SPARTA

D4.4

Cybersecurity threat prediction framework

Project number	830892
Project acronym	SPARTA
Project title	Strategic Programs for Advanced Research and Technology in Europe
Start date of the project	1st February, 2019
Duration	41 months
Programme	H2020-SU-ICT-2018-2020

Deliverable type	Report
Deliverable reference number	SU-ICT-03-830892 / D4.4 / V1.0
Work package contributing to the deliverable	WP4
Due date	May 2022 – M40
Actual submission date	15th June, 2022

Responsible organisation	IND
Editor	David Pérez Comendador
Dissemination level	PU
Revision	V1.0

Abstract	This document describes a prediction framework for anticipating and projecting cyber situations, including design, capabilities, instantiation recommendations, etc.; as well as an in-depth revision of the state of the art on cyber threat prediction and any other topic of the concern of projecting the acquired cyber situational awareness
Keywords	Anticipation, Cybersecurity, Prediction, Situational Awareness, SPARTA,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892.

Editor

David Pérez Comendador (IND)

Contributors (ordered according to beneficiary numbers)

Zadnik, Martin (CESNET)

Marc Ohm (UBO)

Mario Reyes De Los Mozos (EUT)

David Pérez Comendador, Jose Francisco Suarez Mulero, Jorge Maestre Vidal, Roberto Sáez Ruiz, Roumen Daton Medeneu Choumanof, Estefanía Fuentes Fernández, Jorge Álvaro González, Enrique Herrero Martínez, Juan Jesús Marqués Ortiz, Miguel Páramo Castrillo, Álvaro Morales Moreno, Sergio Diez García, Javier Ignacio Rojo Lacal, Marco Antonio Sotelo Monge, Jesús García Molina, David Álvarez Orviz, Álvaro Aguilar Alhama (IND)

Paolo Mori, Oleksii Osliak (CNR)

Jocelyn Aubert (LIST)

Filipe Apolinário (INOV)

Miguel Correia (IST)

Reviewers (ordered according to beneficiary numbers)

Yuri Gil Dantas (FTS)

Fabio De Rosa (CINI)

Disclaimer

Executive Summary

This document is the result of the work of the T-SHARK program, known as SPARTA WP4. This document, the Deliverable 4.4, addresses the Cybersecurity threat prediction framework, including the definition of approach, methods to be used and process organization to provide comprehensive prediction of full-spectrum cybersecurity threats.

In this deliverable several beneficiaries are involved, including 7 industrial partners, 4 SMEs, 16 research organizations, 14 universities and 3 non-profit organizations. This constellation enables to tackle the problem of predicting cyber situations with an exhaustive approach, including researchers, teachers, engineers, developers and end-user needs.

First, the deliverable reviews the state-of-the-art in situational awareness and prediction of cyber threats, including key concepts, research directions, problematic, predictable threats, methods for validation and discussion of the most widely adopted standard and solutions in the context of threat prediction is described. This first focus also includes a SWOT analysis of the proposed prediction frameworks, evaluating the applicable requirements.

Secondly, the proposed prediction framework side-wide design decisions, including objectives, assumptions, requirements, limitations and other relevant aspects to be considered are presented. This section is made in order to identify and resolve possible divergences and to detail the mentioned framework.

Thirdly, the situational awareness prediction framework, which assist to the proactive response to cybersecurity threats is introduced. This third section also explores the fundamental architectural design principles and relevant aspects within the prediction framework, such as the reasoning of its results, proposed methods and procedures for the validation.

Finally, this deliverable reviews the requirements, summarizes the lessons learned, and presents the key achieved conclusions while pointing out suggested future research directions.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

Although the concept of hybrid warfare is as old as war itself, the implications of non-conventional threats extend to all modern conflict dimensions: politics, society, diplomacy, economics, etc., being a clear example of this the dissemination of false and manipulated information. Consequently, a feasible approach to threat detection and prevention must be analogously hybrid, while bearing in mind both reactive but also proactive decision-making and countermeasures. On one hand, we must consider the heterogeneous composition of information sources and, on the other, adapt threat/risk analyses to complex geostrategic constrains, like regulation, ethics, proportionality, rules of engagement, etc.

The level of complexity of modern conflict scenarios increases drastically if we focus on critical infrastructure environments, where the impact of a possible attack can lead to dramatic consequences. It is obvious that the requirements in terms of safety and security are the highest in those infrastructures whose systems, resources and services are essential for the progress of society and ensure the continuity in the normal operation of the services provided by states, public administrations, etc. which obviously include the industrial ecosystem that sustain the local, national, and international economies. Additionally, modern cybercriminals look for vulnerabilities in critical infrastructure systems for financial gain, but new interests are emerging, such as obtaining relevant information, taking control of an activity or an entire organization, and what could be worse, paralyzing or terminating the activity, which, in many cases, are difficult to remedy once they happened, thus suggestion more proactive defensive approaches.

As response to the effect-focused consequences channeled by cyber vectors on such dimensions, together to the raising need for acting faster against cyber-attacks (before the damage caused becomes critical and irreversible), the capability of predicting and hence proactively responding postulates as an essential cornerstone for managing cyber situations on real operational environments. But despite its relevance, there are important concerning how to adopt an harmonize prediction-based defense posture, which among others has so far been segregated on the basis of the nature of specific proprietary solutions, local application frameworks, or limited to a specific action in terms of threats to be considered, prediction horizons, granularity of information, types of sources, specific information on threats, etc.

1.2 Objective and Contextualization

For the reasons above, and assuming the ambition of the SPARTA project, the Consortium is obliged to establish the most rigorous levels in terms of protection and security, applying the most reliable standards, and the most avant-garde techniques and tools; also in the context of proactive defense postures, where the capability of anticipate to cyber situations is essential. With this in mind:

> The main purpose of the present research (SPARTA deliverable D4.4) is to introduce a cybersecurity threat prediction framework, which shall include the definition of approach, methods to be used and process organization to provide comprehensive prediction of full-spectrum cybersecurity threats

In contrast with previous related work, this deliverable arises from the consensus of a large ecosystem of beneficiary and associated actors, including direct feeds form external expert boards and cybersecurity R&D&I centres in Europe: the consortium. It is a well-balanced mixture of 7 industrial partners, 4 SMEs, 16 research organizations, 14 universities and 3 non-profit organizations. This constellation enables to tackle the problem of predicting cyber situations with an exhaustive approach, including researchers, teachers, engineers, developers and end-user needs.

The result may not be the most incisive in certain areas of application, or some areas of application may not have been fully covered (this was not the intention). Instead, the focus has been on the identification, analysis, and formulation of a joint solution, scalable to the specific needs of each participant, and interoperable with the growing ecosystem of available off-the-shelf solutions. A rational thinking exercise has been undertaken in which, after identifying commonalities, directions have been sought to harmoniously address points of divergence; all this without adding closures for further research, innovation, and development actions.

1.3 Document Identification

This Deliverable (D4.4 *Cybersecurity threat prediction framework*) was created and reviewed as part of the activities addressed by the H2020 SPARTA (*Strategic programs for advanced research and technology in Europe*) project. SPARTA gathers a large ecosystem of cybersecurity stakeholders, which together aims to set up unique collaborations, build transformative capabilities and form world-leading expertise centres. Through innovative governance, ambitious demonstration cases and active community engagement, SPARTA intends to re-think the way cybersecurity research is performed in Europe across various domains and fields of expertise.

Within the SPARTA project, this Deliverable is part of the activities conducted on its *Full Spectrum Situational Awareness* (T-SHARK) research program. T-SHARK aims on stablishing a Full-Spectrum Cybersecurity Threat Intelligence Framework by developing comprehensive solutions based on novel technology developments and cross-disciplinary breakthroughs. It will provide decision-making tools, fostering a common cybersecurity culture, raising preparedness for possible disruptions and attacks. This is directly linked to the project Work Package (WP) 4: *Program #1: T-SHARK – Full-spectrum cybersecurity awareness*; where the need for bringing "*the definition of approach, methods to be used and process organization to provide comprehensive prediction of full-spectrum cybersecurity threat*" is directly addressed.

1.4 Document Overview

This document is composed by 10 main Sections or Chapters, the first of them (**Chapter 1**) being the present introduction. **Chapter 2** reviews the state-of-the-art on situational awareness and prediction of cyber threats, including key concepts, research directions, problematic, predictable threats, and methods for validation. **Chapter 3** discusses the most widely adopted standard and solutions in the context of threat prediction. **Chapter 4** presents the proposed prediction framework side-wide design decisions, including objectives, assumptions, requirements, limitations, methodology and an early analysis of Strengths, Weaknesses, Opportunities, and Threats (SWOT); the later assisting to identify commonalities between the participant stakeholders and resolving divergences. **Chapter 5** introduces and details a novel prediction framework for assisting the proactive response to full-spectrum cybersecurity threats, as well as instantiating the framework itself for different scenarios. **Chapter 6** suggests methods and procedures for verify and validate instantiations of the proposed prediction framework. Based on the above, **Chapter 7** briefly recaps how each assumed requirement has been tacked. **Chapter 8** illustrated some of the most relevant design/implementation decisions made. **Chapter 9** summarizes the lessons learned, and **Chapter 10** presents the key achieved conclusions while pointing out suggested future research directions.

Chapter 2 Research Background

The main purpose of this section is to introduce the reader to existing landscape on predicting cyber situations, including topics like situational awareness, prediction of cyber threats, the management of concept drifts with anticipative purpose, the predictable attacks/threats, evaluation procedures, and existing related resources (standards products, research trends, etc.). The identification and analysis of this background constituted a preliminarily step towards defining the problem statement to be developed during the rest of the sections, as well to bring clarity on the convergent/divergent vision of the participants; the later allowing to progress towards achieve a joint vision.

2.1 Situational Awareness

[1] Situational Awareness (SA) is a human “brain state” that refers to being conscious of the operational context and the development of planned/ongoing actions with the aim on selecting and planning more effective reactive/proactive Courses of Action (CoAs). This term has been actively revisited by Mica Endsley, who proposed the most adopted SA model in recent decades [2]. The Endlsey’s model layered the SA into three mayor phases: Perception of the operational environment; Comprehension of the perceived information so the inference of new related knowledge is possible; and the Projection of the SA at different future time horizons. Other authors like Bedny and Meister [3] and Smith and Hancock [4] proposed alternative models of SA, which have been widely modified since their publication. A well-known abstraction of the SA paradigm is the OODA (Observe Orient Decide Act) loop model proposed by former USAF colonel John Boyd for supporting fast decision-making, actually constituting the core pillar of the Command and Control (C2) solutions.

By following the coining SA philosophy, the research community has put significantly effort towards adapting these cognitive models to the cyberspace, referring to the resultant mind state as Cyber Situation Awareness (CSA). As Jajodia et al. stated in [6]: to protect critical network infrastructures and missions, we must understand not only the vulnerabilities of each individual system, but also their inter-dependencies and how they support missions, which gains difficulty when operating on emerging technological ecosystems [7] [8], combining the perception of both insider and outsider threats [9] or facing adversarial evasion tactics [10], [11]. At the same time, they proposed a framework to obtain mission-centric SA (Cauldron) combining data fusion, network paths of vulnerabilities, alert correlation, mission impact analysis and recommended reactive/proactive mitigation actions [12] [13]. Based on the three Endsley’s SA levels, McGuinness and Foy [14] added a fourth layer towards developing an alternative CSA model grounded on Perception, Comprehension, Projection and Resolution. On the other hand, Lenders et al. [15] made evolve the John Boyd’s OODA loop to a cyber-perspective. Buckshaw et al. [16] proposed MORDA (Mission Oriented Risk and Design Analysis of Critical Information) based on attack trees, adversary models, user models, service provider models and analysis models to define a quantitative risk assessment and management by means of Multiple Objective Decision Analysis algorithms and SMEs (Subject Matter Expert).

As response to the increasing need for standardized CSA solutions, MITRE developed a related framework fitted to the NATO Communications and Information Agency [17] which comprised Threat Intelligence, Dependency and Impact Analysis, Analysis of Alternatives and Emerging Solutions module to be aware of threats and actors, dependencies and possible countermeasures against discovered threats. Public convergence actions, as is the case of the project H2020 PROTECTIVE, are exploring the context awareness to assess assets critically, the later relying on three basic features: Mission Impact Management, Asset State Management, and Mission and Asset Information Repositories. Complementary, research like [18] is exploring the definition of intelligence systems for critical infrastructure protection under hybrid scenarios (physical and cyber data sources), the latter by taking advantage of three main modules: Data Gathering, Data Analysis and Data Visualization. Finally, in [19] Franke and Brynielsson analyzed both CSA approaches: on one hand industry control and critical infrastructure and in other hand military. They concluded that first

approach has been widely researched whereas military CSA application has much less bibliographical coverage.

2.2 Prediction of cyber threats

Nowadays, organizations are aware or become aware very fast, that the Internet is not a safe place for their applications, services, infrastructure, or users. To face this challenge, the organizations deploy measures to reduce the threat landscape and, subsequently, the measures to recognise that a network threat materializes into an attack. The organizations deploy a variety of network monitoring and analysis systems, including intrusion detection systems (IDS), honeypots, network flow monitors, etc. Data from these systems are more and more shared among multiple organizations using various data-sharing platforms, which allows for more proactive solutions (e.g. blocking the most dangerous attackers seen in other networks) rather reactive ones.

The pieces of information about detected cyber-attacks, especially, when being shared in large sharing communities, for example in [20], [21], [22], and the advance in artificial intelligence, enables to undertake research into proactive measures such as the prediction of cyber-attacks in [24], [25] and [26].

Methods for prediction of cyber threats in cyber security revolves around three topics [25]: attack projection, intrusion prediction and network cyber security situation.

Attack projection predicts the next move of an attacker based on the progress of an ongoing attack and to estimate the goal of an attack. A survey, made by [28], reviews the methods of attack projection till 2014. Recent research, such as [25], [26], [27], proposes to use data-mining methods such as pattern inference in the series of alerts to estimate the most probable upcoming step. Another works ([28], [29] and [30]) investigate Hidden Markov Models to build a propagation model of the multi-step attacks. Another example, presented by [31] experiments with LSTM for prediction.

The situation forecasting estimates what is the overall cyber security situation in the network, e.g. the number of vulnerabilities, the number of detected attacks. The two examples of this domain are investigated in [32] and [33]. They apply time series analysis to estimate trends. Another example, [34], assess the current network security situation and feeds it to the neural network to predict its future situation.

The intrusion prediction tries also to predict parameters of the attacks such as its type, location/victim and the time. The survey made by [35] and [24], summarizes the research of intrusion prediction. In [35], the authors identify multiple methods the prediction can be based upon such as alert correlation, sequences of actions, statistical and probabilistic methods, and feature extraction using techniques such as machine learning, Bayesian networks, Petri nets and data mining.

Bayesian networks and Bayesian Attack Graphs are used in [36]. Bayesian estimation in combination with Kalman Filter are used in [37] to correlate events along observation time. Bayesian networks are also used in [38] to discover attack strategies. Naive Bayes Classification with characteristic vector represent attack samples, each dimension consists of seven attributes like protocol, time, IP addresses is used in [39]. Naive Bayesian networks are also used in [40] for event correlation, especially to detect coordinated attacks, e.g., DDoS attacks.

Scenarios Graphs are used in [41] to identify related alerts that could be part of a coordinated attack plan that could be missed by a human analyst. Sequential pattern mining algorithm was used in [42] to discover complicated multistage attack behaviour patterns. Data mining techniques for fusing alerts based on their similarities into scenarios are used by [43]. Hidden Coloured Petri-Nets are used by [44] for alert correlation and understanding, i.e., alerts are aggregated (fused) into sequences of scenarios (the paper plans to cover detection of coordinated attacks as the future work). Machine learning approach, specifically Multilayer Perceptron (MLP) and Support Vector Machine (SVM), in combination with alert correlation matrix (representing estimated probability of similarity between alert pairs) are used in [45] to determine which alerts in the past correlate with the current alerts. The goal of [45] is to group alerts and represent the correlated alerts as attack scenarios. Granger Causality Test (GCT), which is a time series analysis method, is introduced by

[46] for temporal correlation of related security alerts in DEF CON 9 dataset. GCT aims to test if a time series variable X correlates with another time series variable Y by performing a statistical hypothesis test.

The third topic revolves around the situation of a network or an entity (e.g. IP address). Prevalently, the works investigate how to model a reputation score of suspicious networks or particular hosts. The non-uniform distribution of malicious sources in the IP address space was studied in a series of works on the so called bad neighbourhoods [47], [48], [49] and [50], which is the term used for networks with high ratio of malicious IP addresses. The authors propose to aggregate IP addresses listed on various blacklists by their common prefix (usually of length /24) and create lists of prefixes (networks) with too many blacklisted IP addresses.

However, to block the whole network is often too harsh, therefore there appeared works considering individual hosts[51]. is one of the examples. This uses reputation defined as a probability of the next attack based on previous history of the network host. Further works in this domain propose methods to explicitly predict which malicious sources are likely to attack in a near future (in contrast to classic blacklists, which only list those attacking in the past). The goal of the proposed methods is to prepare a blacklist for each organization which contains those sources that are the most likely to attack the organization network within the next day.[52] introduced the concept of creating these targeted highly predictive blacklists and provided a method based on leveraging correlations among sets of attackers targeting individual organizations. [53] presented a method that significantly improved the precision of generated blacklists. It models the problem as a recommendation system which combines several prediction methods.

Another supportive research is investigated by [54]. They analyses lists of IP addresses reported as malicious by various Google services. For example, they show that 1% of the most active malign IP addresses are responsible for 48--82% of all attacks (depending on the service attacked). They also found significant correlations between lists of addresses attacking different services, i.e. in some cases a single address is used to attack multiple services. Similar characteristics of behaviour of malicious IP addresses are observed in other works, such as Wahid's work[55] or by [56].

