Future Generation Computer Systems 113 (2020) 488–505

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

New approach for threat classification and security risk estimations based on security event management[☆]

José Carlos Sancho, Andrés Caro [*], Mar Ávila, Alberto Bravo

Department of Computer and Telematics Systems Engineering, University of Extremadura, Av/ Universidad S/N, ES-10003, Cáceres, Spain

ARTICLE INFO

Article history:
Received 14 January 2020
Received in revised form 6 June 2020
Accepted 6 July 2020
Available online 18 July 2020

Keywords:
SIEM
Cybersecurity
STRIDE
Knowledge extraction
Data processing
Bug bar

ABSTRACT

Security Information and Event Management (SIEM) systems are essential for identifying cyber attacks, being an extended practice in organizations to detect threats, vulnerabilities and to estimate security risks. The management of events and information related to security is done through systems that provide all the information, processing different data sources. The developing of alternative models that provide complementary information to commercial solutions, based on the same data sources, is presented as a novel and interesting challenge, not only for organizations, but also for the scientific community. This paper presents a new system to classify security threats, computing their criticality according to the Bug Bar technique, with the aim of addressing threats in order of priority. High correlations were achieved between severity risk values achieved from commercial systems and results computed by the new approach. Accordingly, the new proposal could complement the information of SIEM systems, and help in the prediction of criticalities of future threats.

1. Introduction

Security Information and Event Management (SIEM) has been increasingly implemented in organizations, due to the growing importance of cyber security for companies in the last years. These systems supply very useful information about security-related events and potential threats, risks, and vulnerabilities. The detection of unusual behaviors, the generation of alerts and the monitoring of components, based on the information provided by these systems, allows organizations to be better prepared against future risks.

A SIEM system distinguish between legitimate use and a malicious attack in the complex Information Technology infrastructures of todays' organizations. Success in the operation of SIEM systems is based on obtaining data from a variety of sources, as extensive and heterogeneous as possible. Not only for threat detection and threat response, but also for massive data management and analytics, such as risk assessment and security threats rating.

The possibility of assigning a numerical level of severity (*criticality*) to threats detected by a SIEM could be a noteworthy complement to these systems. Furthermore, the possibility of classifying threats in different categories, according to the type of threat (elevation of privileges, denial of service, repudiation ...), would also be certainly appreciated.

This paper proposes a model, called *Viewnext-UEx*, for threats rating, which assigns numerical criticalities to threats, classifying them into different types, and finally weighing the total degree of criticality in order to respond first to the most critical threats. The system ranks threats in order of priority, to process and address threats in that order. The proposed model was developed based on real data, and was also validated against real datasets.

In this sense, datasets are definitely essential for any research paper, especially when real data is considered rather than synthetic data. As an example of this, Migdal and Rosenberger discussed in [1] about the importance of real datasets. The real datasets used in this paper were obtained from the SOC (Security Operations Center) of the *Viewnext* company, and were collected by a commercial SIEM. Before being processed, data was properly adapted as a previous ETL stage (*Extract, Transform, and Load – ETL*). It is essential to manage appropriately the extraction, transformation and loading process when data is retrieved from multiple sources. This procedure is critical, since several solutions could be achieved and from different security perspectives.

The use of real data could be considered as a positive feature of this work. However, the datasets supplied by the software development company were highly unbalanced, since they were collected from real data, which represents a good sample of data distribution on real-world problems. Indeed, unbalanced distributions directly affects the performance of algorithms and models

[☆] This work is financed by the Consejería de Educación y Empleo, Junta de Extremadura, Spain and the European Union (ERDF funds) through the support funds to research groups GRU18138 and IB16089 project.

[*] Corresponding author.
E-mail addresses: jcsanchon@unex.es (J.C. Sancho), andresc@unex.es (A. Caro), mmavila@unex.es (M. Ávila), albertobg@unex.es (A. Bravo).

https://doi.org/10.1016/j.future.2020.07.015
0167-739X/© 2020 Elsevier B.V. All rights reserved.

of data mining and machine learning, because they tend to favor the class with the largest number of samples (the majority class). Therefore, it is desirable to have the classes as balanced as possible, to prevent the effects of unbalanced datasets. SMOTE (*Synthetic Minority Oversampling Technique*) [2] and RUS (*Random Undersampling*) have been traditionally the most used and tested techniques to balance datasets. SMOTE is an over-sampling approach in which the minority class is over-sampled by generating "synthetic" examples, while RUS randomly eliminates instances of the majority class. In this paper, an innovative combination of both techniques is proposed to balance the datasets, avoiding possible situations that can lead to overfitting or underfitting.

As a result, this paper presents the Viewnext-UEx model, a system to allow threats rating and severity classification based on knowledge extraction related to SIEM systems. This experimental approach is performed in an industrial environment, being an interesting standpoint that all data is extracted from the SOC of a leader company. For this reason, the study case presented is a significant issue of this paper. The main features of the Viewnext-UEx model are detailed, discussing some of the strengths of the approach, such as the methods to get balanced datasets, to compute the criticality of threats, and to obtain severity classification. The final objective of the paper is the developing of a system to address security threats in order of priority.

The main contribution of this paper can be summarize as follows: (i) A new emerging model for security threat risk rating is proposed; (ii) Real threats provided by a commercial SIEM have been used to develop and validate the approach; (iii) the new model could help complete and complement the monitoring information provided by SIEM systems; and (iv) the new approach could predict the criticality of future threats, based on current data.

This paper is organized as follows. Section 2 presents the background and related works. Section 3 describes the datasets used in the experiments. Section 4 introduces the new methodology presented in this paper, the Viewnext-UEx system. Then, Section 5 presents the results for the different experiments. Finally, Section 6 presents the conclusions and future works.

2. Background and related works

SIEM tools have become a core part of identifying and addressing cyber attacks. It is increasingly important to incorporate cybersecurity tools and threat detection in business and corporations. Security Event Management (SEM) is based on monitoring and correlating security events in real time, while Security Information Management (SIM) processes data and stores, analyzes and generates reports. Thus, Security Information and Event Management (SIEM) collects, aggregates and analyzes activity from many different resources, finding security breaches and allowing organizations to investigate alerts in real time.

Many organizations have SIEM systems based on well-known commercial solutions, such as QRadar from IBM [3], Arc Sight from HP [4] or alternative solutions to these big corporations, such as Symantec Security Services [5], McAfee SIEM [6], Alien Vault [7], OSSIM [8] or FortiSIEM from Fortinet [9].

The information provided by SIEM systems is essential to know and improve the security level of computer systems [10–12]. Many studies focuses on detecting threats in real time. Thus, Hubballi and Suryanarayanan present in [13] a review of the main commercial SIEM systems for false alarm minimization techniques in signature-based Network Intrusion Detection System (NIDS); Kufel explores the efficiency of threat detection in distributed systems [14]; Al-Duwairi et al. identify and block malicious traffic from IoT devices in [15]. Others researchers, such as El Arass and Souissi, propose in [16] an open source SIEM composed of a Big Data platform ELK integrated with other intrusion detection and load-balancing tools, comparing their results with the ones obtained by IBM Q-Radar.

An interesting point of this paper is the development of a methodology to generate complementary knowledge to the information obtained by SIEM systems. Different studies combine the information extracted from systems with intelligent treatment and detection of real threats. Thus, Gong et al. use in [17] a dataset from four cyber-threat intelligence sources to strengthen incident response and improve security strategy. Lee et al. present a technique for cyber-threats detection, based on artificial neural networks, using two benchmark datasets (NSLKDD and CICIDS2017) and two datasets collected in the real world [18]. Suarez-Tangil et al. use neural networks to classify events according to the corresponding context established for the attack [19].

Another important aspect is the development of a risk assessment system based on the STRIDE model [20]. STRIDE allows the classification of threats into six categories: Spoofing of user identity, Tampering, Repudiation, Information disclosure, Denial of service (D.o.S), and Elevation of privilege. Several authors use STRIDE to model threats instead of using other approaches such as MITRE ATT&CK [21]. STRIDE and MITRE ATT&CK are indicated for slightly different purposes. STRIDE is used to identify vulnerability areas that an attacker can exploit, classifying them in classes based on systems. In contrast, MITRE ATT&CK describes different tasks of the stages of an attack, based on the overall flow of an attack, and not on the whole system. Thus, Xin and Xiaofang [22] use STRIDE threat model to analyze an online banking security case study. Venkatasen and Mani propose in [23] STRIDE to improve the application of electronic government. In [24] Hirano et al. rely on STRIDE to perform a security analysis for mitigating problems of storage forensics in Infrastructure-as-a-Service (IaaS). Seifert and Rez [25] also use STRIDE to determine possible security issues of cyber–physical systems architecture for healthcare.

Different implementations and studies of the categorization of threats with STRIDE are considered [26–29] [30]. Among the variants of the STRIDE model [20] two are emphasized:

- STRIDE per element: specially indicated when it is possible to obtain a diagram of potentially vulnerable elements, identified and categorized. This approach makes it easier to find threats by focusing on a set of threats on each element.
- STRIDE per Interaction: This model focuses on vulnerable interactions in a system. STRIDE per interaction considers tuples of (*origin, destination, interaction*) and enumerates threats on them for categorization.

Both strategies lead to the same number of threats, but the STRIDE per interaction approach was selected for the Viewnext-UEx model. Mainly because the datasets contained mostly information from events and status of information systems.

STRIDE is used as a model for security threat rating. Being originally proposed by Microsoft, STRIDE was combined traditionally with DREAD [31], which was also proposed by Microsoft. DREAD evaluates security threats, providing five categories for risk rating: Damage, Reproducibility, Exploitability, Affected users, and Discoverability. Nevertheless, the ratings assigned by DREAD were not strongly consistent and finally DREAD was out of use by 2008, and replaced by Bug Bar [32].

Bug Bar organizes the priority of threats, according to the classification accomplished by STRIDE. Currently, Microsoft is using Bug Bar in its Security Development Lifecycle (SDL) [33]. The Viewnext-UEx model also includes Bug Bar to compute the severity (criticality) of security threats. Thus, in this paper, threats are categorized according to the STRIDE model and their priority (or severity) is determined according to the Bug Bar model bases.

Data mining and machine learning techniques are used in this paper to obtain knowledge based on the analyzed data. This knowledge could be used to complement the information obtained by the SIEM, and to predict the priority of future threats (prediction of actions). But, also to confirm the validity of the Viewnext-UEx model, correlating the predicted results with the real ones. In this sense, predictive models are used in [34], obtaining high accurate estimations for packet loss prediction. In [35], an interactive method of data visualization for machine learning is presented, whereas massive log prediction based in predictive models is proposed in [36]. In [37] decision trees are used to classify and infer the root cause of security alerts. Therefore, the importance of predictive models applied to security systems is clear.

3. Datasets

Real data was used in the experiments, obtained from the SOC of an Information Technology Services company (called *Viewnext*), as a case study of this company. Data was appropriately anonymized to be used in the experimental design of the proposed approach. The *Viewnext* company is currently formed by a team of more than 4500 professionals specialized in software development, with offices and centers of technological innovation located in Spain and Portugal.

Two different data sources of *Viewnext* were considered, heterogeneous in content, purposes, and objectives. These files were part of the IBM QRadar SIEM dataset of *Viewnext*. This commercial software centralized and monitored different computer systems and devices, managed by the *Viewnext* SOC. According to this company, a SIEM system installed in an infrastructure of an average-size company generates 1000 events per second. That represents approximately 86 million of daily events.

