks and for the elaboration of the Systematic Literature Review (SLR) protocol. Therefore, most of the keywords used to represent the information leaks and security logs were included. A threat to validity exists if the initial selection of the keywords was incomplete. We hope that snowball techniques can mitigate this threat.

8.3. Construct Validity. The research questions we defined may have reduced the number of relevant studies on information leaks. In fact, we systematically performed data extraction within the specified criteria and retained the most relevant studies. To ensure the reliability of the research process, the first author applied the research process and the second author periodically cross-checked the search results.

8.4. Conclusion Validity. Incorrect data extraction impairs the classification and analysis of the selected studies. The selected studies were evaluated by the researchers to ensure a common understanding. Two researchers read each article, and a third researcher was involved in the discussion as needed. We mitigate possible threats to conclusion validity using the best practices of three different guidelines on systematic studies [2–4].

As described in Kitchenham and Charters and Petersen et al., both systematic mapping studies (SMS) and systematic literature reviews (SLR) have the same methodologies for conducting search and data extraction. One difference between these two guidelines is their final goal. Specifically, the RQs in an SMS are more general; they identify research trends and detect topics within the field [3]. On the other hand, SLR tries to collect data from primary studies and subsequently answer the RQs [2]. Following the recommendation of Webster and Watson (2002), we applied the snowballing technique with retrospective research (identifying and examining the study's references) and future research (identifying and considering studies that mention a specific paper) [4]. We applied these best practices at each step of the study and documented each step in a research protocol available to the public (https://bit.ly/ExtractionPapersDataLeaks). This transparency facilitates the study's replication by other researchers looking for relevant studies on the use of security logs for data leak detection.

9. Related Works

Security log data has been widely used to analyze and prevent cyber attacks and draw an increasing amount of attention from researchers and professionals. There have been a few literature reviews on data leaks from different perspectives.

Mahmood and Afzal [44] present a review of seven different types of cyber attacks and analytic systems used for data leak detection. They also identify the principal elements of a security analytic model, some examples of security analytic outputs, and steps for implementing a security analytic solution. On the other hand, our SLR features 20 types of attacks, 30 algorithms used to identify data leaks, and the 20 public datasets used to detect attacks.

Goldstein and Uchida [45] present a comparative evaluation of 19 algorithms used for data leak detection using ten different datasets from different areas. They also outline the algorithms' strengths and weaknesses concerning their usefulness for particular use scenarios. Their study serves as a guideline for selecting an appropriate data leak detection algorithm for a given task.

A study by Zoppi et al. [46] proposes a methodology based on analytical and experimental investigations to identify anomaly detection algorithms and attack families to define guidelines for the selection of unsupervised intrusion detection algorithms. The authors used four datasets (KDD-99, NSL-KDD, ISCX-12, and UNSW-NB-15) and one algorithm from each one following families: neighbor-based, clustering, angle-based, classification, density-based, and statistics. This work covers some unsupervised datasets and algorithms used to detect information leakage. Our study extends this contribution, identifying twenty datasets, thirty algorithms, and twenty-one types of attacks. Besides, we propose a new classification of information leaks that follows the GDPR principles.

A study by Ring et al. [47] identified fifteen characteristics of datasets covering a wide range of criteria. Based on these characteristics, the authors provided a comprehensive overview of existing datasets, highlighting each dataset's peculiarities. This study is similar to ours because it presented the essential characteristics for the creation of network-based datasets to detect data leaks. However, no mention was made on the importance of following the GDPR principles. In addition, the different types of attacks that can be identified in each dataset are mentioned but without describing each one.

An approach to classifying data leak threats based on their causes, whether intentionally or inadvertently leaking confidential information, was proposed by Cheng et al. [48]. The authors compare their approach to other methods based on the origin of the leak: internal or external. There is no reference in the study to the GDPR principles, with the authors focusing on data leak prevention and detection (DLPD).

Seo and Kim's [8] study proposes a classification of internal threats, dividing them into internal threats and external threats. The authors conducted a study of threats that were examined through the definition, classification, and correlation/association analysis of various human-machine records of acts associated with security breaches that occur in an organization. A quantitative process and decision-making tool were developed for internal threats by establishing various internal scenarios of information leakage. Even though a large study contributes metrics to evaluate internal threats, the proposed classification system does not meet the GDPR principles. Thus, companies that adopt these metrics are not able to meet the seven fundamental principles of GDPR.