To this end, it makes sense to create a scoring system which will capture the reputation derived by various methods. An attempt to implement such a system is described by Dulanoy [57], where the authors propose the scoring of IP addresses as well as other identifiers (so called indicators of compromise) within the MISP threat sharing platform. The authors define a method to manage the score. The score is used to estimate whether an indicator is still relevant or not. It is based on indicator observations, assigned tags and reliability of data sources. The score of an indicator is reset to its maximum value every time an observation of that indicator is reported, and it decreases in time by a predefined formula. When the score reaches zero, the indicator is marked as expired and can be discarded.

Some recent approaches also include non-technical data sources, like sentiment analysis on social networks [58] or [59] to predict potential attacks against an organization or changes in user behaviour [60],or detection of changes in user behaviour to predict insider threats [61].

To sum up, multiple predictive methods in the area of cybersecurity has been proposed in recent years and while they are good proofs of concept showing that predicting future attacks is possible, they still have very limited capabilities and, therefore, limited use in practice. For example, they only allow to predict the expected number of detected attacks in a future time interval, the most probable next step of an already ongoing multi-stage attack, or just the probability there will be some attack originating from a given source.

Regarding previous uses of machine learning methods for attack prediction, most proposed approaches are simple shallow learning methods. To the best of our knowledge, the only works using some kind of deep learning methods, are few already presented in the previous sections. These works focus mostly on projection of an ongoing attack, rather than prediction of new attacks.

Most of the previous works focus on prediction of future attacks (or rather future steps of a complex attack) against a single target. However, recent works showed, it is also useful to predict future behaviour of previously identified malicious sources. Such a view can be especially useful in

connection to various alert sharing platforms, which are being increasingly used in the last years, since it allows to leverage information about attacks against different targets to predict future ones.

Potential sharing of predictive models to improve prediction capabilities, prediction based on asset and known threats.

2.3 Management of Concept drift

Initial efforts on cyber threat prediction were settled on the grounds of statistically static/linear distributions, mainly focusing on revealing “next attack steps” relying on the incrementally principle of cyber threat management, but also on direct prognosis on host-based and network-based metrics. The prediction of **attack sequences** was usually addressed by associating each attack step (e.g. phases within Cyber Kill Chains [62]) with statistical stages, where well-known modeling approaches Like Hidden Markov Models [63] easily allow to adapt the prediction capabilities to different multistep attacks for enabling early and even anticipative response.

Concerning **network-based indicators**, Traffic prediction and hence network-level anticipation has been extensively investigated since the acceptance of the self-similar and the long-range dependence nature of networks traffic. Note that with the advent of the emergent communication technologies, old network activity modeling-based solutions, as those based on the Poisson distribution for client-server interactions, fall into disgrace [64], recently being compared with black/brown noise, and displaying non-linearity [65]. In contrast, studies like [66] discovered that despite heterogeneity, there universally exist some traffic statistical modeling characteristics at server or application granularity. Makris et al. summarize some of the more relevant changes between past context definitions and current networks [67]: 1) firstly, they noted that conventional approaches toward anticipatory networking represented the observed situations as sets of numerical values, but now they require to be expressed as factual knowledge that allows reasoning and conclusion inference; 2) they tend to provide a static vision of the monitored environment, while recent networks demand a representation of its evolution, usually with holistic nature; and Unlike its predecessors, 3) modern solutions need to consider event dependencies and facilitate reactive/proactive actuations unnoticed by users, as is the case of the closed-loop SON paradigm. These can even apply beyond network-based predictive analysis

For example, the **prediction of host-based indicators** entails a combination of both modeling states and threat prognosis when forecasting historic and current value situation of the services and host features (user behaviors, data/directory exploring and access, privilege requests, system call concatenation, etc.), where a static vision of the situation may not-support the real operational dynamism on current computing systems [68]. On the other hand, and in the grounds of the “*Dealing with Attackers, not only Attacks*” perspective [69], some researcher are exploring the combination of technical evidence with DIME (Diplomatic, Information, Military and Economic) **hybrid dimensions** also relevant for the attacker motivations and the success of cyber situations that aim on cause hybrid effects. An example well-known for the cybersecurity community is the tracking of payments related activities based on cryptocurrency transaction records [70]. Another example is in the analysis of suspicious activities in social media, forums, etc. that may reveal a potential attacker intention [71]. Predictive analysis is also being studied for anticipating cyber cascade effects, where not only ICT level traits by also physical observations can be considered for protecting and enable business continuity on complex (often critical) cybernetic systems like smart grids [72].

Several surveys collect the recent advances and trends in the above [73] [74] [75], but despite the specificity of each proposal or how it complements the other ones, there is a common cornerstone for their proper operation: most machine learning and data mining methods on which they rely on, especially those aimed at anomaly recognition for intrusion detection [76], assume that the reference data collections have stationary distributions. They also tend to assume that the information to analyses come from an environment with similar characteristics to the models which they were trained, a situation that is not often satisfied when operating in real scenarios. If proposal considered these situations, researchers usually delegated on the system operators the need for keep updated models, which tends to be a supervised/semi-supervised with significant operational cost and high

human skill dependency, leading to errors or inadequate updating momentum. Consequently, and as discussed in [77], this can lead to unrealistic and unpredictable behaviors able to thwart the supported system effectiveness while misguiding cyber response decisions. In predictive analytics and machine learning, the situation described when the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways, is referred to as concept drift. Terms such as Lifelong Learning, Adaptive Learning or Continual learning have become popular keywords in this context, resulting a practical need for predictive AI adoption in real operational environments. The following subsections will delve into this concept and its problematic surrounding to adopt prediction models for early or anticipatory cyber response.

2.3.1 Problematic and consequences

One of the most widely adopted assumption about the concept drift in anticipate cybersecurity systems was issued by Elwell and Polikar [78] as part of the discussions on the incremental learning of concept drift situations at systems exposed to non-stationary environments, as is the case of the cyber domain:

(…) Concept drift refers to a change in the class (concept) definitions over time, and therefore a change in the distributions from which the data for these concepts are drawn. An environment from which such data is obtained is a non-stationary environment (...)

Formally, let the observation x and the class w, the probability of distribution changes in the monitoring environment is defined from the following expression under the Bayes Theorem:

$$P(\omega|x)=\frac{P(x|\omega)P(\omega)}{P(x)}$$

On which the posterior probability changes over time $(P_{t+1}(\omega|x) \neq P_t(\omega|x))$, were $P(x)$ describes the feature-based probabilities (evidence) of the data used to build cyber predictive models; $P(x|\omega)$ describes the likelihood of observing the element x within a particular class; and $P(\omega)$, defines class prior probabilities as well as class balance to the overall distribution. Elwell and Polikar two core situations on the above: virtual and real drift. The incoherency between concepts is commonly referred to as *concept shift*. Accordingly:

- **Virtual Drift**. Occurs when class drift happens without overlapping of true class boundaries, usually interpreted as a situation where the learner is being provided with additional data from the same environment revealing an incomplete representation of the distribution of the considered data; typically solved by supplemental learning (e.g. reinforcement learning)
- **Real Drift**. There is class overlapping, leading to a situation that may derive in a wrong representation of the true distribution of the current data. Hence old knowledge becomes irrelevant, and it must be revised

Three situations may lead to the scenarios described above: Where 1) $P(w)$ is susceptible to change over time; 2) the distribution at w may vary; 3) $P(w|x)$ also may vary on future [78]. A more detailed classification of the concept drift adopting as discrimination criteria the severity and speed of the derived situation was discussed in [80]: sudden, gradual, incremental, recurring and blips, which are described below (see Figure 1)

- Sudden Drift. There is an immediate transition between $P_t(\omega|x)$ and $P_{t+1}(\omega|x)$ but $P_{t+1}(\omega|x) \neq P_t(\omega|x))$. This is a raw real drift with high severity, which usually demands a full rebuild of the prediction models.
- Gradual Drift. The transition between $P_t(\omega|x)$ and $P_{t+1}(\omega|x)$ randomly intercalates $P_{t+1}(\omega|x) \neq P_t(\omega|x))$ and $P_{t+1}(\omega|x)=P_t(\omega|x))$ observations. Ensemble and adaptive learning may gradually update the prediction models since the early evidences of distribution changes are revealed
- Incremental Drift. This is a particular case of Gradual Drift but with lowest ratio of changes, where the real distributions slowly but increasingly evolved to different ones. The scenario has not high statistical relevance, being easily countered by adaptive learning procedures

- Recurring Drift. The distribution changes periodically change by following stationary patterns. For example, how a network usage changes may vary based on the hour (e.g. form working to non-working time periods) or month of the year (e.g. from holydays to high end-user concurrency months). This can be solved by adaptive learning, but in a most cost-effective manner, by switching between models according to the stationary momentum.
- Blips Drift. Defined as outliers, blips can occur randomly due to diverse situations: instrumentation errors, connectivity issues, etc. As suggested in [81] blips should be treated as outliers and should not have any long-term impact on the predictive functioning (note: not to be confused with discordance observations when anomaly-based IDS, where the outlier refers to the observation itself -thus having contextual meaning- and not to the model distribution on which they may be differentiated – as referred within the concept drift context-)

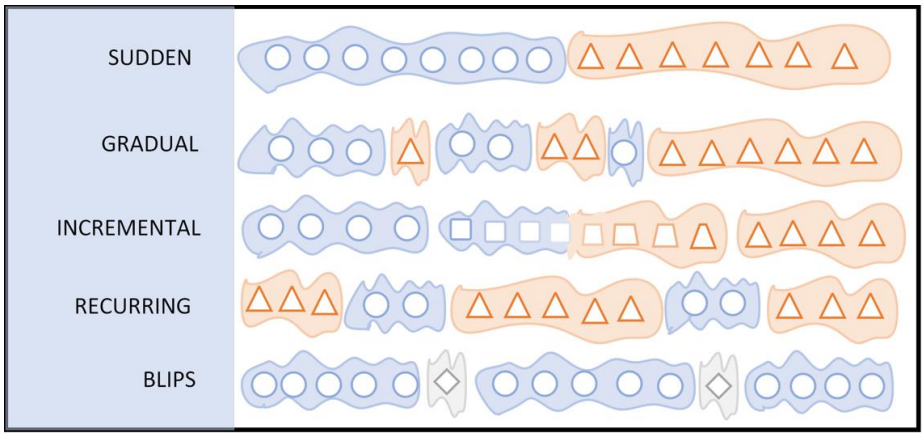

Figure 1: Concept drift situations based on severity and speed [81]

It is remarkable that concept drift situations may be linked to the normal (and legitimate) use of the protected systems and networks, but also to malicious act. The latter are usually related to the Adversarial Machine Learning paradigms, where intruders may attempt to thwart conventional learning (e.g. mimicry attacks, thus making threats resemble line normal events at both network [83] but also local [84] levels) or adaptive learning capabilities (e.g. poisoning attacks, thus forcing remodeling in order to hide further offensive actions [85])

2.3.2 Impact of Concept Drift in different cyber threat modeling approaches

The concept drift may affect in different way on learning for modeling with prediction purposes, which in the context of cyber threats may impact in different ways based on the adopted knowledge building paradigms. The following reviews some cases on widely adopted AI-supportive learning and application paradigms: supervised, semi-supervised, unsupervised, reinforcement, transduction, multi-task, federated and frugal learning.

Concept Drift on supervised learning

Building models for cyber situation prediction on supervised learning means to assume labeled inputs (known) correctly tagged, from which new observations are expected to be inferred at different time horizons (t+1). Given the great amount of reference information this entails, they tend to build accurate approximations with less tendency to emit false positives and more robust against adversarial AI attacks than other solutions. Another benefit is that supervised learning for prediction facilitates the participation of the human operator –in the loop; thus, taking advantage of its experience in optimization procedures (see Frugal learning) but also making the prognosis algorithms more Explainable and Interpretable for human beings. Its main drawbacks are difficulties when obtaining reliable samples properly labeled, and the fact that the reference datasets usually contain fewer samples of some classes (e.g. “normal” observations) than from others (e.g. very specific intrusion indicators), which leads to risks related with imbalance between classes [86]. In

this context, operating with massive information with full tagged and structured data may be also a great challenge, to which cognitive bias and mislabeling can lead to unexpected errors that are difficult to detect and interpret [87]. Supervised learning has been widely adopted by cyber prediction tools relying in symbolic AI, usually enabled by decision trees, SVM, Artificial Neural Networks, deep learning, etc. being proven very effective in situations that among others demand to anticipate malicious network traffic situations [88], management of contextual information from human feeds [89] or revealing future attack steps by predicting on system call sequences [90].

The presence of concept shifts and drifts when predicting cyber threats in a supervised learning way require mechanisms that allow forgetting dated information while the model is adapted to more recent observations. It is assumed that previously processed inputs were correctly labeled based on the previous concept but not valid to the current one, so they need to be dismissed at the same time that novelties are introduced into the updated forecasting models. This is particularly useful in the management of anticipatory threat perception as the protected environment evolves. For example, and under certain local network topology and service supplying structure, observations which lower severity (e.g., quiet enumeration, information request, etc.) may be label as unharmed or deprioritized when only insider actors may access to the asset of the organization. But if the same organization includes additional ICT assets and allows external actors to operate in some services (e.g. web services for allowing customers online transactions), this labeling may change, hence demanding to update their tackled prediction models. Another and more recent example is illustrated in how the wide spreading of teleworking practices leaded to readjust the conventional detection models [91], leading to update the set of aspects (and their severity) to be taken into consideration in order to anticipate threat situations, like data leaks or man-in-the-middle scenarios.

Concept Drift on unsupervised learning

The unsupervised-based prediction capabilities do not consider categories a priori (i.e. the data adopted to construct forecasting models is unlabeled). Instead, they manage the monitored data as a set of random samples, from which it is possible to build density models and defining their projections over time [92]. The state-of-the-art typically adopted this paradigm for 1) anticipating and analyzing anomalies [93] which are typically referred to as outlying deviation from the normal (and hence “legitimate”) operation of a system/network [94]; and 2) segmentation, which assist the discrimination of potential situations according to different criteria (priorities, nature, impact, etc.), being particularly adopted for user profiling [95]. These have proven high effectiveness when automatizing processes, thus reducing the human intervention in proactive SOAR (security orchestration, automation, and response) cycles. For decades, this also allowed CTI analysis to dig hidden patterns able to anticipate intrusions, showing less complexity compared to the supervised learning tasks, and taking advantage of the fact that having unlabeled data is significantly more frequent data acquiring labeled samples. On the contrary, concept drift situations on unsupervised projection may amplify the natural drawbacks of unsupervised learning, like dealing with the unknown of the observations and the inferences, lesser accuracy even at short forecasting horizons, and a costlier interpretation of the results, which are more dependent of alert correlation systems [96].

Concerning the first application (dealing with anomalies), concept drift may lead to disharmonizing situations where the “normal” from the classification/prediction models differs from what is really “normal“ based on the current protected system/network usage; thus heavily penalizing the intrusion detection and response effectiveness due to increasingly false positives and negatives [97]. This is very often in raw non-stationary operational environments (as is the case of the cyberspace), leading for example to confuse massive confluences of legitimate users into online services (typically referred to as flash crowds) with Distributed Denial of Services (DDoS) or Economical Denial of Sustainability (EDoS) scenarios [98]. Similarly, concept drift situations in predictive segmentation may lead to wrongly estimate attacking profiles and each the intruder next step inferred from them, in this way reducing the proportionality of the decision made, decreasing effectiveness and even revealing hardening and additional attack surfaces resultant from the mistakenly enforced

countermeasures. This may be exploited by skilled attacker trying to evade/thwart Intrusion Detection and Response systems like honeynets [99] or web-based segmentation [100].

Concept Drift on Semi-supervised learning

Unlike supervised nor supervised learning, semi-supervised learning for threat prediction embraces a mixture of labeled and unlabeled samples, which often are considered at different processing stages (typically, an early unsupervised step able to define a prognosis on unclassed observations and then a supervised stage for revealing practical information from the predictions [101]. For example, in [102] semi-supervised learning and LSTM are combined to predict the malware labels of nodes at successive time intervals in the neighborhood of a context-specific environment with critical assets and links involved. As indicated in [103] the primary goal of semi-supervised learning is to harness unlabeled data for the construction of better learning procedures, resulting in a smoothening of the benefits and cons of adopting supervised on unsupervised learning separately. However, these authors pointed out the difficulty in adopting unsupervised procedures for solving regression problems due to the fact that their value spaces are continuous (in contrast with classification, where a categorical value space facilitates dealing with outliers and never seen before data). Many such methods model a real-valued function in an intermediate step and incorporate the real-valued predictions in a regularization term in the objective function. These real-valued predictions can be readily utilized in the regression scenario. This has several applications in the cybersecurity context, ranging from analyzing binary file appearance logs of machines to predict which machines are at risk of infection months in advance [104], intrusion prediction on Cyber Physical Systems (CPSs) communication networks [105], or end-to-end insider threat prediction, through which it is possible enhance time response to prevent harmful influences of insider threats in advance [106].

The concept drift implications on the semi-supervised prediction schemes derive from their apparition at some (or even both) of its data processing stages (supervised or unsupervised analytics). However, semi-supervised methods combined with prediction/classification ensembles may facilitate the identification and selection of the important data instances to mark labels and achieve a comparatively higher prediction performance using just a small amount of labeled data without any extra overhead [107], which may apply in most of the concept drift mitigation procedures presented in the next Section. On the other hand, proposal like proven the potential of semi-supervised approaches for enabling 1) to eliminate dependency on the labels of emerging records; 2) fast incremental learning with real-time update ability to tackle concept drift; 3) efficient novel class detection ability to tackle concept evolution [108]

Concept Drift on Reinforcement Learning

Reinforcement learning is a branch of machine learning where an agent tries to learn from the environment by interacting with it. The agent is rewarded for its performance on the task [109]. It has been used in many applications including playing go [110], optimizing hardware resource management on computers [111] and, of course, in the cybersecurity field [112]. In 2019 Caminero et al. designed an IDS that uses reinforcement learning to detect intrusions in a network [113].