The data sources provided by the *Viewnext* SOC stored real data, which were conveniently anonymized and exclusively processed for the case study. Both data files consist of 10 000 tuples, enough for the research purposes of this work. The content of each file is briefly commented:

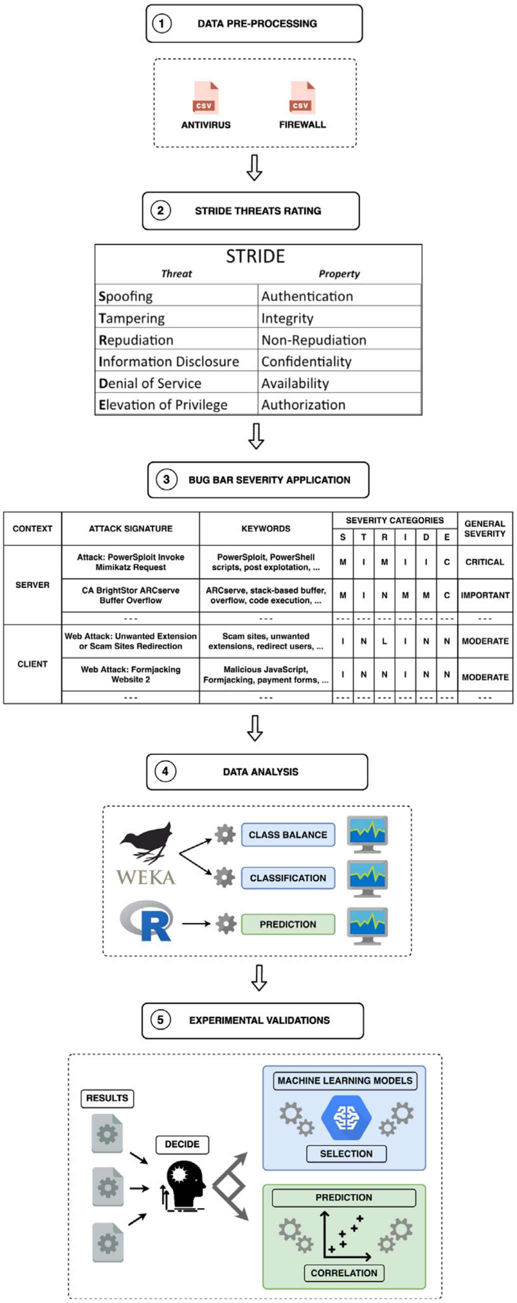
Antivirus: It stores information about the events captured by the antivirus software, specifically Symantec Endpoint Protection from the Symantec company. With a total of 18 columns at this file, the most significant fields are the following:

- device: Name of the device affected by the threat.
- ipAddress: IP address that identifies the specific device on the network.
- severity: Severity assigned by the Antivirus provider.
- protocol: Specifies the protocol used (TCP, UDP…).
- application: Name of the affected application.
- localPort: Local port of the affected device.
- url: Formatted URL address where the threat was detected.

Firewalls: File that contains data about the firewall used, concretely FortiGate [38]. The file consisted of 22 columns, where 8 of them were of research interest:

- severity: Risk level of the identified event.
- dstip: Destination IP address of the produced event.
- dstport: Port number of the traffic's destination.
- action: Firewall provider reaction against the event.
- service: Type of web service protocol (HTTPS, SMB, TCP…).
- attack: Attack signature detected by the firewall provider.
- crlevel: Client reputation level affected by the event.
- profile: Security web profile assigned to the event.

The two files were CSV formatted and had an internal substructure called SYSLOG [39] assigned by the SIEM system.



STRIDE

Threat	*Property*
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-Repudiation
Information Disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

CONTEXT	ATTACK SIGNATURE	KEYWORDS	SEVERITY CATEGORIES						GENERAL SEVERITY
			S	T	R	I	D	E	
SERVER	Attack: PowerSploit Invoke Mimikatz Request	PowerSploit, PowerShell scripts, post explotation, ...	M	I	M	I	I	C	CRITICAL
	CA BrightStor ARCserve Buffer Overflow	ARCserve, stack-based buffer, overflow, code execution, ...	M	I	N	M	M	C	IMPORTANT
	---	---	---	---	---	---	---	---	---
CLIENT	Web Attack: Unwanted Extension or Scam Sites Redirection	Scam sites, unwanted extensions, redirect users, ...	I	N	L	I	N	N	MODERATE
	Web Attack: Formjacking Website 2	Malicious JavaScript, Formjacking, payment forms, ...	I	N	N	I	N	N	MODERATE
	---	---	---	---	---	---	---	---	---

Fig. 1. Viewnext-UEx threat rating system based on SIEM data.

4. The viewnext-UEx knowledge extraction system

The Viewnext-UEx model proposed in this paper is shown in Fig. 1. This system could be implemented in other organizations, since it allows the incorporation of a customized system with machine learning so that the companies have the capabilities to protect themselves against threats.

As can be seen, the system is based on five fundamental steps that are explained below.

STEP 1: Data pre-processing.

First, information of possible threats and vulnerabilities was gathered through different log files, focuses especially on the *Antivirus* and *Firewalls* files. This information was preprocessed and correctly formatted, to be analyzed later. Possible inaccuracies in the source information was also corrected.

The Viewnext-UEx system can be adapted to any organization and to any SIEM system, adjusting the threat pre-processing phase.

STEP 2: STRIDE threats rating. Machine learning was applied to classify the threats into the corresponding STRIDE categories. The impact of the threats on the system was determined, based on the STRIDE categories. As a result, the degree of severity and the STRIDE categories for each threat were determined.

STEP 3: Bug Bar severity application. The proposed system prioritized threats to process first the most serious ones. For this reason, an analysis on the datasets was performed to determine the severity of each one, by using Bug Bar. Four levels were considered (Low, Moderate, Important, and Critical), which allowed determining the criticality of threats and, consequently, their order of priority.

STEP 4: Data Mining analysis. Once the STRIDE categories of the threats and their criticality level was determined (on steps 2 and 3 respectively), the Firewalls and Antivirus datasets were analyzed, mainly to check and avoid problems of unbalanced data.

STEP 5: Experimental validations. The validation of the proposed system was determined by the experimental results. The criticality levels computed by means of the Viewnext-UEx system was compared and correlated to the severity degrees obtained from commercial approaches. The correlations could help to corroborate the validation the proposed model.

The Viewnext-UEx model provides a threat classification process that do not require a third-party proprietary software; it presents a ranking to solve threats with higher priority; and very useful information on the impact, severity and components affected by those threats. In the following subsections, the five steps are explained in detail.

4.1. Data pre-processing (step 1)

The data preprocessing and formatting for both files were based on a semi-automatic process, performing the following steps:

1. Detection and deletion of the content considered as expendable, according to values or fields with useless information or with few relevant information for the study.
2. File information unification: both files presented irregularities in the structure, such as different types of rows or log vectors (not all vectors had the same number of fields). Accordingly, the implemented preprocessing step ensured a uniform schema for the resulting file.
3. A mapping of the information was also performed in this preprocessing step. For instance, columns that stored values like "IP_Source: 129.345.0.128" were mapped to "129.345.0.128" as column value and "IP_Source" as column header.
4. Finally, a validation process was executed in order to guarantee the correct formatting of each value. The pandas library (implemented in Python) was used to achieve it.

In this way, output files were generated in this task with the same type as the input files (CSV), already right formatted to be analyzed.

4.2. STRIDE risk rating security threat (step 2)

The second step was the automatic classification and categorization of threats. One of the main contributions of the Viewnext-UEx model is the development of a new methodology to classify threats based on the dataset processed. This classification was done following the STRIDE model [20], which categorized security threats into six groups: Spoofing of user identity, Tampering, Repudiation, Information disclosure, Denial of service (D.o.S), and Elevation of privilege.

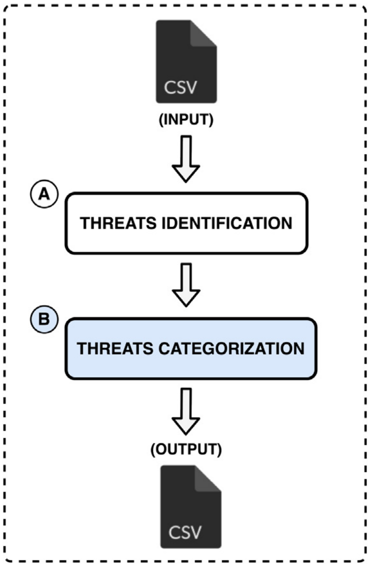


Fig. 2. Threat classification stages.

The full process followed for the classification of threats is illustrated in Fig. 2.

A. Threat identification

The system identified threats in the two files (antivirus and firewall), by analyzing specific fields:

- Antivirus.csv: The total amount of events was 10 000, but after discarding false positives, the system obtained a total of 6672. There were 62 different attack signatures.
- Firewalls.csv: The total amount of events was also 10 000, but after discarding false positives, the system obtained a total of 2124. There were 16 different attack signatures.

Threats were grouped by unique values so that repeated and duplicate threats were not classified in later stages.

B. Threat categorization

The Viewnext-UEx system categorized the threats according to the process described in [20]. For threat rating, the *context* (elements and software components diagram) needs to be defined previously.

First, a table with all existing *interactions* was created using the different vulnerabilities found on the defined context, with their respective outputs based on the STRIDE categorization (Table 1).

Then, a threat classification for the two files (antivirus and firewall) was performed, considering the interaction table. Several interactions were assigned to each threat, creating a STRIDE chain as a result. Table 2 shows examples of threat attack signatures of the Antivirus and Firewall files, once the corresponding interactions were assigned.

Main information used for this threat classification process is shown in Table 3 for several threats of the Antivirus and Firewalls files.

Once the characteristics of each threat were identified, the different interactions were analyzed, linking threats to interactions whenever their applicability was possible. Thus, the applicability of an interaction to a specific threat depends mainly on three aspects:

1. Affected Component: element or component (router, server, computer, switch…) affected by the threat.

Table 1
Threats interaction scheme for STRIDE classification.

Id	Element	Interaction	S	T	R	I	D	E
1	Process (target)	Process has outbound data flow to data store.	X			X		
2		Process sends output to another process.	X		X	X	X	X
3		Process sends output to external interactor (code).	X		X	X	X	
4		Process sends output to external interactor (human).			X			
5		Process has inbound data flow from data store.	X	X			X	X
6		Process has inbound data flow from a process.	X		X		X	X
7		Process has inbound data flow from external interactor.	X				X	X
8	Data Flow (commands/responses)	Crosses machine boundary.		X		X	X	
9	Data Store (database)	Process has outbound data flow to data store.		X	X	X	X	
10		Process has inbound data flow from data store.			X	X	X	
11	External interactor (processes, services, …)	External interactor passes input to process.	X		X	X		
12		External interactor gets input from process.	X					

Table 2
Example of threats classification based on their interaction (Table 1).

Log file source	Threat attack signature	Id interaction	S	T	R	I	D	E
Antivirus	Web Attack: Malicious Domains Request 2	11	X		X	X		
	Attack: Eir D1000 Modem CWMP Router Code Execution	4			X			
		8		X		X	X	
		4–8		X	X	X	X	
Firewalls	CA BrightStor ARCserve Buffer Overflow	7	X				X	X
		8	X		X	X		
		7–8	X		X	X	X	X
	TCP Out of Range Timestamp	3	X		X	X	X	

2. Threat characteristics: as Table 3 shows, description and additional information is important to decide whether to apply a STRIDE category to a threat.
3. Impact: The impact is determined by the affected component and the characteristics of the threat signature: severity of the threat, affected products (hardware, software or both), preventive measures, used service (https, imap, telnet, socks...) and many others. The objective is not to apply the impact of a threat to any context, but to the specific one.