The study that most closely relates to ours is Buczak and Guven's [49], which discuss several approaches to data leak detection using machine learning. Their study provides a summary of the essential algorithms, inclusive of the algorithmic time complexity. Some public datasets to conduct detection and detection methods are also discussed. This survey presents algorithms and datasets used in threat detection, but our SLR also discusses the different types of information leakage, types of attacks, and public datasets used to detect data leaks. However, it does not mention the importance of data privacy and compliance with GDPR principles.

The related works cited here mainly focus on using datasets and machine learning techniques to detect information leaks. However, there is no systematic literature review that presents the importance of data privacy and GDPR principles. Besides, our study covered security logs and public datasets, machine learning techniques, and types of attacks used in practice in academic literature. Therefore, this study aims to fill that gap.

10. Conclusion

The purpose of this study is to provide broader investigation on the relationships between research contributions in the use of security logs to detect anomalies and information leaks. We systematically reviewed 33 studies and produced a clear answer to five research questions.

The data set used for training and building the model can be considered one of the essential points, as the quantity and quality of the data influence the training result. Besides, according to the studies we surveyed, it must be correctly labeled, regularly updated, containing real traffic data, and not anonymized for a dataset to be considered relevant. Among the twenty described here, only the PREDICT dataset covers all of these characteristics. On the other hand, PREDICT does not contain every type of attack. The creation of data sets is considered an arduous task and requires several assessments and validation, but its contribution to this research field is significant.

Several algorithms have been used in data leak detection systems. In our study sample, no single algorithm or machine learning technique seems capable of detecting every type of attack. Therefore, efforts at improving the detection rate and reducing false alarms must be continuously improved. Since the challenge is to detect a rare event, an unconventional attack in a massive record of regular events, this can be considered a problem of unbalanced data. If a dataset is incorrectly labeled, it can create a PU learning

problem (positive and unlabeled, as someone can identify attacks in the records, but if it is not labeled as an attack, it will remain undiscovered) [50]. Likewise, when the dataset contains only a very small number of instances of a given class (as is the case for new attack types) creating the model can be especially challenging. This issue should be paid attention to if researchers are interested in creating labeled datasets.

The paper's main contributions are

(i) A new classification of information leaks, following the GDPR principles

(ii) A description of 20 public datasets used in anomaly detection

(iii) An explanation of 20 types of attacks present in public datasets

(iv) An Investigation of 30 algorithms used in threat detection

The results of this study will benefit researchers and professionals who wish to further contribute to the field and practitioners who want to understand existing research. Finally, we provide researchers with a comprehensive list of the main benefits of using security logs to detect anomalies and information leaks. We encourage professionals to evaluate their methods using several datasets, algorithms, and machine learning techniques to avoid overfitting, reduce the influence of non-labeled datasets, and identify different types of attacks and false alarms with accuracy.

Appendix

A.1. List of Selected Studies Included in the Review

(S1) V. Eliseev and A. Gurina, "Algorithms for network server anomaly behavior detection without traffic content inspection," in Proceedings of the 9th International Conference on Security of Information and Networks, pp. 67–71.

(S2) S. Khan, A. Gani, A. W. A. Wahab, M. A. Bagiwa, M. Shiraz, S. U. Khan, R. Buyya, and A. Y. Zomaya, "Cloud log forensics: Foundations, state of the art, and future directions," in ACM Comput. Surv., vol. 49.

(S3) M. Taram, A. Venkat, and D. Tullsen, "Context-sensitive fencing: Securing speculative execution via microcode customization," in Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 395–410.

(S4) L. Yang, P. Manadhata, W. Horne, P. Rao, and V. Ganapathy, "Fast submatch extraction using OBDDs," in Proceedings of the Eighth ACM/IEEE Symposium on Architectures for Networking and Communications Systems, pp. 163–174.

(S5) S. Österlund, K. Koning, P. Olivier, A. Barbalace, H. Bos, and C. Giuffrida, "KMVX: Detecting kernel information leaks with multi-variant execution," in Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 559–572.

(S6) P. Najafi, A. Mühle, W. Pünter, F. Cheng, and C. Meinel, "MalRank: A measure of maliciousness in SIEM-based knowledge graphs," in Proceedings of the 35th Annual Computer Security Applications Conference, pp. 417–429.

(S7) A. Tall, J. Wang, and D. Han, "Survey of data intensive computing technologies application to to security log data management," in Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, pp. 268–273.

(S8) S. Peiró, M. Munoz, and A. Crespo, "An analysis on the impact and detection of kernel stack infoleaks," in Logic Journal of the IGPL, vol. 24, pp. 899–915.

(S9) D. Cotroneo, A. Paudice, and A. Pecchia, "Empirical analysis and validation of security alerts filtering techniques," in IEEE Transactions on Dependable and Secure Computing.