Although reinforcement learning agents could have some adaptation capabilities to concept drift due to their plasticity, which is the agent's ability to change over time, this plasticity may cause over reaction to noise and even forget its previous knowledge (catastrophic forgetting) due to big learning rate, a lower learning rate may cause the agents been unable to adapt to non-stationary conditions which is known as entrenchment effect. This also known as the plasticity-stability dilemma [114] [115].

In 2018 Han, Yi, et al designed a system for autonomous defense system for software defined networks [116], in this publication the researchers designed a system which monitors the network traffic and uses reinforcement learning to detecting intrusions and can isolate compromised nodes or migrate the critical parts of the network to prevent the attacker from reaching them. Although this approach manages to adapt to changes in an attacker with evolving strategies, the results vary depending on the scenario, with cases in which the algorithm is not able to converge. Another problem with this approach is that the space to represent the scenario grows linearly with respect to

the number of network nodes as well as the number of attack types that the opponent can perform, so it is currently not suitable for representing a real network scenario with a wide set of intrusion techniques.

Concept Drift on Transduction

Learn by transduction is a modification of Support Vector Machine for classification of objects that not only returns the prediction but also returns a confidence value of the prediction [117]. As a variant of SVM it can be considered a supervised learning algorithm so the same considerations and problems for handle with concepts drift in supervised learning apply for transduction.

Concept Drift on Multi-task Learning

Multi-Task Learning (MTL) is a subfield of machine learning in which multiple tasks are solved at the same time while trying to capture the intrinsic relatedness between those tasks [118]. This is motivated by the idea of what is learned for a task can help other tasks be learned better. This is very helpful in the data sparsity problem when the task has no sufficient data to be trained accurately as the multi-task learning aggregates the data for all the task in the spirit of data augmentation so it can increase the accuracy of each task [119]. For capturing the relatedness between tasks there are different approaches sharing the information between the tasks like feature-based MTL, parameter-based MTL and instance-based MTL. Feature-based MTL tries to learn common features among the tasks in order to share the knowledge between them. This can be done by finding a feature common representation for all the task by transforming the original feature representation in the dataset o by selecting a subset of the original features as the common representation for the task. Many approaches can be performed for sharing parameters between tasks, like share hidden layers in neural networks between tasks while keeping separated the output task-specific layers [118], or grouping the tasks in clusters of mutually related tasks [120]. There are few works dealing with instance based MTL but we can use [121] as a representation of this approach. In this work they try to estimate the density ratios between probabilities that each instance belongs to its own task and also to a mixture of all the tasks and then use these ratios to determine the instance weights and learn the model parameters for each task. The knowledge sharing capabilities of MTL can be used for network analysis and malicious traffic detection as the extraction of similar features from task with large amount of data can help to the training of tasks with shorter data allowing to avoid the class imbalance problem [122]. Also, these knowledge sharing capabilities makes MTL well suited for identify and forecast the next steps of complex attacks like in [123]. As concept drift may cause the similar issues with multitask learning that in the previous machine learning schemes, as there is no guaranty that trained task specific models or knowledge sharing mechanism can handle the drift without adaptation later it will be seen the similarities between multi-task learning and the passive adaptation mechanism of ensemble learning.

Concept Drift on Federated Learning

As a succinct definition, Federate Learning (FL) is the decentralized taste of machine learning dealing with distribute and private datasets. A more elaborated definition of Practical Federate Learning was provided by Li et al. as follows [124]:

"In a federated learning system, multiple parties collaboratively train machine learning models without exchanging their raw data. The output of the system is a machine learning model for each party (which can be same or different). A practical federated learning system has the following constraint: given an evaluation metric such as test accuracy, the performance of the model learned by federated learning should be better than the model learned by local training with the same model architecture."

It provides a couple of specific advantages: First, ensuring privacy principles by design, since, for example, on this variation of ML data remains on the clients' device. Second, inference takes lower latency, because the updated model can be used to make predictions on device. Third, it has been reported to provide smarter models, given the collaborative training process [125]. And finally, it requires less power consumption, as models are design to be run on computationally limited devices. Depending on the distributed learning strategy, FL can be classified as Horizontal, where data

distribution is being split among clients, Vertical, where feature space and models are the target for the distributed strategy, or Hybrid FL, a mixture of both VFL and HFL. Some of the security challenges that FL has to deal with are related to the major attacks identified. In the case of FL these are poisoning attacks and inference attacks [126]. The privacy preservation strategies followed principally on federated learning systems are threefold: 1) Differential Privacy [127], 2) Secure Multi Party Computation [128], and 3) Homomorphic Encryption [129]. The communication strategy is another important dimension on FL with multiple variations of synchronous and asynchronous centralize or distributed models but there are many others. An initial attempt to provide a framework suggested the following dimensions, see Figure 2 [130].

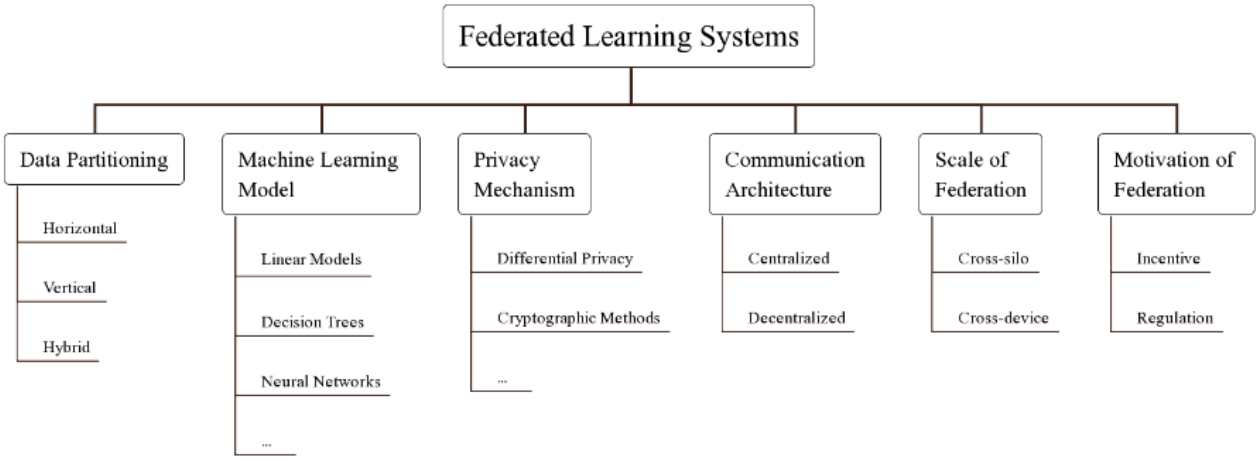

Figure 2: Taxonomy for Federated Learning Systems [84]

The most usual strategy followed in FL is the Horizontal, some of the most notable algorithms applied are Federate Average (FedAvg) [131], Federate Matched Average (FedMA) [132] and Federated Stochastic Gradient Descent (FedSGD) [135]. FL is being extensively explored in the field of cyber threat detection. DeepFed outperformed state of the art performance on various types of cyber threats on industrial environments [133]. It has been proposed in the context of Advanced Persistent Threats [134]. Even, a community sharing approach has been proposed that federates learning through the merging of models without the need to share sensitive cyber-logging data [137].

Regarding Concept Drift, Federate Learning presumes the same problems than their non-federated counterparts, previously presented on Supervised Learning, Unsupervised Learning and Reinforcement Learning sections. As some authors highlight, concept drift represents a major disadvantage on the operation of FL models [138]. There are some additional disadvantages on the use of FL related to concept drift since most of the FL algorithms lays on the supposition that the process generating the data is stationary. Some solutions have been proposed to make them adaptive. For example, FedAvg comes without convergence guarantees and can diverge in practical settings when data are heterogeneous [139], the adaptive-FedAvg algorithm, instead, makes use of a passive adaptation strategy to promptly react to any change over the input distributions and also improve the accuracy in stationary conditions[136]. To tackle with Concept Drift on asynchronous FL, Chen et al. proposed FedConD framework to detect different types of drift, balance local updates and control the number of clients performing local training in order to obtain a more dynamic and fairer global model on the server [140].

Concept Drift on Frugal Learning

The latest wave of hype around AI has been driven, in part, by the widespread availability of data. The *Imagenet* moment in AI is known as the one where the famous image classification contest became a competition between companies where the most powerful models that were trained on the most data, via augmentation techniques, won the competition. The problem with this approach

is that it is neither efficient nor scalable, and it has implications for both the production of environmentally friendly AI that is fair and equally accessible to people. Recently, a trend has emerged that seeks to produce new models inspired by human plasticity and its ability to learn a concept from a single sample (one-shot learning). Contrary to other approaches that propose transfer learning and the extraction of knowledge from previously trained models via model distillation [141]. Other methods seek to obtain a refactoring of the same and decomposition of the sample space from transformations; among the techniques used are the Flexible Multilayer Sparse Approximations and butterfly factorizations. A disadvantage of these models is that they are complicated to train and pose their data domain since the feature engineering process involves a transfer of the knowledge domain [142][143]. Another approach is Tiny ML, which seeks to adapt the models to more resource-light and energy-efficient options in order to be able to apply the learning process in embedded devices. Disabato and Roberi research sought a passive adaptation mechanism to concept drift by combining a classifier from a deep neural network and unsupervised methods (KNN) [144]. Another approach from the active point of view based on meta-learning applies a fine-tuning strategy to adapt the distilled models to the data stream [145]. Finally, another possible approach to the Concept Drift detection problem from an efficient point of view would be from the Exponential Weighted Average [146], which is a lightweight metric that has been demonstrated useful to characterize data input streams.

2.3.3 Adaptation to the concept drift

The state-of-the-art has developed and still researching in novel applications for concept drift mitigation and prevention, some of the earlier efforts summarized in [147]. Although this review deepened into their application for assisting anomaly-based detectors, their basis can be transposed to the learning procedures usually considered during cyber situation prediction. Accordingly, three great clusters of initiatives can be distinguished: those based on model prediction, and those that do not require to predict the drift/shift itself but constantly update the prediction models as the operational environment evolves. In [148] they are referred to as Active and passive adaptation methods, where:

Active Adaptation groups the solutions facing the concept drift by reactive or proactive (anticipative) response once a concept shift is detected or forecasted. Alternatively, a response may be addressed if the prediction errors exceed certain thresholds, thus suggesting that the models and/or algorithms could be inaccurately calibrated. In [149] the reactive response is referred to as *detection and response;* once a significant change is discovered, the system discards the obsolete knowledge and adapts to the new environment. The knowledge adaptation may be instant (a large retrain based on the new observations) or constant (minor changes are conducted when early traits of a concept drift are detected/predicted). The detection and response process mainly leads to three reconfigurations:

- Adaptation by Windowing. Being the most popular in the state of the art, during the adaptation by windowing a slicing windows selects a subset of the samples within datasets and replace them via FIFO (“First-In, First-Out”), hence the more recent observations in the dataset are disposed; or LIFO (“Last-In, First-Out”), which in the opposite, replace the oldest. The selection of FIFO or LIFO slicing should depend on how the original models where built.
- Adaptation by Weighting: This procedure explores beyond the temporality of sampling by assigning and managing feasibility scores on the samples that shall guide the development of prediction models. The better the weights, the greater relevance; where the weighting may vary based on the dataset management criteria and early model building assumption, thus considering features like temporality, criticality, trust in the sources, etc.
- Random sampling: In opposite with the previous approaches, the replacement of random observations may enrich the diversity and uncertainty on the resulting models; being specially interesting when high computation capabilities are available and multiple what-if situations may explore a wider tree of potential solutions

The Figure below introduces a general framework for active adaptation to concept drift, which was presented in [150]. Assuming streamed data, the predictor is expected to build forecasting models at Training and Learning. The authors indicated that “*this stage should aim to abstract the retrieved data and extract the key features containing sensitive information, that is, the features of the data*

that most impact a system if they drift". On these models the Predictor will conduct a prognosis, which resulting accuracy combined with other field observations may be able to guide the framework on detect concept drift situations. If a drift is revealed, the situation will be analyzed, leading to the most suitable adaptation procedure.

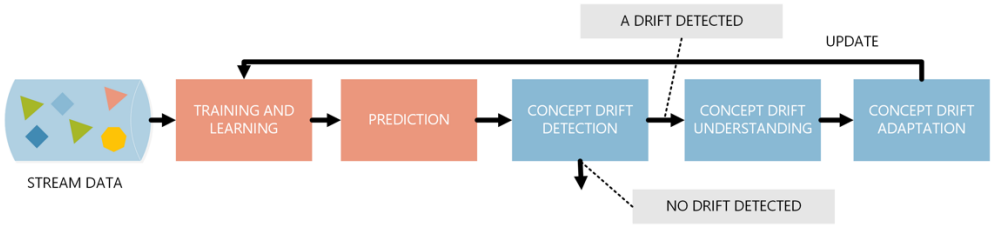

Figure 3: Framework for passive adaptation to concept drift

Passive Adaptation groups solutions that do not rely on the previous identification or prediction of concept drifts/shifts. In the grounds of emerging Machine Learning paradigms like adaptive learning or reinforcement learning, these solutions tend to embrace the premise that there are constants changes in the environment, that implicitly may be discovered and re-interpreted by the learning algorithms, resulting in cumulative fluctuations on the regression models. According to [151], the different contributions to this field can be grouped into two categories: those that focus on updating a single prediction system, and those that affect systems that integrate different predictors (typically, ensembles). The first are more efficient, and therefore more recommended for systems operating in real time. This is illustrated in [152], where decision trees are applied for the analysis of information sequences. Another example is [153], where fuzzy logic and methods based on the use of sliding windows on reference samples are combined. Finally, in [154], Extreme Learning Machines (ELM) strategies are applied on neural networks whose adjustment varies over time. On the other hand, sensor ensembles have been shown to behave much more stable than individual predictors in stationary monitoring environments. This is because they tend to compensate for the error of the worst performing predictors in each use case. In addition, they facilitate the incorporation of new data into the models they have built and provide strategies for discarding less relevant information. In some circumstances, and as occurs with random sampling, the diversity of sensors in a combination beneficially affects its results. An example of the application of different levels of diversity in order to improve adaptation to the environment is illustrated in [155]. In [156], some of these proposals for sequence analysis are brought together, and a new technique based on weighting and decision trees is proposed.

Based on the above, the adaptation of cyber threat prediction models can be sustained on the cornerstones described in [157]: Memory, Change Detection, Learning and Loss Estimation.

- **Memory**. Beyond mere samples extracted from the operational environment, the knowledge needed for predicting cyber situations also demands information about potential attack surfaces, actors, vulnerabilities, and any other Cyber Threat Intelligence (CTI) source. A proper adaptation may demand not only the capability of update this knowledge, but also forget old or poisoning data. The update of the memory may be enabled by different approaches, ranging from those able to incorporate individual data (typical from transduction and/or online learning) or multiple knowledge as a whole. The methods described above (windowing, weighting, random sampling) may be valid for both simple or multiple information onboarding. On the other hand, forgetting may adopt several schemes, usually LIFO/FIFO in a gradual (e.g., reinforcement learning) or abrupt way.
- **Change Detection**. As the system works with evolving data that changes over time it is desirable to include a mechanism that detect when those changes occur and distinguish them from simple noise. Although machine learning algorithms can adapt themselves to those changes over the time, their performance and adaptation time are boosted by the use of change detection systems. This component identifies the concept drifts by locating the

change points (small-time intervals) when these changes occur. In [157] there are proposed different methods to achieve this:

- Sequential Analysis: These methods are based on statistically measuring the deviation of the input data from the mean and detecting that a change has occurred if this deviation exceeds a certain threshold value.
- Control Charts: Those systems are based on calculating the error-rate of the system, two desired confidence levels for the error rate and defining three states based on the confidence levels:
  - In-Control: The error of the system is stable
  - Out-of-Control: The error has increased significantly. This means that the recent data becomes form a different distribution than the previous one (a change in the data).
  - Warning: This state is between the previous. This indicates that the error is increasing, this does not necessarily mean a drift but may be caused by noise in the data.
- Monitoring two distributions: These methods are based on comparing the data between a sliding data window with the most recent input and a fixed data window with the past information and comparing the difference between the two data distributions to detect a change.

- **Learning**. The learning component refers to the set to technics used to update the model as the data coming to the model evolves. There are two main key points when updating the models and these are deciding when the model is going to be updated and how it is going to be done.
  For deciding how to update the model when new data is available, whether retraining the model by merging the training data set with new labeled data to create a new model or incrementally updating the model each time new examples incomes to the system. Each of these approaches has its own pros and cons. Retraining the model from scratch will make it more resistant to noise but will have a higher compute and memory cost than the incremental approach. For deciding when update the model according to the changes in the incoming data. There are two main approaches to trigger the model changes one is the blind approach that updates the model as new data incomes regardless of whether the data changes as in the incremental update strategy. The other main approach is the informed approach that relies on the model update whether a trigger has been flagged those triggers may be change detectors discussed above or explicit trigger designed for the model.
  In the case of ensemble learning when multiple models are combined to make a prediction [158] there is also the need to update the combination mechanism that merges the predictions of each of the models to generate a single prediction. To achieve this, we can change the combination rules as the data evolves, update the models, by retraining them, or managing the set of models adding or deleting them from the collection over the time based on their performance on the evolving data. These three technics are not mutually exclusive so the three of them can be used at the same time.
- **Loss Estimation**. This is a bad prediction indicator that measures how bad the model's prediction was for a single example. There are two ways for calculating this measure, model dependent way or model independent. The idea for model dependent loss estimation is to recognize concept changes using Support Vector Machines for estimating the leave-one-error. The leave-one-error is a method that for each training example sets leaves one of the cases out of the set and retains the model with the rest of them and then uses the leaved out case for testing the trained model, if the prediction for this case is incorrect then it's counted as a leave one out error. The number of the leave one out errors divided by the number of training example sets is the leave-one-out estimation [159]. The leave-one-error is a very expensive estimator is a very accurate method but also is very expensive. Instead of using leave-one-out estimation the more efficient ςα-estimate can be used [159].
  The idea for model independent loss estimation is based on maintaining two sliding windows: a short window containing the most recent data and a large window containing a larger set of reset data. Then if the error estimation increases significantly in the short windows, but it maintains the level in the long data windows it points to a concept drift [160]. Also, using

fading factors a Page-Hinkley (PH) test [161] can be used to monitor the evolution of ratio of both estimators and signally a drift when this ratio increases significantly [162].