As a result of this threat rating process, some of the STRIDE categories were assigned to the threats. Table 4 shows an example of the result of threat rating.

4.3. Bug bar severity application (step 3)

Fig. 3 shows the methodology to perform the Bug Bar process in the Viewnext-UEx model. This proposal analyzes threats (step 1) by using the Microsoft SDL Bug Bar table (step 2), computing the severity for a specific threat on the STRIDE categories (step 3).

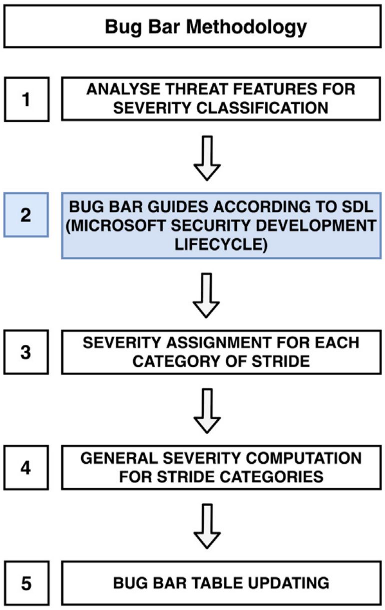

Bug Bar Methodology
1
ANALYSE THREAT FEATURES FOR SEVERITY CLASSIFICATION
2
BUG BAR GUIDES ACCORDING TO SDL (MICROSOFT SECURITY DEVELOPMENT LIFECYCLE)
3
SEVERITY ASSIGNMENT FOR EACH CATEGORY OF STRIDE
4
GENERAL SEVERITY COMPUTATION FOR STRIDE CATEGORIES
5
BUG BAR TABLE UPDATING

Fig. 3. Methodology to perform the 'Bug Bar' process.

The severity score for a threat used in step 3 is explained in Table 5, which describes the security vulnerabilities and scores used in the Bug Bar tables in the Viewnext-UEx model. The categories considered from highest to lowest severity were the following:

1. Elevation of Privilege.
2. Denial of Service.
3. Information Disclosure.
4. Spoofing.
5. Tampering and Repudiation.

The assignment of a severity score to a single STRIDE category for a threat was based on the following features:

1. Threat Scope: Information about the degree of affectation of the software and hardware components.
2. Information about the attacker: any information about the attacker, not only about the identity, but whether the attack is internal or external:

   - Authenticated: The attacker had access to the systems, either through usurpation of credentials or because he/she was an employee. As the attacker had privileges in the system, the system damage is significantly increased.

Table 3
Descriptive information of the threats classified in Table 2.

Attack signature	Severity	Description and additional information	Affected
Web Attack: Malicious	High	You have attempted to visit a known malicious IP address. Visiting this web site could potentially put you at risk to becoming infected. It is recommended that you do NOT visit this site. Users can be silently infected just by visiting a web site with attacks known as drive-by downloads or social engineering attacks where misleading applications can attempt to trick users into installing fake antivirus solutions or fake video players.	All Products
Attack: Eir D1000 Modem CWMP Router Code Execution	High	This signature detects attempts to exploit various vulnerabilities in Eir routers. Various models of Eir routers have multiple vulnerabilities that can allow Authentication ByPass, Information Disclosure, Remote Code Execution and Command Injection.	Various Eir routers, mainly Eir D1000 Modems.
CA BrightStor ARCserve Buffer Overflow	Critical	Computer Associates BrightStor ARCserve Backup is prone to a remote stack-based buffer overflow vulnerability because the application fails to properly check the bounds of user-supplied data prior to copying it to an insufficiently sized buffer. A successful exploit will allow an attacker to execute arbitrary code with system level privileges. The impact on the system leads an attacker to an arbitrary code execution.	Computer Associates BrightStor ARCserve Backup Laptop & Desktop 11.1 - 11.0 - 11.1 SP1
TCP Out of Range Timestamp	Low	This indicates detection of a TCP packet with out-of-range Timestamps option. The Timestamps option is used in PAWS (Protect Against Wrapped Sequences). It carries two four-byte timestamp fields. The Timestamp Value field (TSval) contains the current value of the timestamp which is the time of the TCP sending the option. The Timestamp Echo Reply field (TSecr) contains a timestamp value that was sent by the remote TCP in the TSval field of a Timestamps option. The impact on the system leads a potential denial of service. Monitor the traffic from that computer for any suspicious activity is recommended.	Some of the all affected products: Yamaha RTX2000 and more versions. SCO Unixware 7.1.4 and SCO Unixware 7.1.3. Microsoft Windows XP Tablet PC Edition SP1 and similar versions. Hitachi GS4000 and similar versions. FreeBSD FreeBSD 5.4-RELENG and similar versions F5 BigIP 9.0.5 and similar versions. Cisco Unity Server 4.0 and similar and lower versions. Blue Coat Systems SGOS and Blue Coat Systems CacheOS.

Table 4
Result of the threat categorization process from one of the Antivirus threats.

Threat Signature Att.	ID Int.	Component Int.	Description interaction	S	T	R	I	D	E
Backdoor server activity	4	Server	Process sends output to external interactor (human).			X			
	8	Data flow	Crosses machine boundary.		X		X	X	
					X	X	X	X	

Table 5
Description and score of the severities.

Severity	Id initials	Score	Description
Critical	C	4	A security vulnerability that would be rated as having the highest potential for damage.
Important	I	3	A security vulnerability that would be rated as having significant potential for damage, but less than Critical.
Moderate	M	2	A security vulnerability that would be rated as having moderate potential for damage, but less than Important.
Low	L	1	A security vulnerability that would be rated as having low potential for damage.
None	N	0	This rate is given when it does not represent a security vulnerability.

- Non-Authenticated: The attacker did not have direct access to the systems, so certainly the severity of the threat decreases.

3. Scenario/Context Threat: information about the place, scenario or context of the threat.
4. Reproducibility of the threat: the ability of the threat to multiply or spread the effects on the system or other systems.
5. Countermeasures for the threat: information about countermeasures to combat or invalidate the effect of a threat.

The "score" column in Table 5 establishes a rating score for each category of the threat (step 3 of Fig. 3). The overall score was calculated as the sum of the single scores of each category (step 4 of Fig. 3).

The Bug Bar table evolves dynamically when new threats are evaluated and priorities are recomputed. Following the guidelines of [33] the Bug Bar table becomes a living database of threats, which helps to better prioritize them (step 5 of Fig. 3). Then, the final severity was assigned based on this value, following the guidelines of Table 6.

Table 7 shows a small excerpt from the Bug Bar table result, once the process shown in Fig. 3 was applied.

4.4. Data analysis (step 4) and experimental validations (step 5)

Data analysis was performed using Weka [40] and R [41]. Weka was used to compare the results (criticality) computed by means of the Viewnext-UEx system, and the real severity values provided in the original Antivirus/Firewall files (step 4 of Fig. 1).

The algorithms were selected to analyze data according to the most common approaches in the scientific literature. The parameters of these algorithms are briefly described below:

a. Linear regression:

- Simple Logistic [42] with the following parameters:
  - batchSize: 100.
  - heuristicStop: 50.
  - maxBoostingIterations: 0.
  - numDecimalPlaces: 2.
  - useCrossValidation: True.
  - weightTrimBeta: 0.0.

b. Decision trees:

- J48/C4.5 [43] with the following parameters:
  - batchSize: 100.
  - collapseTree: True.
  - confidenceFactor: 0.25.
  - minNumObj: 2.
  - numDecimalPlaces: 2.
  - numFolds: 3.
  - seed: 1.
  - subtreeRaising: True.
  - useMDLcorrection: True.
- Random Forest [44] with the following parameters:
  - bagSizePercent: 100.
  - batchSize: 100.
  - maxDepth: 0.
  - numDecimalPlaces: 2.
  - numExecutionSlots: 1.
  - numFeatures: 0.
  - numIterations: 100.
  - seed: 1.

c. Bayesian models:

- Naive Bayes [45] with the following parameters:
  - batchSize: 100.
  - numDecimalPlaces: 2.
- Bayes Net [46] with the following parameters:
  - batchSize: 100.
  - estimator: SimpleEstimator -A 0.5.
  - numDecimalPlaces: 2.
  - searchAlgorithm: K2 -P 1 -S BAYES.

d. K-nearest neighbors:

- IBK, K-nearest neighbors classifier [47] with the following parameters:
  - KNN: 1.
  - batchSize: 100.
  - distanceWeighting: No distance weighting.
  - nearestNeighbourSearchAlgorithm: LinearNN Search -A "weka.core.EuclideanDistance -R first-last".
  - numDecimalPlaces: 2.
  - windowSize: 0.

e. Support Vector Machine (SVM):

- SMO, sequential minimal optimization algorithm for training a support vector classifier [48] with the following parameters:
  - batchSize: 100.
  - c: 1.0.
  - calibrator: Logistic -R 1.08E−8 -M -1 -num-decimal-places 4.
  - epsilon: 1.0E−12.
  - filterType: Normalize training data.
  - kernel: PolyKernel -E 1.0 -C 250007.
  - numDecimalPlaces: 2.
  - numFolds: -1.
  - randomSeed: 1.
  - toleranceParameter: 0.001.

Cross-validation was selected for all the analysis. The algorithms were mostly configured with the classical parameters. Only the IBK models needed a mandatory parameter to work, the number of K neighbors, that was set to 1 for all of them.

On the other hand, for the experimental validation (step 5 of Fig. 1), R was used in the Viewnext-UEx system to predict criticalities through different and well-known regressors. Linear regressor, random forest, cforest, SVM, penalized, bagEARTH and Earth regressors were used to compute the correlation coefficient r between the real data and the ones obtained by the Viewnext-UEx system, for the two datasets.

An additional data analysis was also performed to compare the results of working with balanced and unbalanced data. Real data acquired from different resources usually produce unbalanced datasets. Because of this imbalance, most of machine learning algorithms tend to favor classes with highest number of samples. Analyzing and exploring the dataset is required to avoid this effect (Exploratory Data Analysis). In this paper, data balancing was performed using two widely used techniques: SMOTE (Synthetic Minority Oversampling Technique) [2] and RUS (Random Undersampling).

SMOTE and RUS have opposite effects on the balance process. SMOTE generate synthetic instances for the minor class from

Table 6
Severities assigned to a threat from the scores of all its categories.

Severity	Client/Server Score	Description
Critical	15–19	A security vulnerability that would be rated as having the highest potential for damage.
Important	10-14	A security vulnerability that would be rated as having significant potential for damage, but less than Critical.
Moderate	5-9	A security vulnerability that would be rated as having moderate potential for damage, but less than Important.
Low	1-4	A security vulnerability that would be rated as having low potential for damage.
None	0	This rate is given when it does not represent a security vulnerability.

Table 7
Small excerpt of the "Bug Bar" table obtained for the Antivirus and Firewalls files.