(S10) D. Jaeger, A. Sapegin, M. Ussath, F. Cheng, and C. Meinel, "Parallel and distributed normalization of security events for instant attack analysis," in IEEE 34th Intern. Performance Computing and Communications Conference.

(S11) K. Vorobyov, P. Krishnan, and P. Stocks, "A low-overhead, value-tracking approach to information flow security," in Information and Software Technology, vol. 73, pp. 19–36.

(S12) I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An evaluation framework for network security visualizations," in Computers & Security, vol. 84, pp. 70–92.

(S13) F. Ullah and M. A. Babar, "Architectural tactics for big data cybersecurity analytics systems: A review," in Journal of Systems and Software, vol. 151, pp. 81–118.

(S14) K. R. Sarkar, "Assessing insider threats to information security using technical, behavioural and organisational measures," in Information Security Technical Report, vol. 15, pp. 112–133.

(S15) M. Landauer, F. Skopik, M. Wurzenberger, and A. Rauber, "System log clustering approaches for cybersecurity applications: A survey," in Computers & Security, vol. 92, p. 101739.

(S16) A. Ekelhart, E. Kiesling, and K. Kurniawan, "Taming the logs-vocabularies for semantic

security analysis," in Procedia Computer Science, vol. 137, pp. 109–119.

(S17) J. Shin, S.-H. Choi, P. Liu, and Y.-H. Choi, "Unsupervised multi-stage attack detection framework without details on single-stage attacks," in Future Generation Computer Systems, vol. 100, pp. 811–825.

(S18) A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," in Computers & Security, no. 3, pp. 357–374.

(S19) S. Iqbal, M. L. M. Kiah, B. Dhaghighi, M. Hussain, S. Khan, M. K. Khan, and K.-K. R. Choo, "On cloud security attacks: A taxonomy and intrusion detection and prevention as a service," in Journal of Network and Computer Applications, pp. 98–120.

(S20) A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cybersecurity intrusion detection," in IEEE Communications Surveys Tutorials, no. 2, pp. 1153–1176.

(S21) S. Peiró, M. Muñoz, M. Masmano, and A. Crespo, "Detecting stack based kernel information leaks," in International Joint Conference SOCO'14-CISIS'14-ICEUTE'14, pp. 321–331.

(S22) T. Pietraszek, "Using adaptive alert classification to reduce false positives in intrusion detection," in Recent Advances in Intrusion Detection, pp. 102–124.

(S23) J. Jung, A. Sheth, B. Greenstein, D. Wetherall, G. Maganis, and T. Kohno, "Privacy oracle: A system for finding application leaks with black box differential testing," in Proceedings of the 15th ACM Conference on Computer and Communications Security, pp. 279–288.

(S24) Chi Cheng, Wee Peng Tay, and G. Huang, "Extreme learning machines for intrusion detection," in The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–8.

(S25) H. Suryotrisongko and Y. Musashi, "Review of cybersecurity research topics, taxonomy and challenges: Interdisciplinary perspective," in 2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA), pp. 162–167.

(S26) N. N. A. Sjarif, S. Chuprat, M. N. Mahrin, N. A. Ahmad, A. Ariffin, F. M. Senan, N. A. Zamani, and A. Saupi, "Endpoint detection and response: Why use machine learning?," in 2019 International Conference on Information and Communication Technology Convergence (ICTC), pp. 283–288.

(S27) M. Idhammad, K. Afdel, and M. Belouch, "Semi-supervised machine learning approach for DDoS detection," in Applied Intelligence, pp. 3193–3208.

(S28) M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," in Journal of Network and Computer Applications, pp. 19–31.

(S29) X. Shu and D. D. Yao, "Data leak detection as a service," in Security and Privacy in Communication Networks, pp. 222–240.

(S30) X. Shu, J. Zhang, D. D.Yao, andW. Feng, "Fast detection of transformed data leaks," in IEEE Transactions on Information Forensics and Security, vol. 11, pp. 528–542.

(S31) Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," in Computers & Security, pp. 439–448.

(S32) A. A. Shah, M. S. H. Khiyal, and M. D. Awan, "Analysis of machine learning techniques for intrusion detection system: A review," in International Journal of Computer Applications, vol. 119, pp. 19–29.

(S33) O. Yavanoglu and M. Aydos, "A review on cybersecurity datasets for machine learning algorithms," in 2017 IEEE International Conference on Big Data (Big Data), pp. 2186–2193.

Data Availability

Data sharing is not applicable to this article as no data sets were generated or analyzed during the current study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. A. Bridges, T. R. Glass-Vanderlan, M. D. Iannacone, M. S. Vincent, and Q. Chen, "A survey of intrusion detection systems leveraging host data," *ACM Computing Surveys*, vol. 52, no. 6, pp. 1–35, 2020.