2.4 Predictable attacks/threats

Attacks and threats can sometimes be foreseeable by the fact that they generally follow a given sequence of actions carried out by the attacker. Based on this principle, Lockheed Martin has adapted the military concept of Kill Chain, describing the structure of an attack, to the world of computer security. Thus, appeared the concept of Intrusion Kill Chain, allowing the modelling of intrusion processes in an information system. Although sometimes criticized, this methodology was quickly adopted by many security experts.

The Intrusion Kill Chain[163] assumes that an attack can be split into different phases in which it is possible to intervene, to stop it or minimize its consequences. These phases cover the entirety of a cyber-attack from reconnaissance prior to data exfiltration, as follows:

- **Reconnaissance**: identification of the target.
  - The attacker selects the target, researches it, and tries to identify vulnerabilities of the target, and the means available to achieve its ends. It may include harvesting email addresses, identifying employees on social media networks, collecting press releases, contract awards, conferences attendee list, discovering internet-facing servers, or gathering of any information potentially relevant to facilitate intrusion.
- **Weaponization**: preparation of the operation.
  - The attacker prepares a malware (virus, worm, etc.) exploiting the knowledge acquired in the previous phase (e.g., vulnerability of identified used component).
- **Delivery**: launching of the operation.
  - The attacker conveys the prepared weapon to the target, either by controlled delivery (direct against web servers) or by released delivery (malicious email, USB stick, social media interactions, etc.).
- **Exploitation**: gaining access to victim's information system.
  - The attacker must exploit a vulnerability to gain access. Once delivered, the weapon's code is triggered, exploiting vulnerable software, hardware or human.
- **Installation:** establishment of a beachhead at the victim.
  - The malware installs a persistent access point to the victim information system (backdoor), the attacker will exploit.
- **Command & Control**: providing access to the victim's environment.
  - The malware allows the attacker to gain remote access to the victim's information system.
- **Actions on Objectives**: achievement of the mission's goal.
  - Having complete access to the victim's information systems, the attacker can perform the desired actions, including collecting user credentials, destruction of the system, overwrite or corrupt data, collect and exfiltration of data, encryption of data for ransom, etc.

Modelling attacks according to this model therefore describes in details the various necessary steps followed by an attacker, regardless of the type of attack. The good knowledge and understanding of the steps followed by an attacker offers an undeniable advantage to the defenders, to put in place, the different strategies, technologies to thwart these attacks. Thus, for each of the aforementioned steps, defence strategies can be determined and implemented.

- **Reconnaissance:** early detection of any attempt to collect information by attackers.
  - In order to reduce or prevent the acquisition of sensitive information relating to the information system, it is necessary, for example, to identify publicly or easily accessible information on the organization, the employees, the information system and take corrective actions as a result (e.g., delete the most sensitive information), regularly check the access logs, to identify questionable connections as quickly as

possible and or even use tools for detecting suspicious actions that may be linked to the recognition phase of an attack (domain or network scans for example).

- **Weaponization:** identify vulnerabilities and security measures necessary to secure resources**.**
    - To reduce the attack surface and the weaknesses of the organization, it is necessary to identify any existing or new vulnerabilities that could be exploited by an attacker, to put in place any security measures that prove necessary, but also to exploit threat intelligence, in order to adopt a proactive, rather than reactive, approach in terms of discovering new vulnerabilities.

- **Delivery:** detect and prevent the installation of malicious code.
    - In order to limit the risks of malicious code execution, different strategies can be adopted: implementation of technical solutions (firewalls, anti-virus, anti-spam, network monitoring) or organizational (employee awareness).

- **Exploitation:** prevent the execution of malicious software.
    - For vulnerabilities that have not been identified and / or that have not been the subject of special protection, various controls can be put in place to prevent the execution of malicious software, including the use of SIEM (Security Information and Event Management) software to identify any suspicious activity, the use of EDR (Endpoint detection and response) type software, educate users, etc.).

- **Installation:** detect an attack to prevent access from being established.
    - Once the malicious software has been executed, it is still possible to prevent the attacker, by setting up defence mechanisms such as HIPS (Host-based Intrusion Prevention System) capable of blocking or signalling any installation attempt, by managing access rights, setting up strong authentication, etc.

- **Command & Control:** detect and interrupt communication between the target and the attacker.
    - If despite the various measures in place, the attacker was able to take control of the system, various measures can be used to detect and terminate it, in particular by monitoring the network (Network Intrusion Detection System), DNS sinkholes, etc.

- **Actions on Objectives:** stop the attack at any cost.
    - Finally, if any attempt to stop the attack has failed, measures taken upstream can reduce or prevent the attacker from achieving his goals, many solutions are possible, among which: data encryption, backup local data, incident management, segregation of sensitive information, etc.

Thus, to deal with each of the stages of an attack, as defined by the Cyber Kill Chain, six types of defensive actions, namely: Detect, Deny, Disrupt, Degrade, Deceive or Contain. This means of defence can be put into perspective with each of the phases of an attack in the form of a so-called Cyber Kill Chain Control Matrix, allowing to formally identify controls in place to deal with the attack phase and categorize them in terms of types of defensive actions. As pointed out earlier, although the concept of Kill Chain is widely accepted, a certain number of limitations have been formulated, particularly in relation to the temporal aspect of the attacks; where an attack does not necessarily stop after a cycle but can persist. In this sense, the FireEye kill chain highlights the persistence of threats[164]. Other critiques of this model focus on the external scope of the first phases and the lack of possibility of actions for the defender, the focus on malwares, as well as the only consideration of external threats, thus ignoring any internal threats.

MITRE in its ATT&CK framework[165] proposes a structured list of attackers known behaviours in terms of tactics, techniques and procedures. This information is presented in different matrixes of

tactics divided in specific techniques, corresponding to a type of attack. Each technique is detailed with explanations, examples, references and suggestions in terms of detection and mitigation.

To overcome the critiques made on the Cyber Kill Chain, Paul Pols proposed in 2017, the Unified Kill Chain[166], that combines elements from both the Cyber Kill Chain and MITRE ATT&CK framework and details the tactics that form the building blocks of cyber-attacks by Advanced Persistent Threats (APTs). This model offers significant improvements over the limitation of the Cyber Kill Chain while benefiting of large knowledge base of MITRE ATT&CK framework.

2.5 Methods for Validation and Evaluation

With the growing number of risks, an appropriate detection framework becomes a necessity. Although automated systems are increasingly being used, the risk of "false / positive results" is still present. This is the reason why validation and evaluation (validation and verification) are crucial for the framework concerned.

Methods for ensuring that their solutions comply with security, cyber security and privacy (SCP) requirements are becoming increasingly important for automated system manufacturers and component suppliers. However, in analysing and characterizing validation and evaluation methods for SCP of detection frameworks, we found that existing classifications are missing or not suitable. At the same time, the descriptions of existing validation and evaluation methodologies are usually not clear enough to help users decide upon how to best classify such methods and to select the best option. If these problems arise, then the selection and use of validation and evaluation methods for detection systems can be less effective, ultimately impacting the cost and dependability of a system. The ultimate goal here is to create clear classification of validation and evaluation methodologies that will help us to protect our framework. For this purpose, more approaches are needed, which could be used for a more comprehensive result of validation and evaluation. The different validation and evaluation approaches addressed to cyber prediction capabilities are usually grouped into three categories, which assume as classification criteria the technologies in which the evaluation procedures rely on: injection, simulation and testing. They are briefly presented below.
On the other hand, the verification and validation condition may vary on different indicators, like hit rate, false positive rate, effectiveness at different time horizons, effectiveness according to the data granularity, strengthening against adversarial tactics, performance, etc. These conditions shall be adequately identified and framed according to the operational needs: for example, in some circumstances the capability of digesting large amounts of information without significantly decaying the prediction performance may be more relevant than the capability to adapt to variations in the traffic entropy. Another example can be illustrated on the level of restrictiveness in which a prediction-based IDS may operate, sometimes maximizing the hit rate while penalizing false positive rates (high restriction), or in the opposite by minimizing to issue false positive at the cost of reduce the capability of discover threats. The latter can be easily measure by Receiver Operating Characteristic (ROC) analysis.

2.5.1 Injection

Injection-based validation and evaluation methods focus on introducing certain characteristics in a system, providing a certain type of input or triggering certain events to confirm that the system behaves suitably under the corresponding conditions. Most common injections are as follows: code injection; CRLF injection; Cross-site Scripting (XSS); OS Command injection; and SQL injection. In other words, the injection may lead to errors that, subsequently, may cause unexpected behaviour of the target component. These errors may boost in the system and subsequently cause failures in other components or even system failures. Validation and evaluation of injection can be used in different phases of the framework development to evaluate (or even predict) how framework as well as specific components behave. During the verification phase, we shall check every entrance where user (or potential attacker) could insert data.

2.5.2 Simulation

Simulation enables early validation and verification of framework. We can either create our virtual environment or use digital models that behave like real-world systems. Simulation-based validation and verification methods provide virtual validation in testing framework. Possible issues in automated systems can be experimented and analysed through simulation. This specific type of validation and verification could provide early vision how framework work.
Among others, simulation methods enable running test without dealing with expensive hardware or test equipment. Simulation can be from our last real scenarios or even some scenarios which we predict thank to some research. At the same time, simulation-based test approaches do not introduce direct safety risks in cases where human-machine interaction exists. However, the effort and cost of the development of simulation and its test processes can be high. The trade-off between simulation accuracy on the one hand and simulation speed, resource consumption and effort for constructing simulation models on the other hand has to be considered.
A major advantage of simulation is that validation and verification can be conducted without producing any physical assets and adding risk to the environment. However, simulation-based applications mostly run on hierarchical models. This narrows the availability of both academic and industrial resources in development. Simulation tools can require significant computational power and limit real-time applications.

2.5.3 Testing

Those types of validation and verification methods are important to prepare two types of fundamental information. First is input data which could be either data already collected or even data which could be generated by prediction software or some expert analysis. Second is „description of the expected output or behaviour “. To run a test, real version or some testing version of our framework shall be prepared. For testing, we can use the final version of framework as well as any artefact used during its development, such as models or specific hardware or software components. For security purposes (in case of testing final version of framework), we need to have some testing environment which is the copy of the real one. After tester insert data to the testing environment, it is necessary to evaluate the result and to compare them with our expectations. This method is useful (suitable) for early detection of some misconfiguration or after some upgrades. Testing should be executing every time something changes or once new attack techniques evolve, as new unexpected error can occur anytime.
We could use “black-box” or “white-box” testing. Every option has some advantages and disadvantages, but always depends on our testing goal. Or we can combine the above mentioned. In this case, we are talking about so called “grey-box testing”. However, we must add that testing is usually incomplete. Even successfully passing a large set of test cases does not guarantee one 100% protection. From time to time, there could be some really unexpected misconfiguration for years. This is a reason why testing shall be done regularly and not only at the beginning or after some big upgrades.

Chapter 3 Prediction and Relevant Existing Solutions

In this section, the most relevant solutions, systems, protocols and tools to cover the different phases defined within the threat intelligence cycle to manage the available information, will be described. These tools can be useful to get information, enrich the data and share it efficiently between different entities.

In today's scenario, where cyber threats are continually increasing in sophistication and cyber attackers have ever more resources at their disposal, it must consider how important gathering and sharing security incident information is. For this reason, transformation and normalization techniques gain special relevance, since they allow for an easy sharing of data as well as its standardization, structuration, and formatting so it is easier to recognize and extract information.

Among all these techniques that make it possible to respond to security incidents more quickly and efficiently, can highlight the importance of data processing standards, which will be used to structure, normalize, store, enrich, share and present information. In the evaluation process of the different tools analyzed, it will take into account that their method of managing data is compatible with one of the following reference standards.

All the tools that will be analyze in this section can be grouped into four different categories according to the purpose for which they have been developed. These categories are CTIP (Cyber Threat Intelligence Platform), SIEM (Security Information and Event Management) SOAR (Security Orchestration, Automation and Response), and prediction frameworks (that will be analyzed using the SWOT methodology).

Below, some existing solutions and tools will be evaluated and classified into one of the previously mentioned categories, in order to identify their relevance for this project.

3.1 Reference Standards

The reference standards allow the different COTS solutions to collaborate, as well as to share information usable by end-users and response teams. Among the different standardization initiatives, the following has been particularly recurrent in the threat prediction bibliography.

A reference standard will help with the integration of the different components that compose the solution, making it able to automatize the process of training the prediction models with the data collected from the CTIP and the SIEM.

3.1.1 Common Event Format (CEF)

Logging and auditing file format developed by ArcSight that aims to standardize the data produced by the tools. CEF is proposed as a text-based format and its main strengths lie in its ability to support a variety of device types offering the most relevant information and ease to be extended. However, CEF neither regulates nor defines ID of the events produced by devices, something which must be accomplished by the device or application.

3.1.2 Incident Object Description Exchange Format (IODEF)

Computer security information sharing incident format where messages are presented in a human-readable way. It provides information about services running, networks and hosts. IODEF employs XML, a language that define a framework for data encoding. The aim of IODEF is to improve communication between entities (typically CSIRTs) by sharing data of incidents in a structured way. It provides a unified, common layout for information sharing. Normalizing security information eases while, at the same time, requires less resources to handle and process data concerning incidents.

3.1.3 Structured Threat Information eXpression (STIX)

Described as a standard and structured language for the purpose of defining, storing and sharing incident and cyber threat information. The main advantage of this behavior of sharing information lies on the fact that it strengthens information net and provides analysts with tools to fight against incidents and recognize attack patterns. The STIX architecture should then include various kinds of pieces of data such as cyber threat actors and initiatives, tactics, techniques, and procedures (TTPs). One of the biggest advantages of STIX is that is transport-agnostic. The absence of a dependency on a concrete transport mechanism for its structure and serialization makes it flexible and adaptable.

3.2 Cyber Threat Intelligence Platform (CTIP)

A Cyber Threat Intelligence Platform (CTIP) is a technology solution that collects, aggregates, shares and organizes threat intelligence data from multiple sources and formats, which can be consumed for prediction purpose, but which operation can be also enriched by predictive analytics. A CTIP provides security teams with information on known malware and other threats, powering efficient and accurate threat identification, investigation and response. Moreover, a TIP allows security and threat intelligence teams to easily share threat intelligence data with other stakeholders and security systems. A CTIP can be deployed either as a software-as-a-service (SaaS) or on-premise. The following presents some of the most relevant CTIPs.

The function of the CTIP is gather all the information about intelligence of threats. It provides in the project the ability the collect information of the threats that can be introduced in the system. In addition, it distinguishes a classification of the different threats, so, it is possible introduce different methods according to the type of the threat. Furthermore, this platform helps to introduce more security in the system. To do this, different tools are available depending on the characteristics that interesting in any moment.

3.2.1 Trusted Automated Exchange of Intelligence Information (TAXII)

Trusted Automated Exchange of Intelligence Information (TAXII) defines how cyber threat information can be shared via services and message exchanges. The three principal models for TAXII include:

- Hub and Spoke: one repository of information.
- Source/subscriber: one single source of information.
- Peer-to-peer: multiple groups share information.

TAXII defines four services. Users can select and implement as many as they require and combine them for different sharing models.

- Discovery: a way to learn what services an entity supports and how to interact with them
- Collection Management: a way to learn about and request subscriptions to data collections
- Inbox: a way to receive content (push messaging)
- Poll: a way to request content (pull messaging)

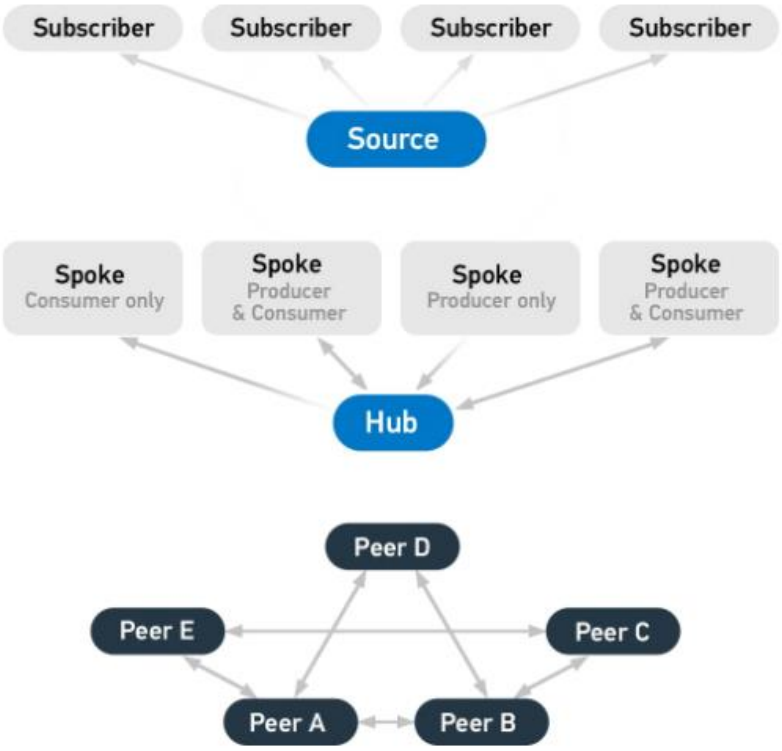

Figure 4: Three TAXII models [167].