Context	Attack signature	Keywords	Severity categories						General severity
			S	T	R	I	D	E	
Server	Attack: PowerSploit Invoke Mimikatz Request	PowerSploit, PowerShell scripts, post-exploitation, penetration test, Code injection	M	I	M	I	I	C	Critical
	Attack: Eir D1000 Modem CWMP Router Code Execution	Eir Routers, Auth ByPass, Information Disclosure, Remote Code Execution	N	I	L	I	I	N	Important
	CA BrightStor ARCserve Buffer Overflow	ARCserve, stack-based buffer, overflow, code execution, system privileges	M	I	N	M	M	C	Important
	...	...	...	...	...	...	...	...	...
Client	Web Attack: Unwanted Extension or Scam Sites Redirection	Scam sites, unwanted extensions, redirect users, fake web pages/tools	I	N	L	I	N	N	Moderate
	Web Attack: Formjacking Website 2	Malicious JavaScript, Formjacking payment forms, e-commerce	I	N	N	I	N	N	Moderate
	Web Attack: Fake Tech Support Website 73	FakeAV, virus, false scans, social engineering attack	I	I	I.	I	I	N	Critical
	...	...	...	...	...	...	...	...	...

their original samples, while RUS randomly decreases samples from the major class. The main objective was to have the largest number of samples of each class in the same proportion. Hence, the application of both techniques consisted in using SMOTE first to generate elements in the minor classes and finally using RUS to eliminate samples and balance the major classes (Fig. 4). Although there is no strict guidance on the percentages or number of samples that should be increased or decreased in a dataset, it is important not to use excessive percentages for any of them, as it can lead to the following problems:

- High percentages in the oversampling phase, can lead to *'overfitting'* situations. Generating a large amount of data that is really similar to the original one would over train the model and could cause the system to make mistakes with new incoming data.
- High percentages in the undersampling phase, can lead to *'underfitting'* situations. Eliminating a large amount of data could lead a lack of information for the model training. Representative information could be randomly deleted in the major class. Due to this, the model will fail in the classification of the samples that were previously eliminated.

Balanced datasets were produced by applying the Viewnext-UEx system, without *overfitting* or *underfitting* situations. Finding the best balance between SMOTE and RUS depends mainly on the distribution and characteristics of the datasets. In this research, different tests were made to find the best distribution.

5. Results and discussion

The final phase of the Viewnext-UEx system is the "Experimental Validations" stage (step 5 of Fig. 1). At this point, the results on the two analyzed files (Antivirus and Firewalls) were contrasted and verified. In this section, only the results obtained for the Antivirus file will be shown and discussed. These results were quite similar to those obtained for the Firewalls samples, and the discussion and conclusions can be extrapolated to both datasets.

5.1. Bug bar severity analysis

In this first experimental analysis, machine learning was used to determine the severity level of each threat, which was established based on Bug Bar for both datasets. The machine learning models supplied the real criticality of threats with high accuracy. In this way, the Viewnext-UEx model selected the most critical threats to be analyzed first, assigning higher priority to these threats. This subsection shows the severity results obtained on the Antivirus file from the point of view of bug bar analysis.

After discarding false positives on the initial 10 000 tuples, the Viewnext-UEx model finally computed the severity for 6672 threats. As Fig. 5A shows, the system classified the 6672 threats of the unbalanced Antivirus file in the four classes: 'Critical', 'Important', 'Moderate' and 'Low'. Most of them were classified as 'Important' or 'Moderate' (4334 and 2188, respectively). Only 113 threats were considered as 'Critical', and 37 as 'Low' criticality. Since algorithms tended to favor the predominant class, SMOTE

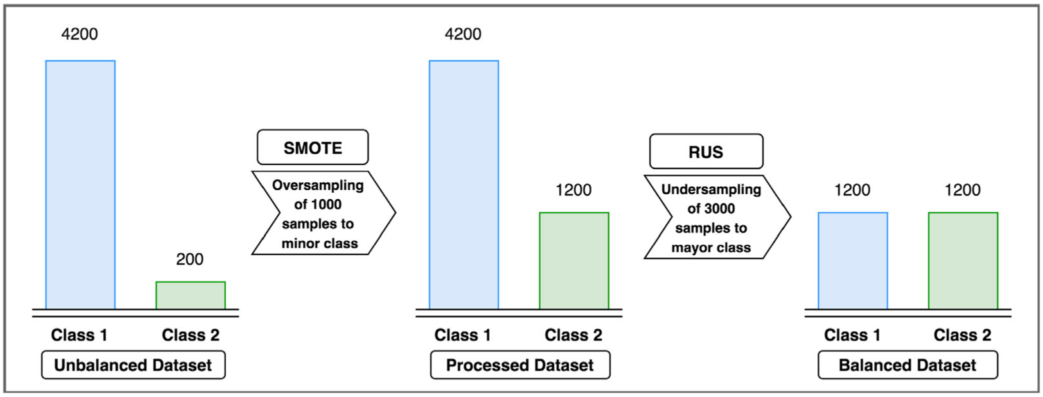


Fig. 4. Scheme of the application process of SMOTE and RUS on a set of unbalanced data.

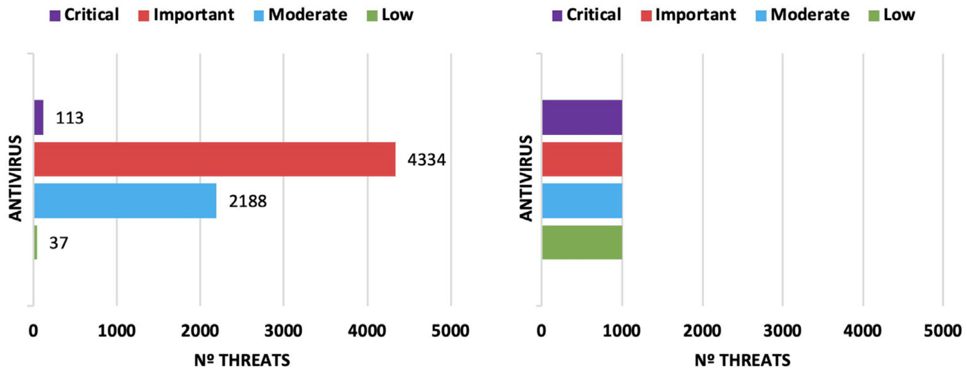


Fig. 5. Classification of severities by Bug Bar (Antivirus file).

and RUS were applied to obtain balancing classes, as mentioned in previous section.

Fig. 5B shows the result after balancing the different classes of the Antivirus dataset, thus obtaining the same number of samples (1000 tuples) after applying SMOTE and RUS, respectively.

The details of the application of SMOTE and RUS on the different severity classes are shown in Table 8. The key aspect in this process was the number of samples to be generated or discarded for each class. Different tests were carried out in our experiments and the distributions that obtained the best results were selected. Although the main objective was to classify all the different levels of criticality as accurately as possible, special focus was taken on the 'Critical' class. This class represents the most critical threats and consequently, the threats of higher interest to be treated in first place.

A first task was accomplished on the Antivirus file by selecting the corresponding input columns for the different machine learning models. The input fields used for this first experimental analysis were the following:

- Device: Name of the device affected by the threat.
- IP address: IP address that identifies the specific device on the network.
- Severity: Severity assigned by the Antivirus software.
- Protocol: Specifies the protocol used (TCP, UDP…).
- Application: Name of the application affected.
- Local port: Local port of the device affected.
- Intrusion URL: URL address where the threat was detected.

The results obtained for the unbalanced and balanced distributions of the Antivirus file are presented in Table 9. As can be seen, good results were obtained for both datasets. The results of CCI (Correct Classified Instances) were greater than 95% for all the classification algorithms. The MAE value (Mean Absolute Error) were mostly low, noting a certain increase in the SMO model. In turn, the Kappa statistics [49] offered very good results, very close to the maximum value of 1. The statistics offered by F-Measure were also high, indicating good results of the Precision and Recall values. From all the classification algorithms tested, linear regression and SMO were selected for the Viewnext-UEx system, since these two models obtained the highest values of CCI.

As mentioned above, the results in Table 9 represent general values without differentiating among severity degrees. On the contrary, Table 10 presents more detailed results, offering information about the classification for each class at both distributions.

Studying the J48 model for the unbalanced dataset, Table 9 showed good results for all the measurements. On the other hand, Table 10a shows that the J48 model classified with great precision the 'Important', 'Moderate' and 'Low' class threats, but it was very imprecise when classifying 'Critical' threats. The Naive Bayes, Bayes Net and IBK models obtained similar results. These values are highlighted in red color in Table 10a. The candidates in this case were linear regression and SMO, as it was before, since they reached the higher scores.

In contrast, on the balanced dataset the results were significantly better, as shown in Table 10b. The accuracy on the 'Critical' class increased significantly for all the previous algorithms (in blue color in Table 10b). In this case, linear regression and J48 tree were selected as candidates for classification algorithms.

As precise algorithms were required for both unbalanced and balanced distribution of the dataset, *Linear Regression* and *SMO* where finally selected for the Viewnext-UEx system, since these two algorithms obtained accurate results regardless of the data distribution used.

Table 8
Number of samples generated and removed by SMOTE and RUS (Antivirus file).

Class name	Balancing technique	Unbalanced data samples	Samples removed	Samples generated	Balanced data samples
Critical	SMOTE	113	0	887	1000
Important	RUS	4334	3334	0	1000
Moderate	RUS	2188	1188	0	1000
Low	SMOTE	37	0	963	1000

Table 9
Classification based on "Bug Bar" severity on the unbalanced and balanced distribution (Antivirus file).

Antivirus dataset	Measures	Linear regression	J48	Random forest	Naive Bayes	Bayes Net	IBK ($k = 1$)	SMO (SVM)
Unbalanced distribution	C.C.I.	99.73	98.17	98.20	95.69	97.20	98.93	99.73
	Kappa	0.99	0.96	0.96	0.91	0.94	0.97	0.99
	M.A.E.	0.00	0.01	0.01	0.02	0.01	0.01	0.25
	F-Measure	0.99	0.97	0.98	0.96	0.97	0.99	0.99
Balanced distribution	C.C.I.	99.55	96.50	97.80	97.52	98.25	98.75	99.72
	Kappa	0.99	0.95	0.97	0.96	0.97	0.98	0.99
	M.A.E.	0.00	0.01	0.03	0.01	0.01	0.01	0.25
	F-Measure	0.99	0.96	0.97	0.97	0.98	0.98	0.99
Selected		X						X

Table 10
Results of the Bug Bar severity classification (Antivirus file).

Severity	Measures	Linear regression	J48	Random forest	Naive Bayes	Bayes Net	IBK ($k = 1$)	SMO (SVM)
a.) Unbalanced distribution								
Critical	TP Rate	0.89	**0.01**	**0.47**	**0.51**	**0.69**	**0.72**	0.90
	F-Measure	0.92	**0.01**	**0.56**	**0.30**	**0.48**	**0.70**	0.92
Important	TP Rate	1.00	1.00	0.99	0.97	0.99	1.00	1.00
	F-Measure	1.00	1.00	0.99	0.98	0.99	1.00	1.00
Moderate	TP Rate	0.99	0.99	0.97	0.94	0.94	0.98	0.99
	F-Measure	0.99	0.97	0.97	0.95	0.95	0.98	0.99
Low	TP Rate	0.97	0.91	0.97	0.91	0.91	0.97	0.97
	F-Measure	0.98	0.95	0.98	0.88	0.86	0.97	0.98
Selected		X						X
b.) Balanced distribution								
Critical	TP Rate	0.98	**0.97**	**0.98**	**0.98**	**0.98**	**0.98**	0.99
	F-Measure	0.99	**0.93**	**0.96**	**0.95**	**0.96**	**0.97**	0.99
Important	TP Rate	1.00	1.00	0.99	0.97	0.99	1.00	1.00
	F-Measure	1.00	0.99	0.99	0.98	0.99	1.00	1.00
Moderate	TP Rate	0.99	0.89	0.93	0.94	0.94	0.96	0.99
	F-Measure	0.99	0.93	0.95	0.96	0.96	0.97	0.99
Low	TP Rate	0.99	0.99	0.99	0.99	0.99	1.00	0.99
	F-Measure	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Selected		X						X

5.2. *STRIDE category analysis*

A severity level was assigned to each threat, based on the previous bug bar severity analysis. These severities were essential, since they determine the order of threats to be analyzed, processing the most critical ones firstly in the proposed system.