[2] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Tech. Rep. EBSE 2007-001, Keele University and Durham University Joint Report, Keele, UK, 2007.

[3] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: an update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[4] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future: writing a literature review," *MIS Quarterly*, vol. 26, no. 2, 2002.

[5] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14*, pp. 1–38, New York, NY, USA, 2014.

[6] A. Cárdenas, P. Manadhata, and S. Rajan, *Big Data Analytics for Security Intelligence*, pp. 1–22, Cloud Security Alliance, Seattle, WA, USA, 2015.

[7] D. A. Tamburri, "Design principles for the general data protection regulation (GDPR): a formal concept analysis and

its evaluation," *Information Systems*, vol. 91, Article ID 101469, 2020.

[8] S. Seo and D. Kim, "Study on inside threats based on analytic hierarchy process," *Symmetry*, vol. 12, no. 8, p. 1255, 2020.

[9] S. Alneyadi, E. Sithirasenan, and V. Muthukkumarasamy, "A survey on data leakage prevention systems," *Journal of Network and Computer Applications*, vol. 62, pp. 137–152, 2016.

[10] A. Shabtai, Y. Elovici, and L. Rokach, *A Survey of Data Leakage Detection and Prevention Solutions*, Springer Briefs in Computer Science, Springer, Berlin, Germany, 2012.

[11] B. Singman, "Dem group exposed millions of email addresses in 'clinton' file, firm reveals," 2020, https://www.foxnews.com/politics/dem-email-addresses-associated-with-clinton-senate-campaign-exposed.

[12] F. Pouget and M. Dacier, "Honeypot-based forensics," in *Proceedings of the CERT 2004, CERT—Asia Pacific Information Technology Security Conference 2004*, Brisbane, Australia, May 2004.

[13] R. Sommer and V. Paxson, "Outside the closed world: on using machine learning for network intrusion detection," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 305–316, Oakland, CA, USA, May 2010.

[14] ISO/IEC 27001, *Information Technology—Security Techniques—Information Security Management Systems—Requirements*, ISO, Geneva, Switzerland, 2013.

[15] ISO/IEC 27002, *Information Technology—Security Techniques—Code of Practice for Information Security Controls*, ISO, Geneva, Switzerland, 2013.

[16] D. D. L. Guillén, V. Morales-Rocha, and L. F. Fernández Martínez, "A systematic review of security threats and countermeasures in SaaS," *Journal of Computer Security*, vol. 28, no. 6, pp. 635–653, 2020.

[17] M. Johns, "Code-injection vulnerabilities in web applications—exemplified at cross-site scripting," *IT—Information Technology*, vol. 53, no. 5, pp. 256–260, 2011.

[18] L. Nkosi, P. Tarwireyi, and M. O. Adigun, "Detecting a malicious insider in the cloud environment using sequential rule mining," in *Proceedings of the 2013 International Conference on Adaptive Science and Technology*, pp. 1–10, Pretoria, South Africa, November 2013.

[19] Ponemon Institute, "2019 global state of cybersecurity in small and medium-sized businesses," https://https://start.keeper.io/2019-ponemon-report, Ponemon Institute, Traverse City, MI, USA, 2020, https://https://start.keeper.io/2019-ponemon-report.

[20] Verizon, *The Verizon Data Breach Investigations Report 2019*, Verizon, New York, NY, USA, 2020, https://enterprise.verizon.com/resources/executivebriefs/2019-dbir-executive-brief.pdf.

[21] C. Modi, N. Chirag, and K. Acha, "Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review," *The Journal of Supercomputing*, vol. 1, no. 73, p. 43, 2017.

[22] W. Wang, M. Kiik, N. Peek et al., "A systematic review of machine learning models for predicting outcomes of stroke with structured data," *PLoS One*, vol. 15, no. 6, p. 43, Article ID e, 2020.

[23] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the International Conference on Machine Learning*, pp. 148–156, Tahoe City, CA, USA, July 1996.

[24] A. Mukherjee, R. P. Sundarraj, and K. Dutta, "Apriori rule-based in-app ad selection online algorithm for improving supply-side platform revenues," *ACM Transactions on Management Information System*, vol. 8, no. 2-3, pp. 1–28, 2017.

[25] A. T. Ihler, J. W. Fisher III., and A. S. Willsky, "Message errors in belief propagation," in *Proceedings of the Neural Information Processing System*, pp. 609–616, Calcutta, India, December 2004.

[26] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: applications and solutions," *ACM Computing Surveys*, vol. 52, no. 4, pp. 1–36, 2019.