3.2.2 Watcher

Watcher is a platform developed by the Thales Group CERT that automatically analyzes different sources to discover new potentially cybersecurity threats. It is a stand-alone solution, which can cooperate with TheHive creating cases and with MISP creating events.

Watcher can be easily deployed using docker and includes multiple sources to collect data, such as GitHub, GitLab or Twitter. More RSS sources can be added, as well as RSS-Bridges to add sources like Facebook, DuckDuckGo, GoogleSearch, etc. To do that, a list of keywords must be added to be monitored in the defined sites. It can also be used as DNS monitoring by tracking domain names and detecting suspicious emerging domain names according to a certain criterion.

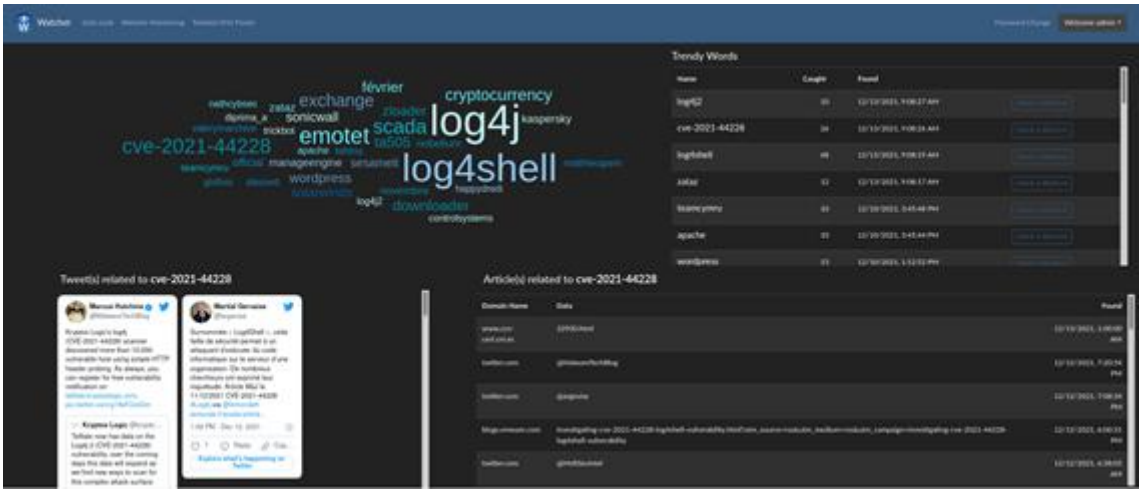

Figure 5: Watcher main view [168].

The solution provides a web interface to explore the results and those results can be exported to TheHive, MISP or as a file (multiple formats available, such as CSV, JSON or HTML).

Following there is a summary of its weaknesses and strengths as a tool:

Strengths	Weaknesses
• Easy deployment • GitHub community • Environmental data	• Limited data depth

Table 1: Strengths/Weakness Watcher

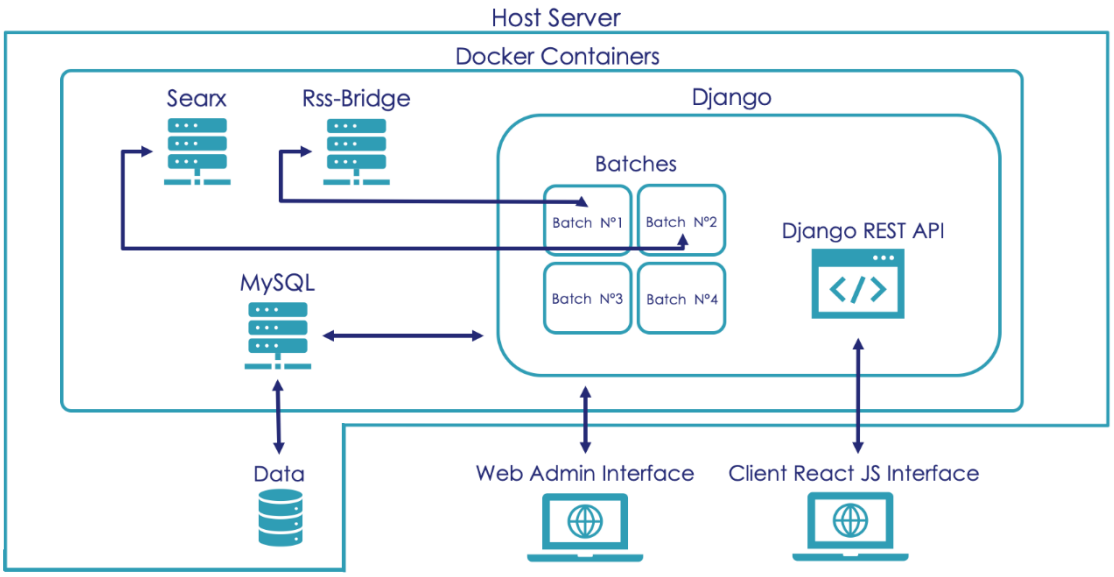

Figure 6: Watcher Platform Architecture [169].

3.2.3 SpiderFoot

SpiderFoot brings an effective tool for automate OSINT for Threat Intelligence, asset discovery and security assessments. This Open-Source Intelligence (OSINT) automation solution, support in python 3.6 and GPL-licensed, integrates with a large number of data sources and provides a wide range of methods to ingest and analyze the data provided. Nevertheless, there is a pay version that includes different extra capabilities making the free version a limited one [170].The Open-Source version can be deployed on several targets offering the user multiple ways of usage, even it can be used offensively, the entities can scan are:

IP address	Domain/sub-domain name	Hostname	Network subnet (CIDR)	ASN
E-mail address	Phone number	Username	Person's name	Bitcoin address

Table 2: Possible Entities in SpiderFoot

This tool is now running the 3.5 version and has a strong community that can access a Discord server for seeking help from the rest of users, requesting features or just general OSINT chit-chat.

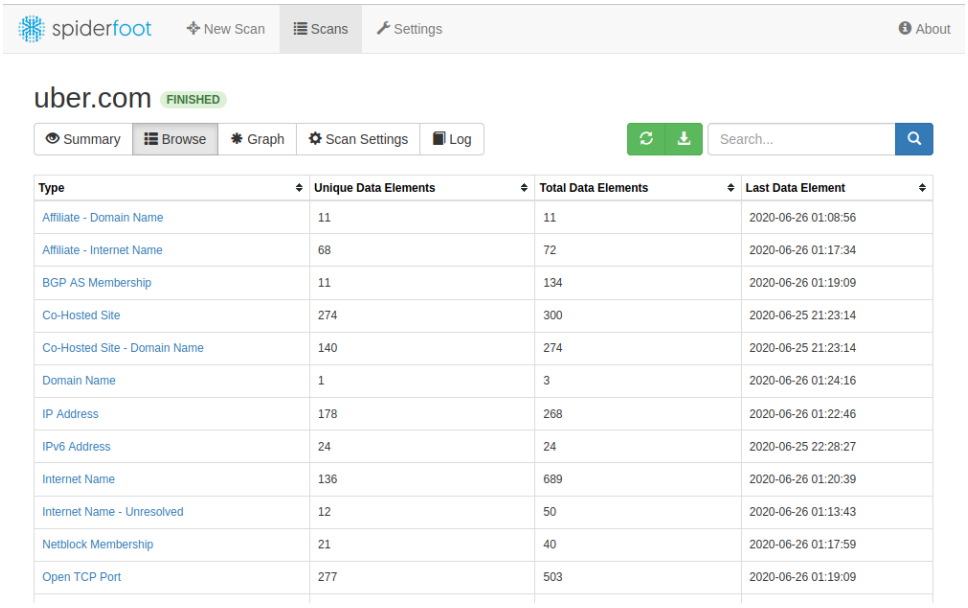

Figure 7: Spiderfoot Scans View [171].

The real improvement opportunity comes along the modules that can be deployed. In the other hand, as we have this pay version, it is logical to think that the improvements this technology can offer are going to take mostly this path.

Below is presented a table with the weaknesses and strengths found in the tool:

Strengths	Weaknesses
• Open source. • Web based UI or CLI • CSV/JSON/GEXF export • API key export/import • SQLite back-end for custom querying • Highly configurable • Fully Documented • Visualizations • Integration with others scanning tools • Developed since 2012 • Dockerization	• Low active ecosystem • Long periods between releases • Need to build from source code between releases. • Few Documentation

Table 3: Strengths/Weakness SpiderFoot

3.2.4 OpenCTI

OpenCTI is an open-source platform developed with the objective of centrally managing cyber threat information. Through its web-based dashboard it allows to collect, structure, store, organize and visualize technical and non-technical information on cyber threats. This tool can be deployed on Docker and in the cloud.

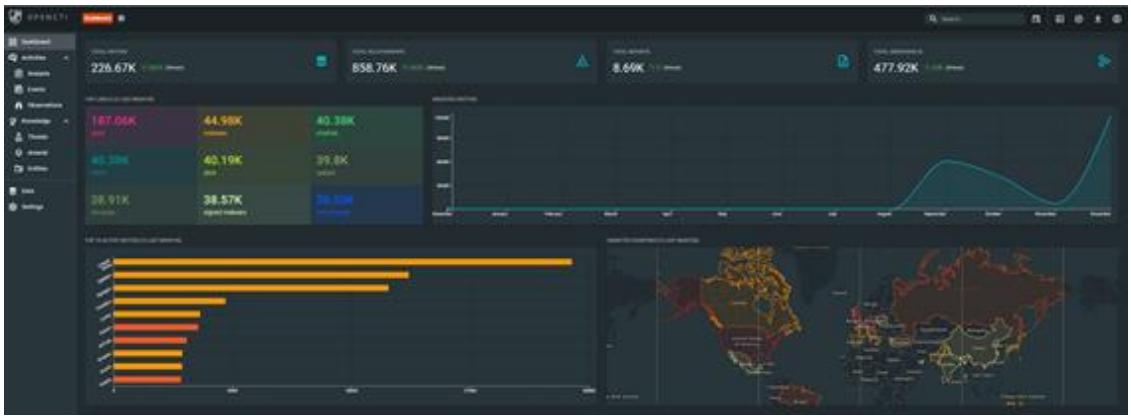

Figure 8: OpenCTI Main View [172].

For the structuring of the data, it follows a knowledge schema based on the STIX2 standard. It also has an API that allows OpenCTI to integrate with a long list of external tools of different types, whether data import (MISP, TheHive, MITRE ATT&CK, etc), enrichment (Shodan, VirusTotal, etc.), stream consumers (Elastic, Splunk, etc.) and other third-party plugins (Maltego, Cortex). All data handled by the tool can be imported or exported in multiple formats (CSV, STIX2 bundles, txt, JSON, etc.).

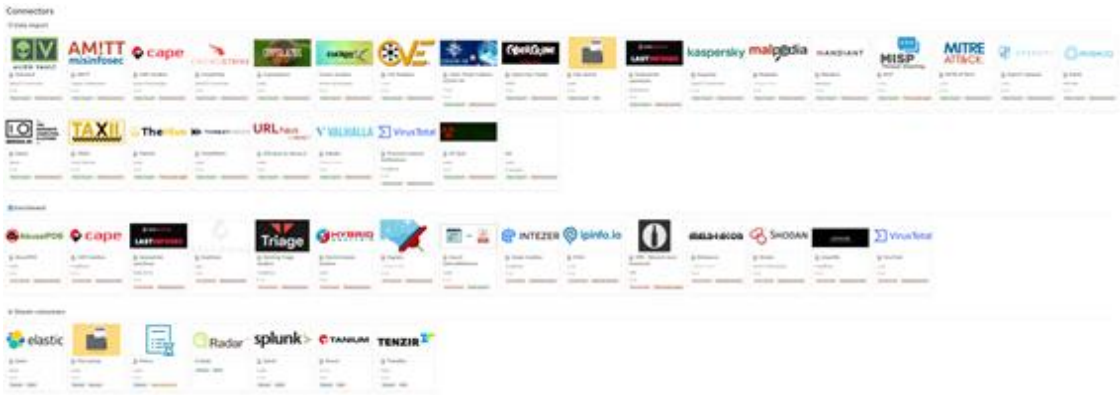

Figure 9: OpenCTI available connectors [173].

OpenCTI is a product powered by the collaboration of the French national cybersecurity agency (ANSSI), the CERT-EU and the Luatix non-profit organization. OpenCTI enjoys a high degree of maturity, a strong implantation in the sector and a more than likely important evolution in the next few years.

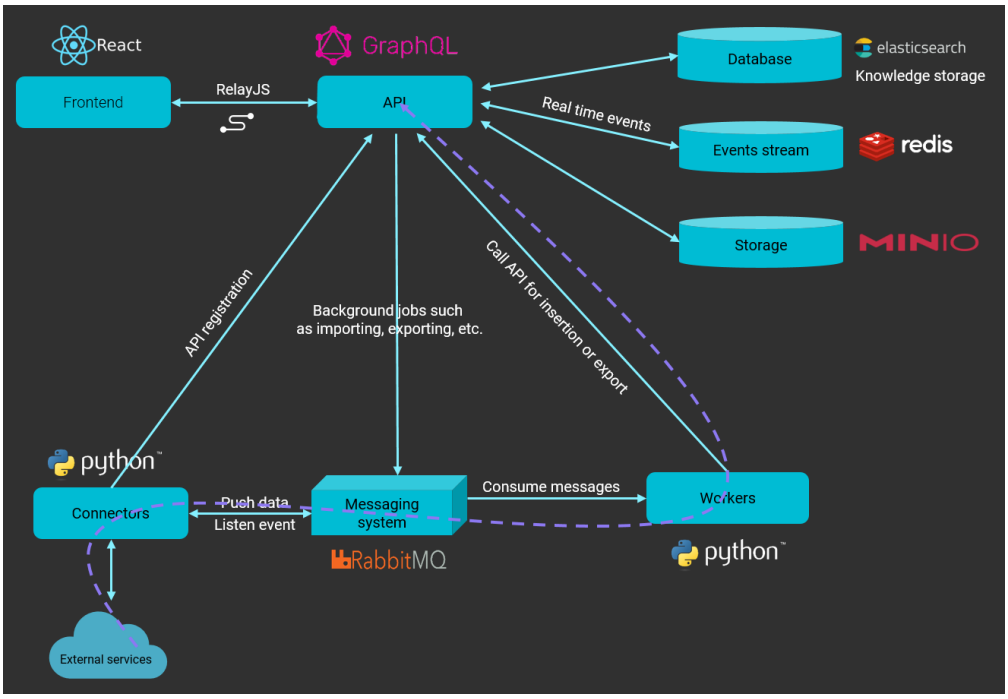

Figure 10: OpenCTI Architecture [174].

As a counterpoint, it should be noted that in order to provide OpenCTI users with cartography features, the platform uses a dedicated OpenStreetMap server. Luatix collect access log information (including IP addresses) to monitor usage and adapt services performances.

The minimal hardware requirements for all components of the platform are:

CPU	RAM	Disk type	Disk space
6 cores	16 GB	SSD	>32 GB

Table 4: OpenCTI Requirements

Within the SPARTA project context, it would be very interesting to be able to consult the numerous and complete sources of information that OpenCTI brings together, which would provide with a great deal of knowledge. Another important aspect that could be beneficial is to be able to use the STIX2 taxonomy to facilitate and promote the transfer of information between all the agents and actors involved in the project.

Below is presented a table with the weaknesses and strengths found in the tool:

Strengths	Weaknesses
• Open source. • High degree of maturity. • Strong presence in the sector. • Numerous and complete sources of information and plugins. • STIX2 standard.	• High hardware requirements. • Luatix collect access log information.

Table 5: Strengths/Weakness OpenCTI

3.2.5 Csirtg-fm

Csirtg-fm is a framework available in GitHub that offers the user a way to manage and automate the retrieval of data from third-party sources of intelligence. This tool can be configured to periodically download the data from a database, automatically parse it regardless of the format and finally process it and export it. This can be achieved by creating and adjusting different options in YAML configuration files.

The workflow that csirtg-fm follows has the next steps, that can be customized using the YAML configuration files:

1. Read input data from either files or URLs in many typical formats like JSON, XML, CSV other not so common ones that some sources may be using. Plugins can also be developed for non-http sources.
2. Csirtg-fm will attempt to parse the different fields on the format provided. A correct label may be provided on the configuration file. If not, the framework will attempt to assign the correct one.
3. Process the data and export it with a common format, independent of the original one. Output can be exported with many formats: JSON, CSV…

As well as creating an own database, it could be interesting to feed an own threat intelligence infrastructure with data from third-party sources. Csirt-fm can be useful as a transversal tool to import real-time data from these sources. The tool will make sure that all these data is up to date and normalized.