Fig. 6a shows the unbalanced distribution of the threats on STRIDE categories from Antivirus dataset, presenting the number of good (V) and bad (X) classifications for each of the six STRIDE categories. Once again, the dataset was certainly unbalanced, especially on the 'S' and 'D' categories. In consequence, a balancing process using SMOTE and RUS techniques was performed for each category, as was done in the previous experimental analysis.

Table 11 presents the details about this process, where the number of generated and removed samples for each category in the dataset are shown. As can be seen, SMOTE and RUS were applied to generate or remove tuples, depending on the number of good (V)/bad (X) classified tuples in each STRIDE category. For the good classified tuples (V), SMOTE generated new samples only in 'R' category, and RUS removed samples in 'S', 'I', 'D', and 'E' categories. However, in the group of bad classified tuples (X), only SMOTE was applied to generate new samples (in 'S', 'T', 'I', 'D', and 'E' categories) since the number of bad classified tuples was much less than that of good classified for all STRIDE categories (except 'R').

With this new balanced distribution, the same number of threats was obtained in each category (Fig. 6b).

The performance of the different machine learning algorithms was examined on the balanced distribution and compared to the results obtained on the unbalanced distribution. Same inputs fields of the first experimental analysis were considered for this second analysis (device, IP address, Severity, Protocol, Application, Local port, Intrusion URL) adding the severity obtained for each threat in the step 3 of Fig. 1 (bug bar severity application, Table 6).

Table 12 presents the results of the analysis of the STRIDE categories in the unbalanced and balanced datasets, for the Antivirus file.

Table 11
Number of samples generated and removed by SMOTE and RUS in STRIDE categories (Antivirus file).

Category	Class name	Balancing technique	Unbalanced data samples	Samples removed	Samples generated	Balanced data samples
S	✓	RUS	6527	5527	0	1000
	×	SMOTE	145	0	855	1000
T	✓	None	3740	0	0	3740
	×	SMOTE	2932	0	808	3740
R	✓	SMOTE	2572	0	1528	4100
	×	None	4100	0	0	4100
I	✓	RUS	4737	1237	0	3500
	×	SMOTE	1935	0	1565	3500
D	✓	RUS	6487	5487	0	1000
	×	SMOTE	185	0	815	1000
E	✓	RUS	4801	1301	0	3500
	×	SMOTE	1871	0	1621	3500

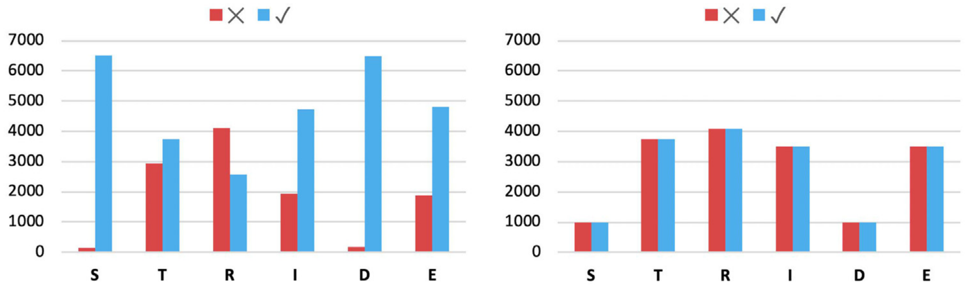


Fig. 6. Incorrect (X) and correct (V) threat classification in STRIDE categories (Antivirus file).

For the unbalanced datasets (Table 12a), the CCI results reached high accuracy, very close to 100% in many categories; and even the lowest CCI values exceeded 90%. High percentages were also obtained for F-Measure value, indicating a notable Precision and Recall values, with some exception. Some F-Measure values with the '?' symbol could not be calculated because the Precision or Recall values were null. These null values may occur due to the inability of J48 algorithm to classify correctly any of the classes of 'S' and 'D' categories. On the other hand, overall correct values of the Kappa measure were obtained, except for some occurrences highlighted in red. In this case, best algorithms were linear regression, J48 tree and SMO, since they reached the higher scores.

For the balanced distribution (Table 12b), results were improved, especially for Kappa or F-Measure, where low values were not reached on Kappa, and F-Measure could be computed correctly in all cases. The candidate algorithms in this case were linear regression, IBK, and SMO.

Although a notable improvement in results was reached with respect to the unbalanced distribution, a deeper analysis was performed for each STRIDE category. Table 13 shows a more detailed information regarding the classification of the different classes of STRIDE.

As mentioned above, for the unbalanced distribution (Table 12a) some values of F-Measure could not be determined; and Kappa was zero in some cases. The reason can be clearly seen in Table 13a, where the J48 algorithm does not classify correctly any of the threats of the 'X' class in the 'S' and 'D' categories ('TP Rate', True Positive Rate, is zero). This caused F-Measure could not be calculated (Precision and Recall values were null). F-Measure near or below 0.5 are highlighted in red, denoting a low relationship between Precision and Recall. Low precision values on the classification of the 'X' class by the 'Random Forest' algorithm in the 'S' and 'D' categories, are also denoted in red.

The balanced datasets reached better results (Table 13b). Values marked in blue at Table 13b showed a great improvement with respect to the ones highlighted in red at Table 13a. 'TP Rate' and F-Measure achieved values above 90% overall, indicating high accuracy and exhaustivity.

In this case, the best algorithms were linear regression, IBK, and SMO for both unbalanced and balanced datasets.

Therefore, considering the above results and discussion, the following algorithms were selected in this second analysis for the Viewnext-UEx system: Linear Regression, IBK, and SMO. These models obtained the best results and, in addition, their results were mostly invariant in both datasets, which qualifies them as the most robust of this analysis.

5.3. Severity analysis for the STRIDE categories

After the two previous experimental analysis, a large amount of very useful information was obtained for the categorization of threats: information of the general severity of the threats and the categories of STRIDE, as well as the initial information for the analysis (input fields selected of the datasets). The STRIDE category for the threats was already determined, discarding in this experimental analysis the threats without assigned categories. In this third experimental analysis, the Viewnext-UEx system was completed, performing several tasks to determine the severities of each STRIDE category.

Fig. 7a presents the unbalanced distribution of the severity levels assigned to the threats, for the STRIDE categories. There is no balance among severity classes as can be seen. Therefore, the same balancing procedure as in the previous experimental analysis was executed. Fig. 7b shows the balanced distribution after applying SMOTE and RUS techniques, and Table 14 shows the details about this process. Classes under 10 samples were not used to the analysis, since SMOTE could not be applied correctly on so small sets.

SMOTE and RUS were applied to generate or remove tuples, depending on the number of samples for the classes 'Low', 'Moderate', 'Important', and 'Critical', in each STRIDE category.

Table 12
Results of the STRIDE classification (Antivirus file).

Category	Measures	Linear regression	J48	Random forest	Naive Bayes	Bayes Net	IBK (k= 1)	SMO (SVM)
a.) Unbalanced distribution								
S	C.C.I.	99.14	97.82	98.26	95.80	96.13	99.05	99.295
	Kappa	0.77	**0**	**0.32**	**0.49**	**0.51**	0.77	0.82
	M.A.E.	0.01	0.04	0.01	0.04	0.03	0.00	0.01
	F-Measure	0.99	**?**	0.97	0.96	0.97	0.99	0.99
T	C.C.I.	93.55	92.14	92.53	84.59	86.66	93.28	93.60
	Kappa	0.87	0.84	0.84	0.68	0.72	0.86	0.87
	M.A.E.	0.10	0.11	0.11	0.14	0.12	0.08	0.06
	F-Measure	0.93	0.92	0.92	0.84	0.86	0.93	0.93
R	C.C.I.	92.65	91.14	90.82	91.93	92.23	91.86	92.37
	Kappa	0.84	0.81	0.80	0.83	0.83	0.83	0.84
	M.A.E.	0.11	0.10	0.11	0.08	0.08	0.09	0.07
	F-Measure	0.92	0.91	0.90	0.92	0.92	0.91	0.92
I	C.C.I.	99.89	96.19	98.20	95.09	95.56	99.41	99.92
	Kappa	0.99	0.91	0.95	0.88	0.89	0.98	0.99
	M.A.E.	0.00	0.03	0.03	0.04	0.04	0.01	0.00
	F-Measure	0.99	0.96	0.98	0.95	0.95	0.99	0.99
D	C.C.I.	99.23	97.22	98.20	94.72	94.28	98.77	99.22
	Kappa	0.86	**0**	**0.51**	**0.47**	**0.46**	0.76	0.85
	M.A.E.	0.01	0.05	0.02	0.05	0.05	0.01	0.01
	F-Measure	0.99	**?**	0.97	0.95	0.95	0.98	0.99
E	C.C.I.	92.08	90.21	90.48	91.69	91.65	91.42	91.92
	Kappa	0.79	0.75	0.75	0.79	0.79	0.78	0.79
	M.A.E.	0.12	0.12	0.12	0.10	0.10	0.10	0.08
	F-Measure	0.91	0.90	0.90	0.91	0.91	0.91	0.91
Selected		X					X	X
b.) Balanced distribution								
S	C.C.I.	99.25	97.35	98.20	98.75	98.80	98.80	99.25
	Kappa	0.98	**0.94**	**0.96**	**0.97**	**0.97**	0.97	0.98
	M.A.E.	0.01	0.04	0.04	0.013	0.01	0.01	0.01
	F-Measure	0.99	**0.97**	0.98	0.98	0.98	0.98	0.99
T	C.C.I.	93.95	92.67	93.42	84.13	85.93	93.85	94.17
	Kappa	0.87	0.85	0.86	0.68	0.71	0.87	0.88
	M.A.E.	0.09	0.11	0.10	0.15	0.13	0.07	0.05
	F-Measure	0.93	0.92	0.93	0.84	0.85	0.93	0.94
R	C.C.I.	94.09	92.93	93.60	92.87	93.23	94.03	94.18
	Kappa	0.88	0.85	0.87	0.85	0.86	0.88	0.88
	M.A.E.	0.10	0.09	0.10	0.07	0.07	0.08	0.05
	F-Measure	0.94	0.92	0.93	0.92	0.93	0.94	0.94
I	C.C.I.	99.92	96.85	99.01	96.77	97.21	99.51	99.94
	Kappa	0.99	0.93	0.98	0.93	0.94	0.99	0.99
	M.A.E.	0.00	0.05	0.03	0.03	0.02	0.01	0.00
	F-Measure	0.99	0.96	0.99	0.96	0.97	0.99	0.99
D	C.C.I.	99.05	96.50	96.70	96.60	96.90	98.30	99.10
	Kappa	0.98	**0.93**	**0.93**	**0.93**	**0.93**	0.96	0.98
	M.A.E.	0.02	0.05	0.05	0.03	0.03	0.01	0.01
	F-Measure	0.99	**0.96**	0.96	0.96	0.96	0.98	0.99
E	C.C.I.	91.07	88.97	90.47	90.02	90.14	90.98	91.27
	Kappa	0.82	0.77	0.80	0.80	0.80	0.81	0.82
	M.A.E.	0.12	0.13	0.13	0.11	0.11	0.11	0.08
	F-Measure	0.91	0.89	0.90	0.90	0.90	0.91	0.91
Selected		X					X	X

Input fields for this third experimental analysis were the following:

- Input fields of the first analysis (Device, IP address, Severity, Protocol, Application, Local port, Intrusion URL) for the Antivirus dataset.
- General severity of threats obtained after the step 3 of Fig. 1 (bug bar severity application, Table 6).
- STRIDE categories assigned to threats (information generated in the output of the second experimental analysis).