[27] S.-B. Hong, J.-S. Kim, J.-H. Baek, and G.-Y. Hong, "Shot change detection using multiple features and classification and regression tree (CART)," in *Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications. MLMTA'03*, pp. 122–128, Las Vegas, Nevada, USA, June 2003.

[28] K. Honda, "Fuzzy clustering/co-clustering and probabilistic mixture models-induced algorithms," in *Fuzzy Sets, Rough Sets, Multisets and Clustering*, V. Torra, A. Dahlbom, and Y. Narukawa, Eds., Springer, Berlin, Germany, 2017pp. 29–43, vol. 671 of Studies in Computational Intelligence.

[29] A. Subramanya and P. P. Talukdar, *Graph-Based Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning*, Morgan & Claypool Publishers, San Rafael,, CA, USA, 2014.

[30] J. Oh, K. Cho, and J. Bruna, "Advancing graphsage with a data-driven node sampling," 2019, http://arxiv.org/abs/1904.12935.

[31] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," *ACM Transactions on Knowledge Discovery*, vol. 5, pp. 1–21, 2012.

[32] S. Basterrech, S. Mohammed, G. Rubino, and M. Soliman, "Levenberg-Marquardt training algorithms for random neural networks," *The Computer Journal*, vol. 54, no. 1, pp. 125–135, 2011.

[33] M. Abou-Zleikha, N. Shaker, and M. G. Christensen, "Preference learning with evolutionary multivariate adaptive regression spline model," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2184–2191, Sendai, Japan, May 2015.

[34] J. Michaelis, M. W. Haslbeck, P. Lammich, and L. Hupel, "Algorithms for reduced ordered binary decision diagrams," *Archive of Formal Proofs*, vol. 2016, 2016.

[35] A. Aiken and B. R. Murphy, "Implementing regular tree expressions," in *Functional Programming Languages and Computer Architecture*, pp. 427–447, Springer, Berlin, Germany, 1991, vol. 523 of Lecture Notes in Computer Science.

[36] L. Wang, Y.-W. Pang, D.-Y. Shen, and N.-H. Yu, "An iterative algorithm for robust kernel principal component analysis," in *Proceedings of the 2007 International Conference on Machine Learning and Cybernetics*, pp. 3484–3489, Hong Kong, China, August 2007.

[37] W. Yu and J. A. McCann, "Random walk with restart over dynamic graphs," in *Proceedings of the 16th IEEE International Conference on Data Mining (ICDM)*, pp. 589–598, Barcelona, Spain, December 2016.

[38] A. Govada, V. S. Thomas, I. Samal, and S. K. Sahay, "Distributed multi-class rule based classification using ripper," in *Proceedings of the 2016 IEEE International Conference on Computer and Information Technology (CIT)*, pp. 303–309, Nadi, Fiji, December 2016.

[39] C. L. Devasena, "Proficiency comparison of ladtree and reptree classifiers for credit risk forecast," 2015, http://arxiv.org/abs/1503.06608.

[40] E. D. Arenas-Díaz, H. Ochoterena-Booth, and K. Rodríguez-Vázquez, "Multiple sequence alignment using evolutionary

algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2009*, pp. 1783-1784, Montreal, Canada, July 2009.

[41] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02*, pp. 538–543, New York, NY, USA, 2002.

[42] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01*, pp. 269–274, New York, NY, USA, 2001.

[43] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *Proceedings of the 16th International Conference on World Wide Web, WWW 2007*, pp. 639–648, Banff, Canada, May 2007.

[44] T. Mahmood and U. Afzal, "Security analytics: big data analytics for cybersecurity: a review of trends, techniques and tools," in *Proceedings of the 2013 2nd National Conference on Information Assurance (NCIA)*, pp. 129–134, Rawalpindi, Pakistan, December 2013.

[45] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS One*, vol. 11, no. 4, Article ID e0152173, 2016.

[46] T. Zoppi, A. Ceccarelli, L. Salani, and A. Bondavalli, "On the educated selection of unsupervised algorithms via attacks and anomaly classes," *Journal of Information Security and Applications*, vol. 52, Article ID 102474, 2020.

[47] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," 2019, http://arxiv.org/abs/1903.02460—Comment: submitted manuscript to Computer & Security.

[48] L. Cheng, F. Liu, and D. D. Yao, "Enterprise data breach: causes, challenges, prevention, and future directions," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 5, 2017.

[49] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[50] X.-L. Li, B. Liu, and S.-K. Ng, "Negative training data can be harmful to text classification," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pp. 218–228, Cambridge, MA, USA, October 2010.