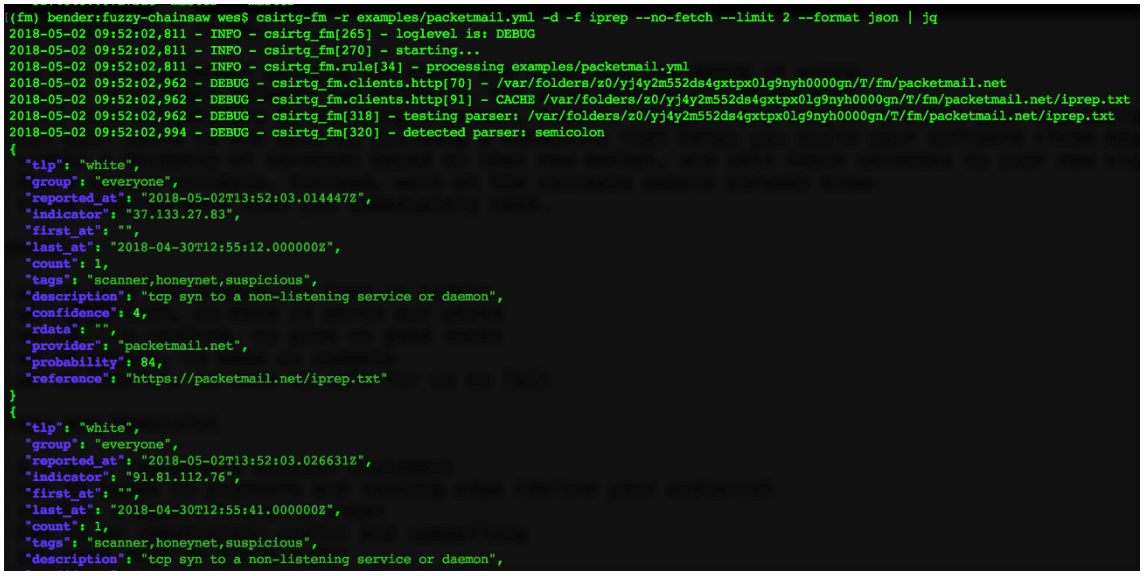

Figure 11: Csirtg-fm view [175].

As a resume, below it shows the strengths and weaknesses of the tool:

Strengths	Weaknesses
• Open source. • Normalization of data • Automatization of data retrieval • Supports uncommon formats	• Low documentation • Low maturity • Supports only text-based data • Not very useful when treating only conventional data formats

Table 6: Strengths/Weakness Csirtg-fm

3.2.6 DeepDarkCTI

DeepDarkCTI is an updated collection of CTI sources from multiple deep and dark web sources. Currently, the collection includes the following compilations:

- CVE_most_exploited: the most used CVE at the moment.
- Exploits: different exploits databases.
- Forum: different forums and mirrors
- Maas: MaaS providers.
- Markets: link to different markets from the Deep and the Dark web.
- Phishing: phishing related link, like databases, statistics or even tools like openphish.
- Ransomware_gang: different communities of different kind.
- Rat: different RAT (Remote Access Trojan) databases.
- Search_engines: different search engines for the dark/deep web.

The resources can be accessed via the proper platform (i.e. Telegram, Twitter, Discord…), common URLs or onion links. Most of the information follows the same format under *Markdown Documentation* files, so it could be used in other process or framework.

3.2.7 InfluxDB

InfluxDB is an open-source time series database developed by the company InfluxData, which integrates the essential time series toolkit:

- Dashboards
- Queries
- Tasks
- Agents

InfluxDB operates with FLUX that offers is basically the possibility to structure queries and separate common logic into functions and libraries, this makes the development an easier and faster task. But this is not the only capability FLUX has, this language can also be used to enrich your time series data with other SQL data stores (Postgres, Microsoft SQL Server, SQLite, and SAP Hana) along with cloud-based data stores (Google Bigtable, Amazon Athena, and Snowflake).

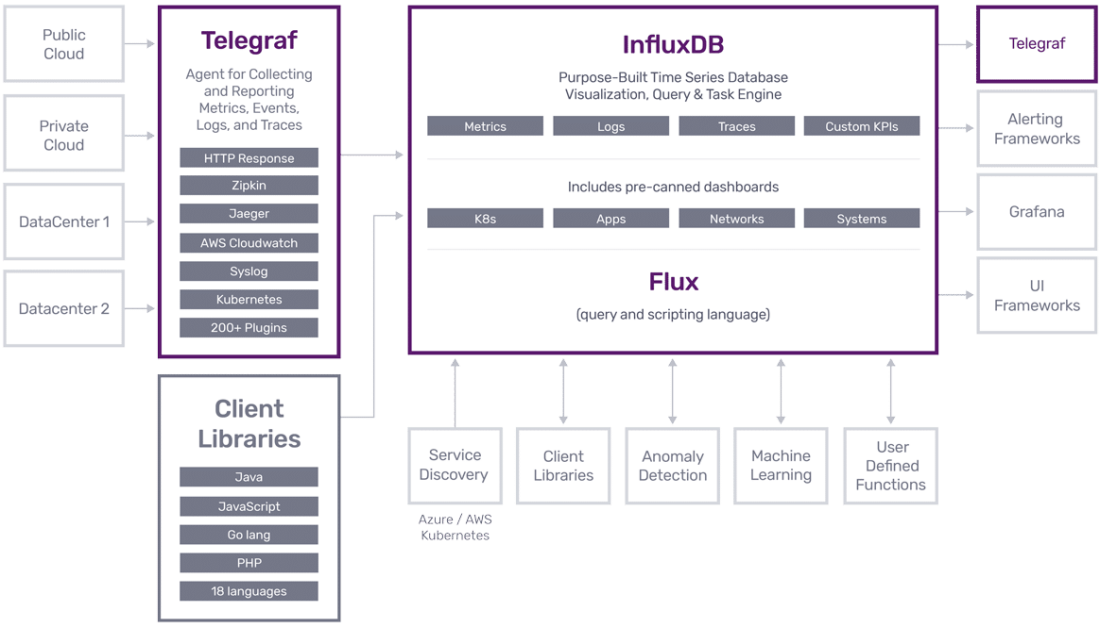

Figure 12: InfluxDB Diagram [176].

An API is provided to enhance the ingestion, query storage and visualization, this is combined with a powerful set of client libraries in Go, Java, PHP, Python and a set of InfluxDB command line tools helps developers develop in a way that is most familiar to them.

Hardware Pre-Requisites:

vCPU or CPU	RAM	IOPS	Writes per sec.	Queries per sec.	Unique series
2-4 cores	2-4 GB	500	< 5000	< 5	< 100000
4-6 cores	8-32 GB	500 - 1000	< 250000	< 25	< 1000000
8+ cores	32+ GB	1000+	> 250000	> 25	> 1000000

Table 7: InfluxDB Requirements

Below it shows the strengths and weaknesses of the tool:

Strengths	Weaknesses
• Custom high performance datastore written specifically for time series data. • Written entirely in Go. • Simple, high performing write and query HTTP APIs. • Plugins support for other data ingestion protocols such as Graphite, collectd, and OpenTSDB.	• Dependencies on TICK components. • 1.6+ no longer supports 32-bit. • High disk usage. • Out-of-memory loops.

Strengths	Weaknesses
• Expressive SQL-like query language tailored to easily query aggregated data. • Continuous queries automatically compute aggregate data to make frequent queries more efficient.	

Table 8: Strengths/Weakness InfluxDB

Kapacitor

Kapacitor provides a Real-time streaming data processing engine. The input that Kapacitor can process it can be both stream and batch data from InfluxDB, due to its programming language TICKscript, this tool manages data in real-time, deploying the information as both a pre-processor to downsample and perform advanced analytics across the infrastructure before shipping it to InfluxDB.

The alerting follows the model publish-subscribe and topics are connection between both parts, alerts (publish) and handler (subscriber), this model and the ability for these to call User Defined Functions ensure Kapacitor very flexible to act as the control plane in your environment, performing tasks like auto-scaling, stock reordering, and IoT device control. In addition, Kapacitor can be integrated with machine learning, pattern matching and rules engines.

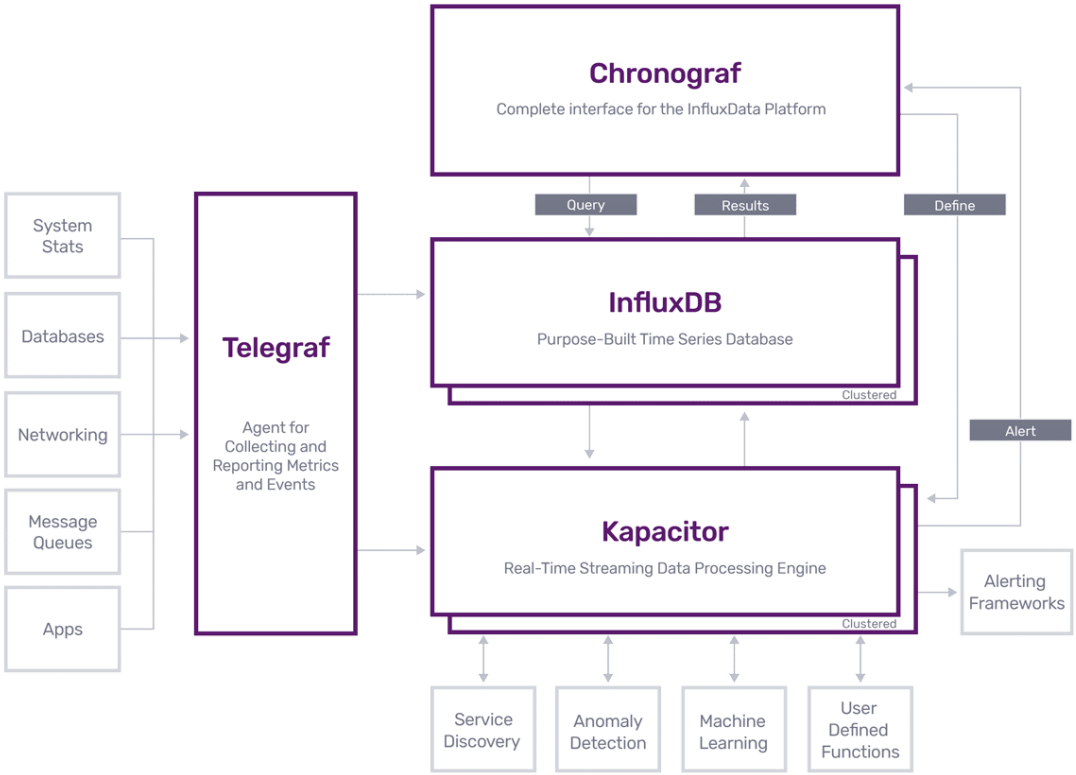

Figure 13: Kapacitor Integration with InfluxDB [177].

A bad aspect 1.6+ no longer supports 32-bit operating systems. If it is being used on a 32-bit operating system, is better to keep using Kapacitor 1.5.x

Builds have been provided for ARM 64-bit architectures, and these are the strengths and weaknesses.

Strengths	Weaknesses
• Action-Oriented. • Streaming analytics. • API or CLI. • Dockerization. • Process both streaming data and batch data. • Add custom user defined functions to detect anomalies. • Highly integrational.	• Dependencies on TICK components • 1.6+ no longer supports 32-bit. • High disk usage. • Out-of-memory loops.

Table 9: Strengths/Weakness Kapacitor

Telegraf

Telegraf is a plugin-driven server agent, which is able to collect and send metrics and events from IoT databases, systems and sensors. This tool is developed in Go and compiles into a single binary.

Telegraf can be used to collect and send all kinds of data; either from databases such as MongoDB, MySQL among others or from systems collecting metrics from containers or orchestrators among others or even from IoT sensors such as critical status data from sensors (pressure level or temperature) and IoT devices. In addition, it is also possible to analyze the input data formats of the metrics. These include InfluxDB Line Protocol, JSON or Nagios among others.

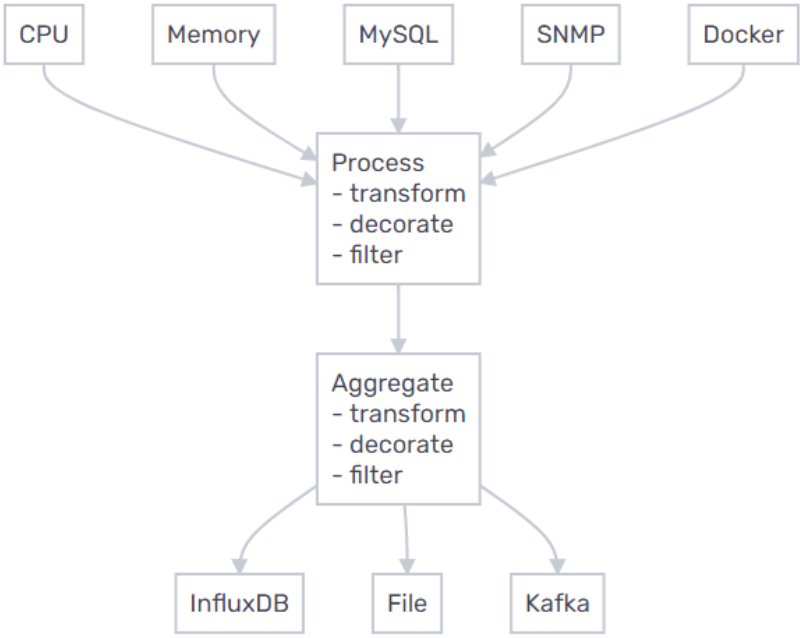

Figure 14: Telegraf aggregator plugins [178].

- Processor plugins process metrics as they pass through and immediately emit results based on the values they process.
- Aggregator plugins, on the other hand, are a bit more complicated. Aggregators are typically for emitting new aggregate metrics, such as a running mean, minimum, maximum, quantiles, or standard deviation. For this reason, all aggregator plugins are configured with a period.

Finally, Telegraph shows a great integration since metrics can be obtained from third party APIs or through Kafka for example. On the other hand, the output metrics can be sent to other services such as MQTT or NSQ among others.

These are the strengths and weaknesses of the tool:

Strengths	Weaknesses
• Open source • GitHub community • Flexibility • High integration • Wide coverage • Written entirely in Go	• Few Documentation

Table 10: Strengths/Weakness Kapacitor

Chronograf

Chronograf is the user interface and administrative component of the InfluxDB 1.x platform, which enables to quickly store data in InfluxDB to build queries and robust alerts. It allows you to build dashboards with real-time visualizations of the data.

This application is the administrative tool for the InfluxDB implementation. It also provides a number of security options such as user authentication services (GitHub, Google, Auth0 and others) and role-based access controls to help administrators provision the right resources (dashboards, InfluxDB and Kapacitor connections) to ensure security and compliance postures.

The tool is also the user interface of Kapacitor, a native data processing engine that allows processing both streaming and batching data from InfluxDB.

Below are different screenshots of the application.

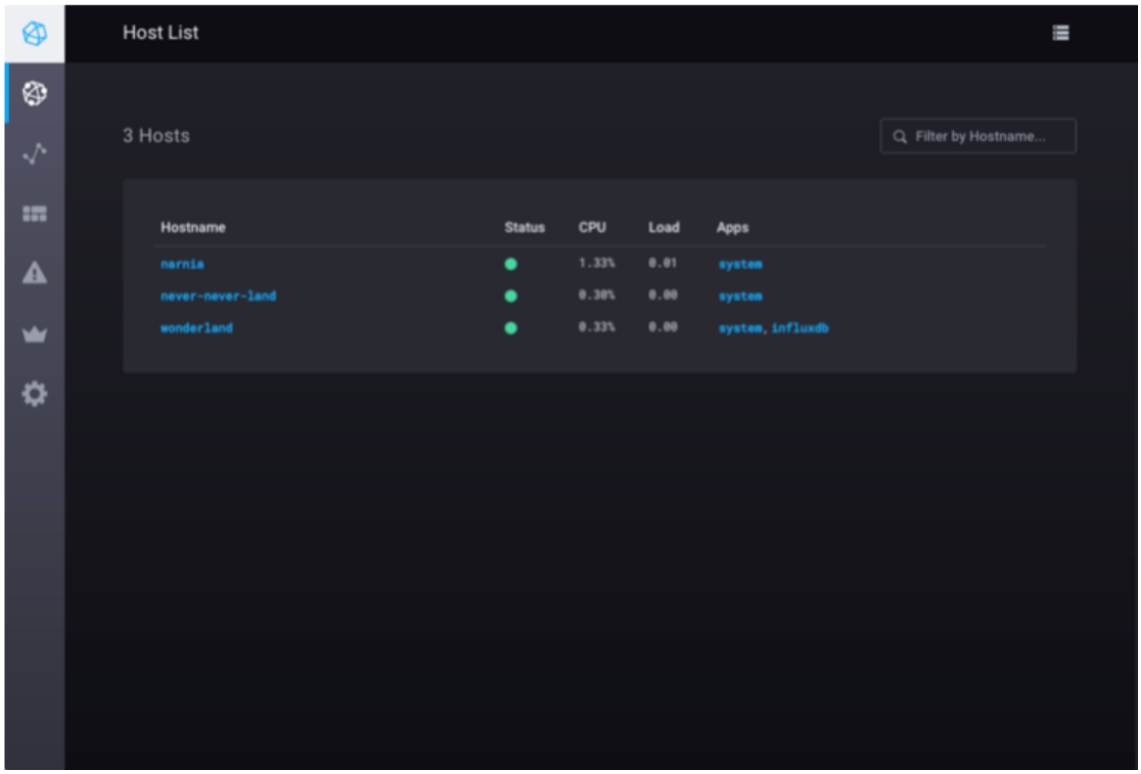

Figure 15: Chronograf – Infrastructure Monitoring [179].