Therefore, the previous results were used as feedback to feed the input of the machine learning algorithms.

Table 15 shows the results of the severity classification in the STRIDE categories in the unbalanced and balanced datasets, for the Antivirus file.

Accurate overall results were obtained for the unbalanced distribution (Table 15a), except for some specific cases, highlighted in red color. F-Measure from 'T' category could not be determined for J48 and Random Forest algorithms, probably caused for the same reason as in previous analysis. In addition, these results had a null Kappa value, which indicates that some of the severities probably were not correctly classified.

Again, results were better for the balanced distribution (Table 15b, in blue, values that improved significantly with respect to the balanced dataset). Kappa and F-Measure from the category 'T'

Table 13
Results of the classification of the different levels of criticality in the STRIDE categories (STRIDE category analysis on Antivirus file).

Category	Classes	Measures	Linear regression	J48	Random forest	Naive Bayes	Bayes Net	IBK ($k = 1$)	SMO (SVM)
a.) Unbalanced distribution									
S	✓	TP Rate	0.99	1.00	1.00	0.95	0.96	0.99	0.99
		F-Measure	0.99	0.98	0.99	0.97	0.98	0.99	0.99
	×	TP Rate	0.67	**0.00**	**0.20**	1.00	1.00	0.77	0.80
		F-Measure	0.77	**?**	**0.33**	**0.50**	**0.52**	0.78	0.83
T	✓	TP Rate	0.90	0.87	0.90	0.94	0.96	0.91	0.91
		F-Measure	0.94	0.92	0.93	0.87	0.89	0.93	0.94
	×	TP Rate	0.97	0.97	0.95	0.72	0.74	0.96	0.96
		F-Measure	0.93	0.91	0.91	0.80	0.83	0.92	0.93
R	✓	TP Rate.	0.97	0.93	0.90	0.94	0.95	0.93	0.94
		F-Measure	0.91	0.89	0.88	0.90	0.90	0.89	0.90
	×	TP Rate	0.89	0.90	0.91	0.90	0.90	0.90	0.90
		F-Measure	0.93	0.92	0.92	0.93	0.93	0.93	0.93
I	✓	TP Rate	0.99	0.94	0.99	0.93	0.93	0.99	0.99
		F-Measure	0.99	0.97	0.98	0.96	0.96	0.99	0.99
	×	TP Rate	0.99	1.00	0.95	1.00	0.99	0.99	0.99
		F-Measure	0.99	0.93	0.96	0.92	0.92	0.99	0.99
D	✓	TP Rate	0.99	1.00	1.00	0.94	0.94	0.99	0.99
		F-Measure	0.99	0.98	0.99	0.97	0.97	0.99	0.99
	×	TP Rate	0.90	**0.00**	**0.35**	0.95	0.97	0.76	0.87
		F-Measure	0.86	**?**	**0.52**	**0.50**	**0.48**	0.77	0.86
E	✓	TP Rate	0.98	0.94	0.95	0.93	0.93	0.95	0.95
		F-Measure	0.94	0.93	0.93	0.94	0.94	0.94	0.94
	×	TP Rate	0.76	0.78	0.77	0.87	0.87	0.81	0.82
		F-Measure	0.84	0.81	0.82	0.85	0.85	0.84	0.85
Selected			X					X	X
b.) Balanced distribution									
S	✓	TP Rate	0.98	0.95	0.97	0.97	0.97	0.98	0.98
		F-Measure	0.99	0.97	0.98	0.98	0.98	0.98	0.99
	×	TP Rate	0.99	**0.99**	**0.98**	1.00	1.00	0.99	0.99
		F-Measure	0.99	**0.97**	**0.98**	**0.98**	**0.98**	0.98	0.99
T	✓	TP Rate	0.90	0.86	0.89	0.94	0.96	0.90	0.90
		F-Measure	0.93	0.92	0.93	0.85	0.87	0.93	0.94
	×	TP Rate	0.97	0.98	0.97	0.73	0.75	0.96	0.97
		F-Measure	0.94	0.93	0.93	0.82	0.84	0.94	0.94
R	✓	TP Rate.	0.99	0.99	0.98	0.95	0.96	0.98	0.99
		F-Measure	0.94	0.86	0.93	0.93	0.93	0.94	0.94
	×	TP Rate	0.88	0.92	0.89	0.90	0.90	0.89	0.89
		F-Measure	0.93	0.93	0.93	0.92	0.93	0.93	0.93
I	✓	TP Rate	0.99	0.96	0.98	0.93	0.94	0.99	0.99
		F-Measure	0.99	0.97	0.99	0.96	0.97	0.99	0.99
	×	TP Rate	0.99	1.00	0.99	1.00	1.00	0.99	0.99
		F-Measure	0.99	0.97	0.99	0.96	0.97	0.99	0.99
D	✓	TP Rate	0.98	0.93	0.94	0.93	0.94	0.97	0.98
		F-Measure	0.99	0.96	0.96	0.96	0.96	0.98	0.99
	×	TP Rate	0.99	**0.99**	**0.98**	0.99	0.99	0.98	0.99
		F-Measure	0.99	**0.96**	**0.96**	**0.96**	**0.97**	0.98	0.99
E	✓	TP Rate	0.92	0.89	0.89	0.88	0.88	0.90	0.91
		F-Measure	0.91	0.89	0.90	0.89	0.90	0.90	0.91
	×	TP Rate	0.90	0.88	0.91	0.91	0.91	0.91	0.91
		F-Measure	0.91	0.88	0.90	0.90	0.90	0.91	0.91
Selected			X					X	X

reached significantly higher values in this distribution. The best algorithms for the balanced dataset were IBK and SMO, whereas for the unbalanced distribution, Bayes Net, IBK and SMO achieved the highest results.

Table 16 shows the results of the classification of the different levels of criticality ('Critical', 'Important', 'Moderate' and 'Low') in the STRIDE categories, providing deeper and more detailed information about the analysis.

On the one hand, for the unbalanced distribution (Table 16a), anomalous results and low values were detected (all of them highlighted in red). J48 and Random Forest algorithms did not correctly classify any threat of 'L' class on 'T' category, which explains why the F-Measure could not be determined. On the other hand, IBK and SMO algorithms classified perfectly all the severities corresponding to 'I', 'D', and 'E' categories. These algorithms also classified with a high percentage the rest of the categories, except 'L' class of 'T' category, possibly because it only had 15 samples. However, Bayesian algorithms got the best results for that class.

Table 16b shows in blue the most significant improvements for the balanced distribution. Algorithms that classified the 'L' class of the 'T' category with errors and anomalous values at the balanced distribution achieved an excellent accuracy with this dataset. At the same time, the rest of the algorithms improved their results in this same class and in most of the remaining ones, approaching all the results obtained at the same level.

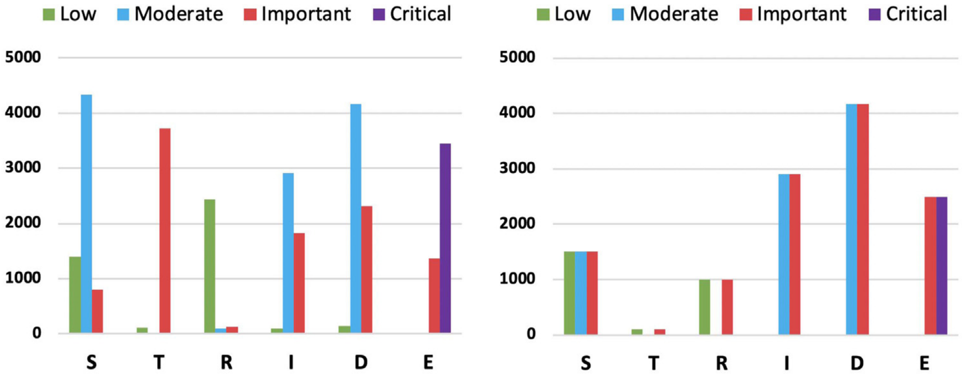


Fig. 7. Distribution of the severity levels assigned to threats, for the STRIDE categories (Antivirus file).

Table 14
Number of samples generated and removed by SMOTE and RUS for Bug Bar classes, in STRIDE categories (Antivirus file).

Category	Class name	Balancing technique	Unbalanced data samples	Samples removed	Samples generated	Balanced data samples
S	I	SMOTE	796	0	704	1500
	M	RUS	4337	2837	0	1500
	L	SMOTE	1394	0	106	1500
T	I	RUS	3725	3625	0	100
	L	SMOTE	15	0	85	100
R	I	SMOTE	127	0	873	1000
	M	Remove	2	2	0	0
	L	RUS	2443	1443	0	1000
I	I	SMOTE	1831	0	1073	2904
	M	None	2904	0	0	2904
	L	Remove	2	2	0	0
D	I	SMOTE	2320	0	1842	4162
	M	None	4162	0	0	4162
	L	Remove	5	5	0	0
E	C	RUS	3441	941	0	2500
	I	SMOTE	1360	0	1140	2500

The IBK and SMO algorithms proved to be the most robust in the performed analysis for both unbalanced and balanced datasets, since their results were mostly invariant and very accurate. Nevertheless, both approaches presented low precision in classifying threats of 'L' class in 'T' category, on the unbalanced dataset. For that reason, the Bayes Net model was also selected. Although this method did not obtain better percentages on most severities (compared to IBK and SMO), the overall classification performance was high and strong in both distributions. Then, the three indicated algorithms (Bayes Net, IBK, and SMO) were suggested for the Viewnext-UEx system.

5.4. Severity prediction using the Viewnext-UEx system

Last experimental analysis tried to predict the real severity of a threat (a tuple of the IBM QRadar SIEM system provided by the *Viewnext* SOC), computed by the Viewnext-UEx system.

The objective of this last analysis lied not only in proving the validity of the proposed system but also in assigning severity levels to new threats by means of the Viewnext-UEx model.

In this sense, high correlations between the predicted severity and the actual one would corroborate that the proposed system could be a real alternative to the commercial approaches.

Thus, in the last analysis, the Viewnext-UEx system was applied to compute a severity value to each threat (SIEM tuple) for each of the STRIDE categories. That is, for each tuple of the IBM QRadar SIEM dataset of *Viewnext* SOC, the Viewnext-UEx system determined the severity for 'S', 'T', 'R', 'I', 'D' and 'E' categories. The experimental analysis consisted of analyzing 6672 tuples of the antivirus file, and 2124 tuples of the firewall file. A supervised system was designed, where the real criticality was known in advance for all tuples (indicated by the commercial antivirus [50] and firewall [9]). Through different and well-known general regressors, the severity of each tuple was predicted.

Correlations obtained by these regressors are shown in Table 17. These correlations between the actual data (obtained by the commercial SIEM), and the predicted data (computed by the Viewnext-UEx system) are high for both Antivirus and Firewall datasets. This fact demonstrates the validity of the proposed model, since it assigns severity values to threats that are similar to those detected by commercial systems. Moreover, high correlations demonstrate the ability to predict severity levels for future threats based on current data.

The Viewnext-UEx system could complement the information of SIEM systems. This new approach could be used as a complementary tool to help in the detection of threats, to avoid false negatives (undetected threats), or false positives (non-threat events, classified as threats). Security is increased when most of the models report similar results in threat classification. Similarly, when discrepancies in the classification are detected for a particular event, this information also enriches and enhances the classification system.

Table 15
Results of the severity classification in STRIDE categories (Antivirus file).

Category	Measures	Linear regression	J48	Random forest	Naive Bayes	Bayes Net	IBK ($k = 1$)	SMO (SVM)
a.) Unbalanced distribution								
S	C.C.I.	99.95	99.90	99.78	99.67	99.95	99.95	99.95
	Kappa	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	M.A.E.	0.01	0.00	0.01	0.00	0.00	0.00	0.22
	F-Measure	1.00	0.99	0.99	0.99	1.00	1.00	1.00
T	C.C.I.	99.59	99.59	99.59	96.60	96.52	99.67	99.70
	Kappa	**0.11**	**0.00**	**0.00**	**0.18**	**0.18**	0.49	0.52
	M.A.E.	0.01	0.01	0.00	0.03	0.03	0.00	0.00
	F-Measure	0.99	**?**	**?**	0.97	0.97	0.99	0.99
R	C.C.I.	99.64	99.22	98.67	95.44	95.21	99.64	99.72
	Kappa	0.96	0.91	0.84	0.65	0.65	0.96	0.97
	M.A.E.	0.01	0.01	0.02	0.04	0.04	0.00	0.00
	F-Measure	0.99	0.99	0.98	0.96	0.95	0.99	0.99
I	C.C.I.	99.95	99.89	100	99.89	99.89	100	100
	Kappa	0.99	0.99	1.00	0.99	0.99	1.00	1.00
	M.A.E.	0.00	0.00	0.01	0.00	0.00	0.00	0.00
	F-Measure	1.00	0.99	1.00	0.99	0.99	1.00	1.00
D	C.C.I.	99.93	99.96	99.44	93.93	95.31	100	100
	Kappa	0.99	0.99	0.98	0.87	0.89	1.00	1.00
	M.A.E.	0.00	0.00	0.02	0.06	0.04	0.00	0.00
	F-Measure	0.99	1.00	0.99	0.94	0.95	1.00	1.00
E	C.C.I.	100	99.95	100	92.73	96.56	100	100
	Kappa	1.00	0.99	1.00	0.83	0.91	1.00	1.00
	M.A.E.	0.02	0.00	0.01	0.07	0.03	0.00	0.00
	F-Measure	1.00	1.00	1.00	0.92	0.96	1.00	1.00
Selected						X	X	X
b.) Balanced distribution								
S	C.C.I.	99.97	99.97	99.91	99.11	99.95	99.97	99.97
	Kappa	0.99	0.99	0.99	0.98	0.99	0.99	0.99
	M.A.E.	0.01	0.00	0.02	0.01	0.00	0.00	0.22
	F-Measure	1.00	1.00	0.99	0.99	1.00	1.00	1.00
T	C.C.I.	99.00	99.00	99.00	99.00	99.00	99.00	99.00
	Kappa	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	0.98	0.98
	M.A.E.	0.13	0.01	0.07	0.01	0.01	0.01	0.01
	F-Measure	0.99	**0.99**	**0.99**	0.99	0.99	0.99	0.99
R	C.C.I.	99.65	98.85	99.25	97.35	99.05	99.5	99.65
	Kappa	0.99	0.97	0.98	0.94	0.98	0.99	0.99
	M.A.E.	0.02	0.01	0.03	0.02	0.01	0.01	0.00
	F-Measure	0.99	0.98	0.99	0.97	0.99	0.99	0.99
I	C.C.I.	99.91	99.86	100	99.91	99.91	100	100
	Kappa	0.99	0.99	1.00	0.99	0.99	1.00	1.00
	M.A.E.	0.00	0.00	0.01	0.00	0.00	0.00	0.00
	F-Measure	0.99	0.99	1.00	0.99	0.99	1.00	1.00
D	C.C.I.	99.92	99.90	99.50	95.29	97.18	100	100
	Kappa	0.99	0.99	0.99	0.90	0.94	1.00	1.00
	M.A.E.	0.00	0.00	0.02	0.04	0.02	0.00	0.00
	F-Measure	0.99	0.99	0.99	0.953	0.97	1.00	1.00
E	C.C.I.	100	99.96	99.94	95.56	98.42	100	100
	Kappa	1.00	0.99	0.99	0.91	0.96	1.00	1.00
	M.A.E.	0.02	0.00	0.01	0.04	0.01	0.00	0.00
	F-Measure	1.00	1.00	0.99	0.95	0.98	1.00	1.00
Selected						X	X	X

6. Conclusions and future works

In this work, a complete model is presented to classify threats according to the STRIDE categories, and to compute the final criticality of threats (level of severity) in order to rank and prioritize threats to process and address them in order.

The Viewnext-UEx model is tested with real data and the case study shows the distribution of different classes from both datasets (Antivirus and Firewall files) was unbalanced. Better results were achieved through the balance of datasets using well-known techniques in the Viewnext-UEx approach.

The Viewnext-UEx model has obtained similar criticalities to those of the Antivirus and Firewall providers, and the high correlation between the threats rating assigned by the proposed system and the real ones provided by commercial systems prove the validity of the proposal.

Finally, the proposed model complements and even complete the monitoring process provided by commercial SIEM systems, allowing the prediction of future threat criticalities based on current data.

New datasets would be incorporated in the future, to feed the threat collections for the Viewnext-UEx model. In this regard, *Viewnext* is considering seriously the inclusion of the information provided by the Viewnext-UEx approach in the company's SOC.

Table 16
Results of the classification of the different levels of criticality in the STRIDE categories (severity analysis for the STRIDE categories on Antivirus file).

Category	Class	Measures	Linear regression	J48	Random forest	Naive Bayes	Bayes Net	IBK ($k = 1$)	SMO (SVM)
a.) Unbalanced distribution									
S	I	TP Rate	1.00	0.99	0.98	1.00	1.00	1.00	1.00
		F-Measure	0.99	0.99	0.99	0.98	0.99	0.99	0.99
	M	TP Rate	0.99	0.99	0.99	0.99	0.99	0.99	0.99
		F-Measure	1.00	0.99	0.99	0.99	1.00	1.00	1.00
	L	TP Rate	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		F-Measure	1.00	1.00	1.00	1.00	1.00	1.00	1.00
T	I	TP Rate	1.00	1.00	1.00	0.96	0.96	0.99	0.99
		F-Measure	0.99	0.99	0.99	0.98	0.98	0.99	0.99
	L	TP Rate	**0.06**	**0.00**	**0.00**	1.00	1.00	**0.40**	**0.40**
		F-Measure	**0.11**	**?**	**?**	**0.19**	**0.18**	0.50	0.52
R	I	TP Rate.	0.94	0.85	0.74	0.97	1.00	0.93	0.94
		F-Measure	0.96	0.91	0.84	0.67	0.67	0.96	0.97
	L	TP Rate	0.99	0.99	1.00	0.95	0.95	1.00	1.00
		F-Measure	0.99	0.99	0.99	0.97	0.97	0.99	0.99
I	I	TP Rate	0.99	1.00	1.00	1.00	1.00	1.00	1.00
		F-Measure	0.99	0.99	1.00	0.99	0.99	1.00	1.00
	M	TP Rate	1.00	0.99	1.00	0.99	0.99	1.00	1.00
		F-Measure	1.00	0.99	1.00	0.99	0.99	1.00	1.00
D	I	TP Rate	0.99	0.99	0.98	0.96	0.96	1.00	1.00
		F-Measure	0.99	1.00	0.99	0.91	0.93	1.00	1.00
	M	TP Rate	1.00	1.00	0.99	0.92	0.94	1.00	1.00
		F-Measure	1.00	1.00	0.99	0.95	0.96	1.00	1.00
E	C	TP Rate	1.00	0.99	1.00	0.89	0.95	1.00	1.00
		F-Measure	1.00	1.00	1.00	0.94	0.97	1.00	1.00
	I	TP Rate	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		F-Measure	1.00	0.99	1.00	0.88	0.94	1.00	1.00
Selected							X	X	X
b.) Balanced distribution									
S	I	TP Rate	1.00	1.00	1.00	0.99	0.99	1.00	1.00
		F-Measure	1.00	1.00	0.99	0.98	0.99	1.00	1.00
	M	TP Rate	0.99	0.99	0.99	0.99	0.99	0.99	0.99
		F-Measure	1.00	1.00	0.99	0.99	0.99	1.00	1.00
	L	TP Rate	1.00	1.00	1.00	0.97	1.00	1.00	1.00
		F-Measure	1.00	1.00	1.00	0.98	1.00	1.00	1.00
T	I	TP Rate	0.98	0.98	0.98	0.98	0.98	0.98	0.98
		F-Measure	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	L	TP Rate	**1.00**	**1.00**	**1.00**	1.00	1.00	**1.00**	**1.00**
		F-Measure	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	0.99	0.99
R	I	TP Rate.	0.99	0.99	0.99	0.98	0.99	0.99	0.99
		F-Measure	0.99	0.98	0.99	0.97	0.99	0.99	0.99
	L	TP Rate	1.00	0.98	0.99	0.96	0.98	0.99	1.00
		F-Measure	0.99	0.98	0.99	0.97	0.99	0.99	0.99
I	I	TP Rate	0.99	0.99	1.00	1.00	1.00	1.00	1.00
		F-Measure	0.99	0.99	1.00	0.99	0.99	1.00	1.00
	M	TP Rate	0.99	0.99	1.00	0.99	0.99	1.00	1.00
		F-Measure	0.99	0.99	1.00	0.99	0.99	1.00	1.00
D	I	TP Rate	0.99	0.99	0.99	0.97	0.97	1.00	1.00
		F-Measure	0.99	0.99	0.99	0.95	0.97	1.00	1.00
	M	TP Rate	0.99	0.99	0.99	0.93	0.97	1.00	1.00
		F-Measure	0.99	0.99	0.99	0.95	0.97	1.00	1.00
E	C	TP Rate	1.00	0.99	0.99	0.91	0.96	1.00	1.00
		F-Measure	1.00	1.00	0.99	0.95	0.98	1.00	1.00
	I	TP Rate	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		F-Measure	1.00	1.00	0.99	0.95	0.98	1.00	1.00
Selected							X	X	X

Table 17
Correlation coefficient r between real data and those obtained through Viewnext-UEx system, for the two datasets.