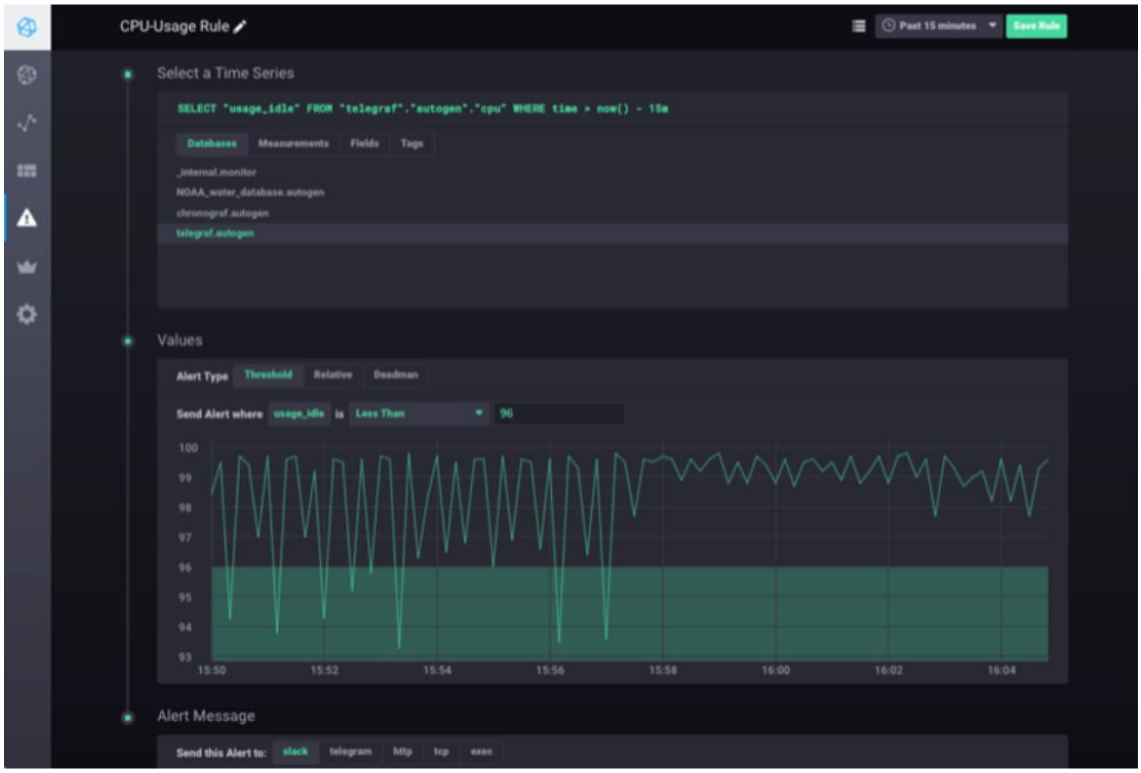

Figure 16: Chronograf – Alert Management [180].

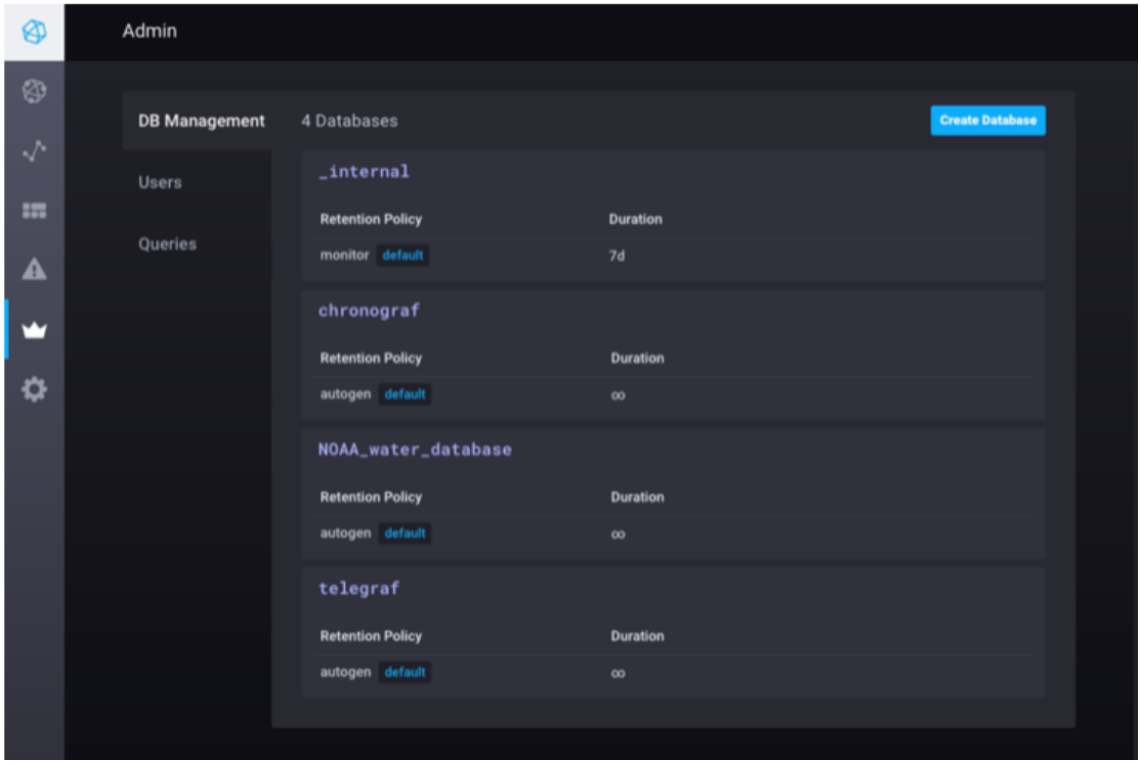

Figure 17: Chronograf – Database Management [181].

Figure 18: Chronograf – Data Visualization [182].

These are the strengths and the weaknesses of the tool:

Strengths	Weaknesses
• Open source. • Dashboard templates. • Simple user interface for the creation of alerts. • Provides user authentication services and role-based access controls.	• Few Documentations. • Dependencies on TICK components.

Table 11: Strengths/Weakness Chronograf

3.3 Security Information and Event Management (SIEM)

The Security Information and Event Management (SIEM) tools entail a family of solutions that provides organizations with the capabilities to threat detection, analysis and response; thus enabling an adequate ecosystem for developing and operating prediction related functionalities and capabilities. The following illustrates some widely adopted SIEM products with the potential of assist predictive analysis.

SIEMs work by collecting and reading the logs generated by the different applications, devices and systems. From this data the SIEM can then analyze it and search for different possible threats, such as failed authentications, malware, denial of service attempts, phishing among others. While this data is used to detect threats currently happening it could be used to learn the most common patterns of those threats, helping us predict them before they happen.

3.3.1 XL-SIEM

XL-SIEM is a robust SIEM solution developed by Atos. The main strength of XL-SIEM is the capability to integrate events from different detection systems, such as Honeypots, antivirus or even other Open-source tools. It is a very scalable solution. As XL-SIEM centralizes information coming from multiple sources, it is able to collect information from different layers, such as the network or the application layer, depending on the source.

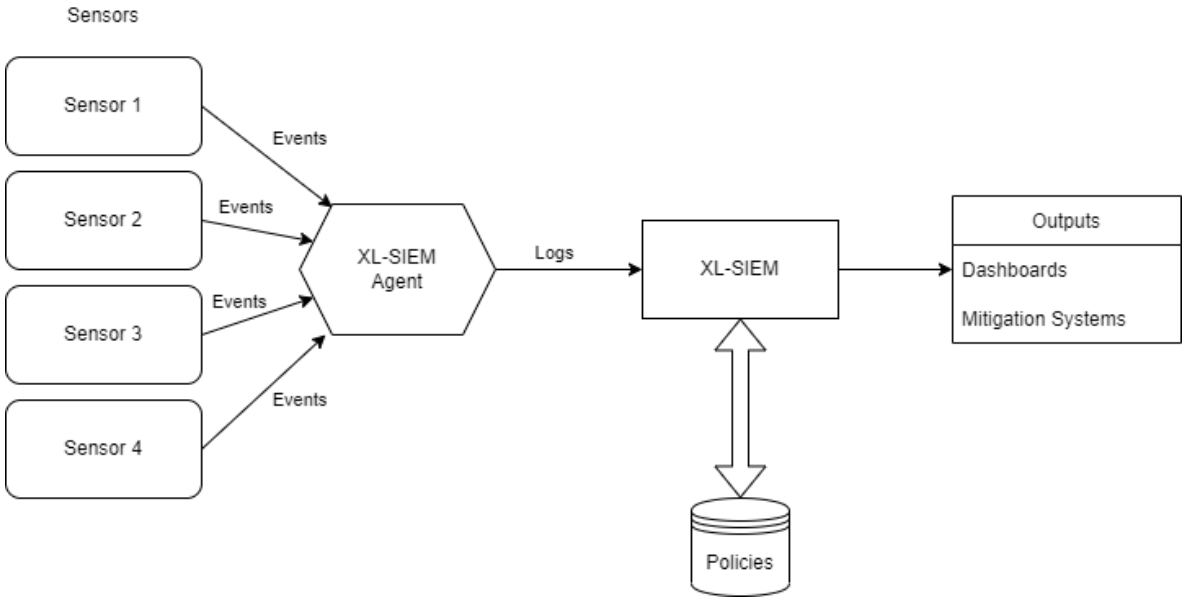

Figure 19: XL-SIEM Flow

As it is shown in the diagram above, the agents will gather and digest the events coming from the sources and will provide the XL-SIEM server with the standardized logs. Once the logs are analyzed, the tool can perform different actions according to the criteria and policies defined. The scalability of the solution improves the performance, allowing to process big amounts of data with complex rules which correlate different events. The XL-SIEM alert system includes plenty of alerts following different types of threats, such as Dos, DDoS, Scannings, Brute Force, rootkits, etc.

The Table below summarizes the list with the strengths and weaknesses identified by the SPARTAN stakeholder of the tool, which were discussed and agreed assuming its application for assisting threat prediction.

Strengths	Weaknesses
• Centralization of the information coming from different systems • Correlation between events • Threat identification • Data normalization	• Needs large amounts of data • Might impact the internal performance

Table 12: Strengths/Weakness XL SIEM Flow

3.3.2 ELK Stack

Also known as ELK or Elastic Stack, is an open-source solution that consists of multiple free SIEM products. The wide variety of SIEM products, such as the Logstash components, make possible for ELK to aggregate logs from almost any data source, and correlate the data using different plugins, although they require manual configurations. The hardware requirements for ELK are the following:

CPU	RAM	Disk type	Disk space
No specific requirement, the more the better.	Desired 64Gb Not less than 8Gb	SSD or >15k RPM spinning media	The more the better

Table 13: ELK Stack Requirements

The basic ELK stack configuration consists of Logstash (log processing), ElasticSearch (data storage, can be clustered) and Kibana to visualize the dashboards and results.

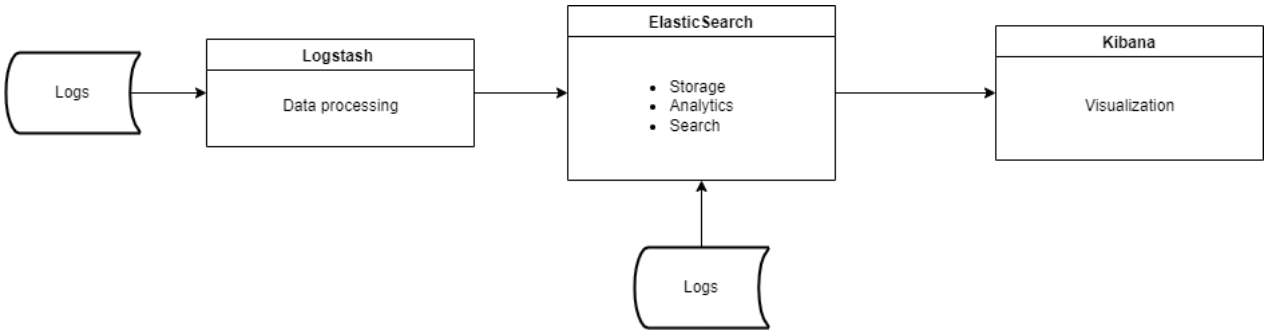

Figure 20: ELK Flow Own graphic

The detection efficacy has been improved by reducing the alert fatigue with high-fidelity alerting, Machine learning-based anomaly detection and MITRE ATT&CK mapping. As the SIEM solution consists of multiple products which can be deployed or not, the solution is scalable to different deployments.

ELK can be easily deployed using docker, and it is possible to deploy multiple containers or to deploy a stacked solution that can include multiple SIEM products. The Table below summarizes the list with the strengths and weaknesses identified by the SPARTAN stakeholder of the tool, which were discussed and agreed assuming its application for assisting threat prediction.

Strengths	Weaknesses
• Correlation plugins. • Adaptability to the architecture. • Open-Source solution. • Multiple components available. • Distributed storage. • Docker deployment available.	• Manual configuration. • Complex data normalization.

Table 14: Strengths/Weakness ELK Stack

3.3.3 Splunk

Splunk is a commercial SIEM product and as such, it acts as central node to gather and analyze the data coming from different sources of intelligence. With this data, the user will be able to create real-time alerts and reports to monitor the infrastructures and detect security incidents. There are different options of Splunk, the paid versions or free version. Some of the functionality offered by Splunk include:

All the data that arrives to Splunk is index according to the source, date and other factors. The user can then specify conditions so that every time that historic or real time data match them, an alert is triggered. Alerts can be configured to have different effects, like send an email, post it on an RSS feed or even run a custom script. Reports can be also created and programmed to run periodically and trigger alerts if they meet particular conditions. Since every deployment is particularized according to the company or project requirements, multiple samples validated architectures were defined by Splunk [183], which can be modified according to the needs. The following example represents a distributed clustered deployment:

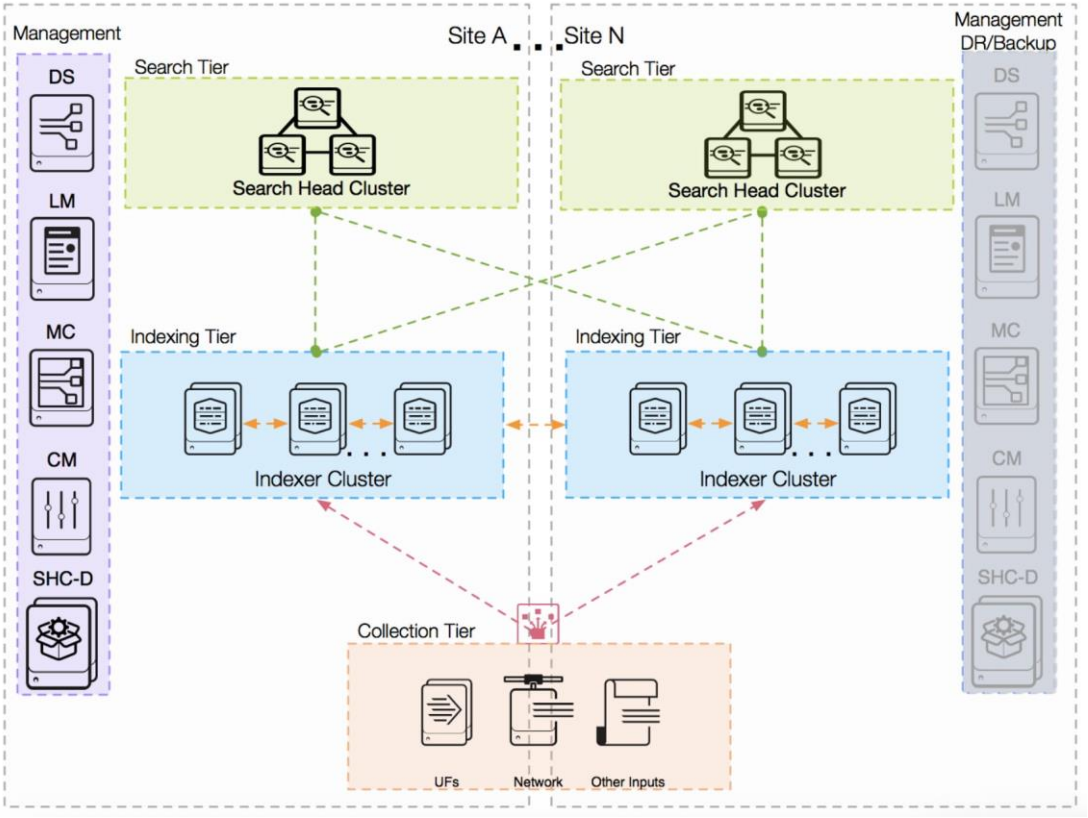

Figure 21: Splunk Architecture [184]

The Table below summarizes the list with the strengths and weaknesses identified by the SPARTAN stakeholder of the tool, which were discussed and agreed assuming its application for assisting threat prediction.

Strengths	Weaknesses
• Real-time analysis • Powerful • Data normalization • Data analytics • Distributed Searches • Live Dashboards	• Not open source • Complex installation and maintenance • High learning curve

Table 15: Strengths/Weakness Splunk

Moreover, these are the minimum requirements for an Enterprise deployment:

- 64-bit processor
- 8 cores or 16 vCPUs
- 12 GB RAM

3.3.4 Ossec

OSSEC is an open-source monitoring solution that mixes HIDS (host-based intrusion detection), log monitoring, and Security Incident Management (SIM)/Security Information and Event Management (SIEM) together. This multiplatform solution is an open-source analyzer and monitoring tool that mixes several types of technologies to conform a solid block that is capable of digest large number of logs due to processes such as **logcollector** or **analysisd**, the former is able to collect the events

and the latter analyses, decodes, filters and classifies the events identified by **logcollector**. OSSEC helps customers meet specific compliance requirements, letting customers detect and alert on unauthorized file system modifications and malicious behavior embedded in the log files of commercial products as well as custom applications. The way OSSEC lets customers configure incidents they want to be alerted on and lets them focus on raising the priority of critical incidents over the regular noise on any system. Integration with smtp, sms, and syslog allows customers to be on top of alerts by sending them to e-mail enabled devices. Active response options to block an attack immediately are also available.