Regressor	Antivirus	Firewall
Linear regressor	0.883889446331373	0.980050308393072
Random forest	0.992922850571811	0.987221477058714
CForest	0.99600861103051	0.990804372486612
SVM	0.986761952064612	0.988101524709246
Penalized	0.883889446331374	0.980050308392981
bagEARTH	0.883885871850257	0.979760939048875
Earth	0.883889446331374	0.97967071594959

CRediT authorship contribution statement

José Carlos Sancho: Conceptualization, Methodology, Writing - review & editing, Visualization, Funding acquisition. **Andrés Caro:** Conceptualization, Methodology, Software, Data curation, Writing - original draft, Funding acquisition. **Mar Ávila:** Software, Validation, Writing - review & editing, Resources, Investigation. **Alberto Bravo:** Software, Validation, Data curation, Resources, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D. Migdal, C. Rosenberger, Statistical modeling of keystroke dynamics samples for the generation of synthetic datasets, Future Gener. Comput. Syst. 100 (2019) 907–920, http://dx.doi.org/10.1016/j.future.2019.03.056.

[2] W.P.K. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, SMOTE synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357, http://dx.doi.org/10.1613/jair.953.

[3] IBM, IBM QRadar SIEM, (n.d.), https://www.ibm.com/es-es/marketplace/ibm-qradar-siem.

[4] A.E.S.M. (ESM), Security Information and Event Management (SIEM), (n.d.), https://www.microfocus.com/en-us/products/siem-security-information-event-management/overview.

[5] Symantec, Symantec Managed Security Services, (n.d.), https://www.symantec.com/services/cyber-security-services/managed-security-services.

[6] McAfee, Información de seguridad y administración de eventos (SIEM), (n.d.), https://www.mcafee.com/enterprise/es-es/products/siem-products.html.

[7] Alienvault, Alienvault Cibersecurity, (n.d.), https://www.alienvault.com.

[8] OSSIM, Open Source Security Information Management, (n.d.).

[9] Fortinet, FortiSIEM: Powerful Security Information and Event Management, (n.d.), https://www.fortinet.com/products/siem/fortisiem.html.

[10] F. Libeau, Automating security events management, Netw. Secur. 2008 (2008) 6–9, http://dx.doi.org/10.1016/S1353-4858(08)70139-9.

[11] M. Chopra, C. Mahapatra, Significance of security information and event management (SIEM) in modern organizations, Int. J. Innov. Technol. Explor. Eng. 8 (2019).

[12] T. Khan, M. Alam, A. Akhunzada, A. Hur, M. Asif, M.K. Khan, Towards augmented proactive cyberthreat intelligence, J. Parallel Distrib. Comput. 124 (2019) 47–59, http://dx.doi.org/10.1016/j.jpdc.2018.10.006.

[13] N. Hubballi, V. Suryanarayanan, False alarm minimization techniques in signature-based intrusion detection systems: A survey, Comput. Commun. 49 (2014) 1–17, http://dx.doi.org/10.1016/j.comcom.2014.04.012.

[14] L. Kufel, Security event monitoring in a distributed systems environment, IEEE Secur. Priv. 11 (2013) 36–43, http://dx.doi.org/10.1109/MSP.2012.61.

[15] B. Al-Duwairi, W. Al-Kahla, M.A. AlRefai, Y. Abdelqader, A. Rawash, R. Fahmawi, SIEM-based detection and mitigation of IoT-botnet DDoS attacks, Int. J. Electr. Comput. Eng. 10 (2020) 2182–2191, http://dx.doi.org/10.11591/ijece.v10i2.pp2182-2191.

[16] M. El Arass, N. Souissi, Smart SIEM: From big data logs and events to smart data alerts, Int. J. Innov. Technol. Explor. Eng. 8 (2019) 3186–3191.

[17] S. Gong, J. Cho, C. Lee, A reliability comparison method for OSINT validity analysis, IEEE Trans. Ind. Inform. 14 (2018) 5428–5435, http://dx.doi.org/10.1109/TII.2018.2857213.

[18] J. Lee, J. Kim, I. Kim, K. Han, Cyber threat detection based on artificial neural networks using event profiles, IEEE Access. 7 (2019) 165607–165626, http://dx.doi.org/10.1109/ACCESS.2019.2953095.

[19] G. Suarez-Tangil, E. Palomar, A. Ribagorda, I. Sanz, Providing SIEM systems with self-adaptation, Inf. Fusion. 21 (2015) 145–158, http://dx.doi.org/10.1016/j.inffus.2013.04.009.

[20] Adam Shostack, Threat Modeling: Designing for Security, 2014.

[21] ATT&CK, MITRE, (n.d.), https://attack.mitre.org/.

[22] T. Xin, B. Xiaofang, Online banking security analysis based on STRIDE threat model, Int. J. Secur. Appl. 8 (2014) 271–282, http://dx.doi.org/10.14257/ijsia.2014.8.2.28.

[23] M. Venkatasen, P. Mani, A risk-centric defensive architecture for threat modelling in e-government application, Electron. Gov. 14 (2018) 16–31, http://dx.doi.org/10.1504/EG.2018.089537.

[24] M. Hirano, N. Tsuzuki, S. Ikeda, R. Kobayashi, Logdrive: a proactive data collection and analysis framework for time-traveling forensic investigation in iaas cloud environments, J. Cloud Comput. 7 (2018) 1–25, http://dx.doi.org/10.1186/s13677-018-0119-2.

[25] D. Seifert, H. Rez, A security analysis of cyber-physical systems architecture for healthcare, Computers. 5 (2016) http://dx.doi.org/10.3390/computers5040027.

[26] M. Abomhara, M. Gerdes, A STRIDE-based threat model for telehealth systems, Mohamed Abomhara, Martin Gerdes, Geir M. Køien Department of Information and Communication Technology, 2015.

[27] J. Lopez, J. Zhou, M.S. Eds, D. Hutchison, STRIDE to a secure smart grid in a hybrid cloud, 2018, pp. 77–90, http://dx.doi.org/10.1007/978-3-319-72817-9.

[28] J. Lopez, J. Zhou, M.S. Eds, D. Hutchison, Towards security threats that matter 1 (2018) 47–62, http://dx.doi.org/10.1007/978-3-319-72817-9.

[29] S. Krishnan, A hybrid approach to threat modelling, 2017, https://blogs.sans.org/appsecstreetfighter/files/2017/03/A-Hybrid-Approach-to-Threat-Modelling.pdf.

[30] R. Klöti, V. Kotronis, P. Smith, OpenFlow: A security analysis, in: Proc. - Int. Conf. Netw. Protoc. ICNP. 2013, http://dx.doi.org/10.1109/ICNP.2013.6733671.

[31] Microsoft, Improving Web Application Security: Threats and Countermeasures - June 2003 / Ch 3: Threat Modeling, (n.d.), https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644(v=pandp.10)?redirectedfrom=MSDN.

[32] M. Vulnerability, S. Classification, Server – Severity Pivot, 2018.

[33] Microsoft, Security development lifecyle: SDL process guidance Version 5.1, 2011, p. 168.

[34] A. Giannakou, D. Dwivedi, S. Peisert, A machine learning approach for packet loss prediction in science flows, Future Gener. Comput. Syst. 102 (2019) 190–197, http://dx.doi.org/10.1016/j.future.2019.07.053.

[35] W. Zong, Y.-W. Chow, W. Susilo, Interactive three-dimensional visualization of network intrusion detection data for machine learning, Future Gener. Comput. Syst. (2019) http://dx.doi.org/10.1016/j.future.2019.07.045.

[36] P. Wu, Z. Lu, Q. Zhou, Z. Lei, X. Li, M. Qiu, P.C.K. Hung, Bigdata logs analysis based on seq2seq networks for cognitive Internet of Things, Future Gener. Comput. Syst. 90 (2019) 477–488, http://dx.doi.org/10.1016/j.future.2018.08.021.

[37] D. Cotroneo, A. Paudice, A. Pecchia, Automated root cause identification of security alerts: Evaluation in a SaaS Cloud, Future Gener. Comput. Syst. 56 (2016) 375–387, http://dx.doi.org/10.1016/j.future.2015.09.009.

[38] Fortinet, Fortinet named a Leader in the 2018 Gartner Enterprise Firewall Magic Quadrant, 2018, https://www.fortinet.com/products/next-generation-firewall.html.

[39] IETF (Internet Engineering Task Force), The Syslog Protocol, 2009. https://tools.ietf.org/html/rfc5424.

[40] I.H. Witten, E. Frank, M. A. Hall, Data Mining:Practical Machine Learning Tools and Techniques, third ed., 2011, 9780123748560.

[41] W.N. Venables, D.M. Smith, Official@ An Introduction to R, R. 0, 2011.

[42] N. Landwehr, M. Hall, E. Frank, Logistic model trees, in: Lect. Notes Artif. Intell., in: Lect. Notes Comput. Sci., vol. 2837, 2003, pp. 241–252.

[43] J. Quinlan Ross, C4. 5: Programs for machine learning, Mach. Learn. 240 (1993) 302, http://dx.doi.org/10.1016/S0019-9958(62)90649-6.

[44] L. Breiman, Breiman2001 - Random Forests, 2001, pp. 1–33.

[45] P.H. Jhon, George, Langley, Estimating Continuos Distributions in Bayesian Classifiers, 1995.

[46] R.R. Bouckaert, Bayesian network classifiers in Weka, Working paper series. University of Waikato, Department of Computer Science. No. 14/2004, University of Waikato, Hamilton, New Zealand, 2004.

[47] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1991) 37–66, http://dx.doi.org/10.1023/A:1022689900470.

[48] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, Improvements to Platt's SMO algorithm for SVM classifier design, Neural Comput. 13 (2001) 637–649, http://dx.doi.org/10.1162/089976601300014493.

[49] A.J. Viera, J.M. Garrett, Vierra 2005 interrater agreement, in: Kappa Statistic, 2005, pp. 360–363.

[50] Symantec, Severity Assessment: Threats, events, vulnerabilities, risks, 2006, https://www.symantec.com/content/en/us/about/media/securityintelligence/SSR-Severity-Assesment.pdf.

José Carlos Sancho received the Computer Science M.Sc. degree from University of Extremadura in 2015. He is currently an Assistant Professor in the Department of Computer and Telematics Systems Engineering at the University of Extremadura and member of the Media Engineering Group. He is currently finishing his Ph.D. studies. His research area focuses on the Audit and Security of Software Development, and Cybersecurity. He has participated in several conferences and workshops, being coauthor of several research papers.

Andrés Caro is an Associate Professor in the Department of Computer and Telematics Systems Engineering at the University of Extremadura since 1999. He received the B.Sc. and M.Sc. degrees in Computer Science in 1993 and 1998, respectively, and the Ph.D. degree in Computer Science in 2006, from the University of Extremadura, Spain. He is the Lab head of Media Engineering Group in University of Extremadura. His research lines are involved with Cybersecurity, Big Data and Machine Learning, and Pattern Recognition. He has participated in several R&D projects and he is coauthor of numerous research SCI journal papers and book chapters.

Mar Ávila received her B.Sc. degree in 1997, and the M.Sc. degree in 1999, both in Computer Science from the University of Extremadura. In 2018, she received her Ph.D. degree in Computer Science. She is an assistant professor in the Department of Computer and Telematics Systems Engineering of the University of Extremadura Since 2002. Her research interest includes Pattern Recognition, Data Mining and Machine Learning, Information Retrieval and Cybersecurity. She has participated in several research projects, being coauthor of several SCI journal papers.

Alberto Bravo received the Software Engineering B.Sc. degree from University of Extremadura in 2018 and he is currently finishing the M.Sc. degree in Computer Science at the same university. He is a researcher in the Department of Computer and Telematics Systems Engineering at the University of Extremadura. His research interests include Data Mining, Information Retrieval and Security Threats Systems. He is coauthor of several papers of cybersecurity and threat detection patterns.