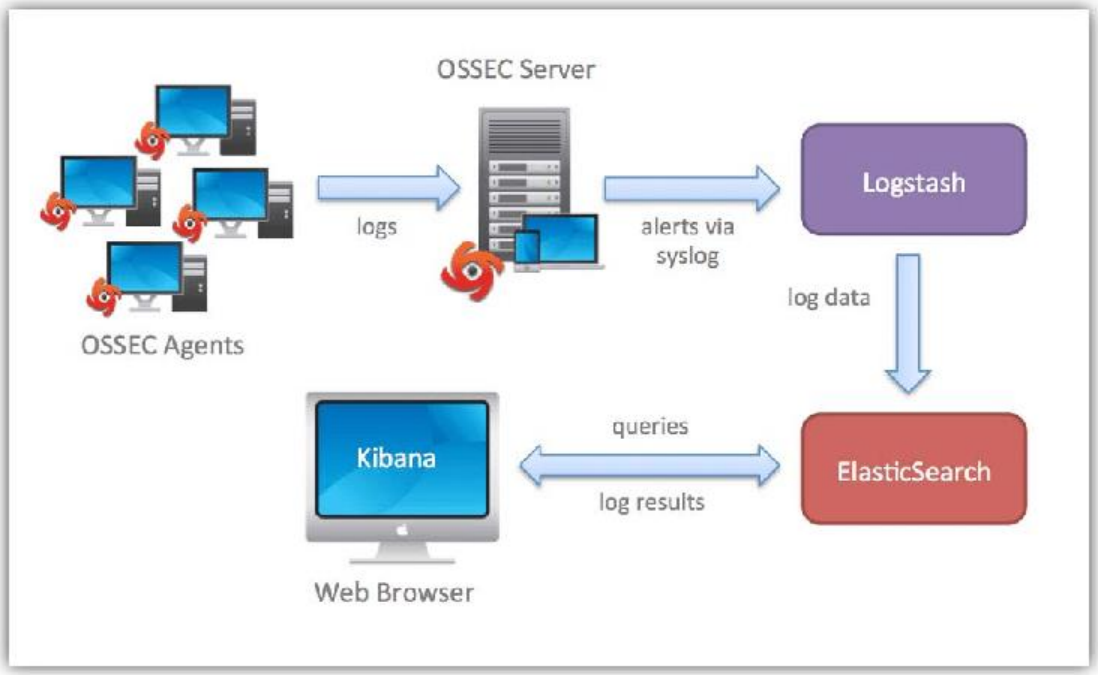

Figure 22: Ossec Architecture [185]

These are the pre-requisites for the tool: For UNIX systems, OSSEC only requires gnu make, gcc, and libc. OpenSSL is a suggested, but optional, prerequisite.

The Table below summarizes the list with the strengths and weaknesses identified by the SPARTAN stakeholder of the tool, which were discussed and agreed assuming its application for assisting threat prediction.

Strengths	Weaknesses
• Open source. • File Integrity checking. • Log Monitoring. • Rootkit detection. • Active response. • Centralized management. • Agent and agentless monitoring. • Integration with current infrastructure. • Compliance Requirements. • Multi-platform.	• Upgrade process overwrites existing rules. • Pre-sharing keys can be problematic. • Windows supported in server-agent mode only. • OSSEC remote protocol uses weak cryptography.

Table 16: Strengths/Weakness Ossec

3.3.5 AlienVault OSSIM

OSSIM is an open-source SIEM developed by AlienVault. It combines many different open-source projects (Snort, Suricata, Munin, Nagios…) into a single package, providing a complete view of all the security-related aspects of an information system. OSSIM achieves this through a web interface through which administrators will be able to see the information gathered by sensor on the devices of the network, and tools to analyze, interpret and display them. Some of the functionality provided by OSSIM include the following:

- Event collection, normalization and correlation
- Asset discovery
- Vulnerability assessment
- Intrusion detection
- Behavioral monitoring
- Integration with OTX

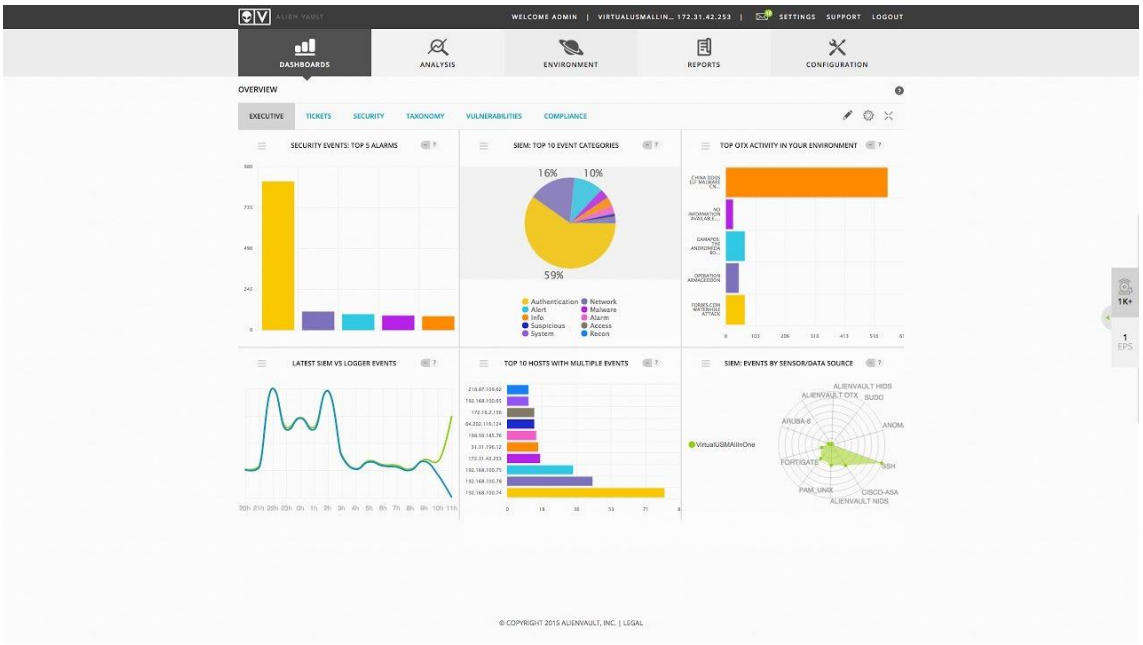

Figure 23: AlienVault OSSIM [186]

The Table below summarizes the list with the strengths and weaknesses identified by the SPARTAN stakeholder of the tool, which were discussed and agreed assuming its application for assisting threat prediction.

Strengths	Weaknesses
• Open source. • High maturity. • Vulnerability assessment. • Simple installation. • Custom tools. • Clean interface. • Easy to access via browser.	• AI not implemented. • Lacks log management. • Not very powerful reporting.

Table 17: Strengths/Weakness AlienVault OSSIM

Minimum requirements (<1000 events per second):

- 2 CPU cores
- 4-8 GB RAM
- 50GB HDD

3.3.6 Metadon

This SIEM solution integrates Security Information Management, SEM, taxonomy for security events and incidents, and an information model leveraged on the industry and government published standards. Metadon is capable of ingesting multiple forms of data via connectors, such as syslog, TCP, HTTP or file based. A domain specific language allows the transformation of input formats, also allows the normalization into common events and field taxonomies. Currently leverages Elastic Search as a back-end for data storage and provides a web interface where the security analytics can query the data. This software implements a query interface in order to communicate other components or users in case abnormal operations are being performed. To complement data discovery, visual analytics can be used to visually display attributes that help operators detect attacks and their causes.

The Table below summarizes the list with the strengths and weaknesses identified by the SPARTAN stakeholder of the tool, which were discussed and agreed assuming its application for assisting threat prediction.

Strengths	Weaknesses
• Easily customized. • Very flexible design. • Easy deployment without single points of failure. • Immediate value. • Dynamic data correlation and has real-time data streaming capabilities.	• Dependency on ElasticSearch. • Important features not available, like LDAP authentication. • An operator from the company is needed to setup and configure. • No documentation.

Table 18: Strengths/Weakness Metadon

3.4 Security Orchestration Automation and Response (SOAR)

Security Orchestration, Automation and Response (SOAR) platforms are a collection of security software solutions and tools for browsing, collecting data from a variety of sources, automation and management security operations. SOAR solutions then use a combination of human and machine learning to analyze this diverse data in order to comprehend and prioritize incident response actions. Since SOAR may facilitate bot reactive but also proactive response, to embrace predictive analysis entails an appreciated advantage when dealing with fast triggering of countermeasures (e.g. anticipatory actions), or understanding the next steps of the attacker

SOAR are tools that help in the coordination and execution of jobs between various people and tools from a single platform, simplifying the whole process of cyberthreats mitigation. This kind of tools can help us automate the processing of the ingested data through machine learning to learn from current threats and apply that knowledge to predict future ones.

3.4.1 PatroOwl

PatrOwl is an open-source application for managing security assets, released under the Affero General Public License, and it is also available as different commercial pro-editions with some extra features (e.g., professional support). In order to develop different scanning, PatrOwl provides multiple engines, which can be used to analyze bearings and known vulnerabilities of the assets, such as NMAP, Cortex or Burp.

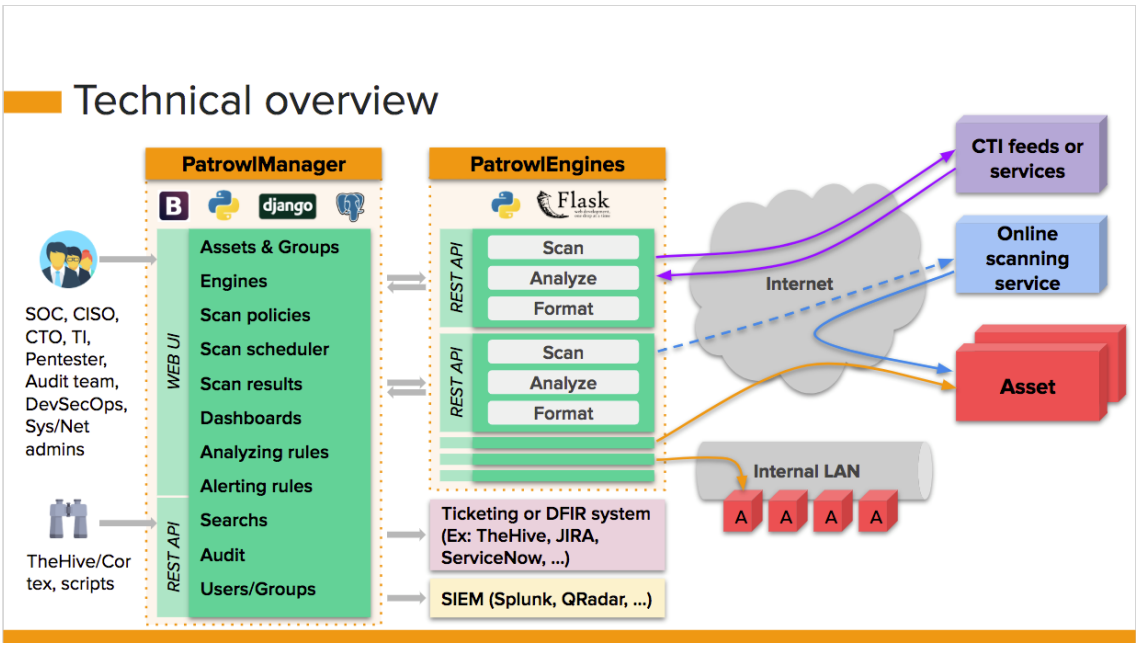

Figure 24: PatrOwl Technical Overview [187]

The deployment consists of a PatrowlManager and some PatrowlEngines and can be done using Docker. PatrowlManager centralizes the information provided by the PatrowlEngines, orchestrates the management and provides a web interface that provides reports and dashboards, but the data can also be consumed via JSON-API.

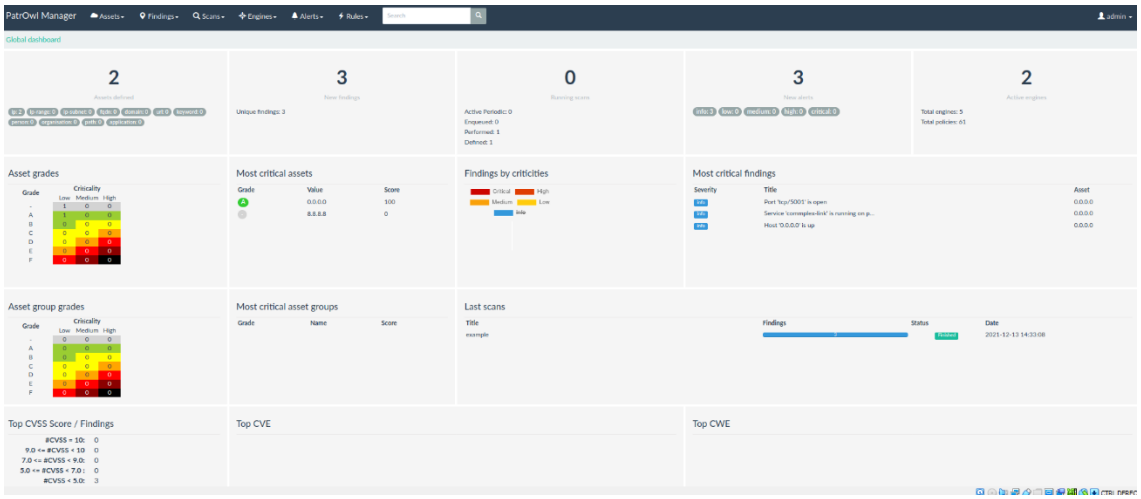

Figure 25: PatrOwl main view

The scanning can be performed manually or periodically, which makes PatrOwl an interesting application for pentesting automation.

The Table below summarizes the list with the strengths and weaknesses identified by the SPARTAN stakeholder of the tool, which were discussed and agreed assuming its application for assisting threat prediction.

Strengths	Weaknesses
• Open Source solution. • Multidisciplinary Engines. • Results normalization. • Scan automation. • Official support and GitHub community. • Pro-edition and SaaS availability. • API availability.	• Engines configuration might be tedious • Limited data.

Table 19: Strengths/Weakness PatrOwl

Although PatrOwl provides plenty of analysis, the information it provides focuses on known vulnerabilities and the status of the assets, instead of analyzing the current environment that could provide useful information to be analyzed in order to predict future events.

3.4.2 TheHive Project

The objective of TheHive Project is to help analysts to deal with security alerts. Developed entirely (with the exception of MISP) by a group of experienced professionals in the field of cybersecurity. TheHive project integrates three open-source tools:

- **TheHive**
- **MISP**
- **Cortex**

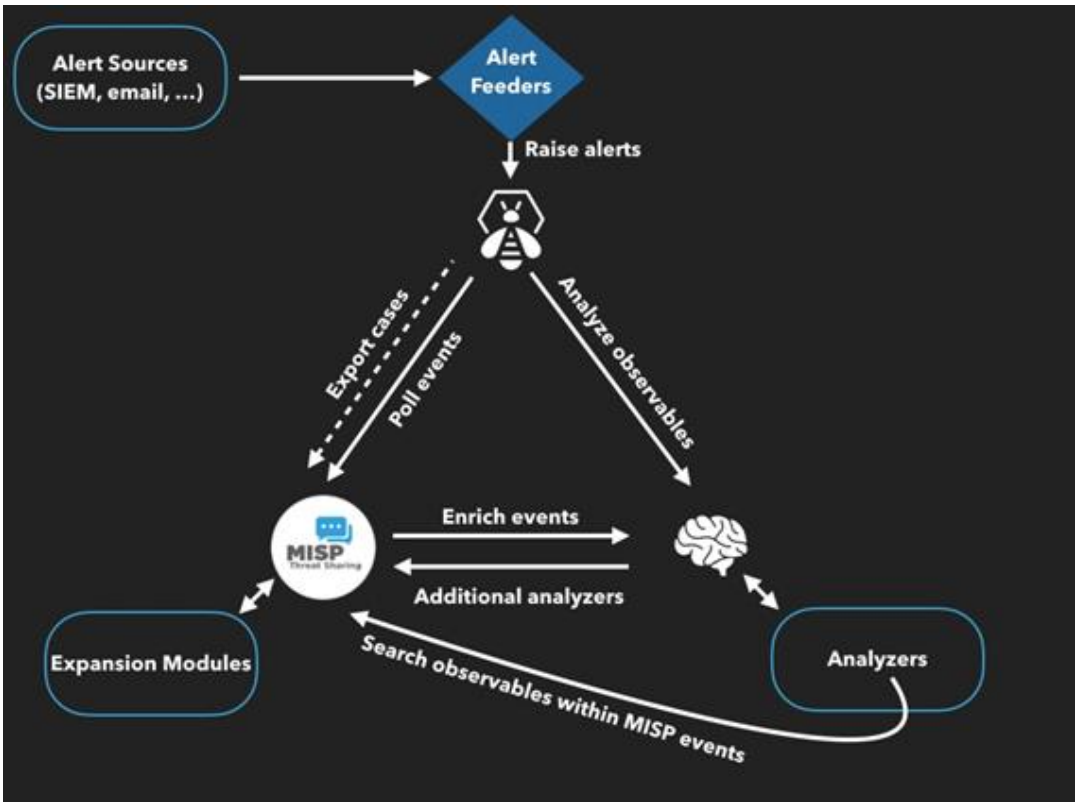

Figure 26: TheHive Project Workflow [188]

The installation and deployment process of the tool is somewhat complicated, since it does not officially allow the use of Docker or other orchestrators and must be performed manually. Its modular architecture forces the user to make important decisions before the installation process, such as selecting the database, the file storage system and the index system.

TheHive

TheHive is an open-source Security Incident Response Platform (SIRP) fully integrated with MISP and Cortex to receive, analyze and share information related to different security events. By transmitting data in real time and receiving alerts from different sources (CTI Providers, SIEM, IDS, email, etc.) through its REST API (TheHive4py), TheHive enables collaborative cross-organizational management of security incident response.

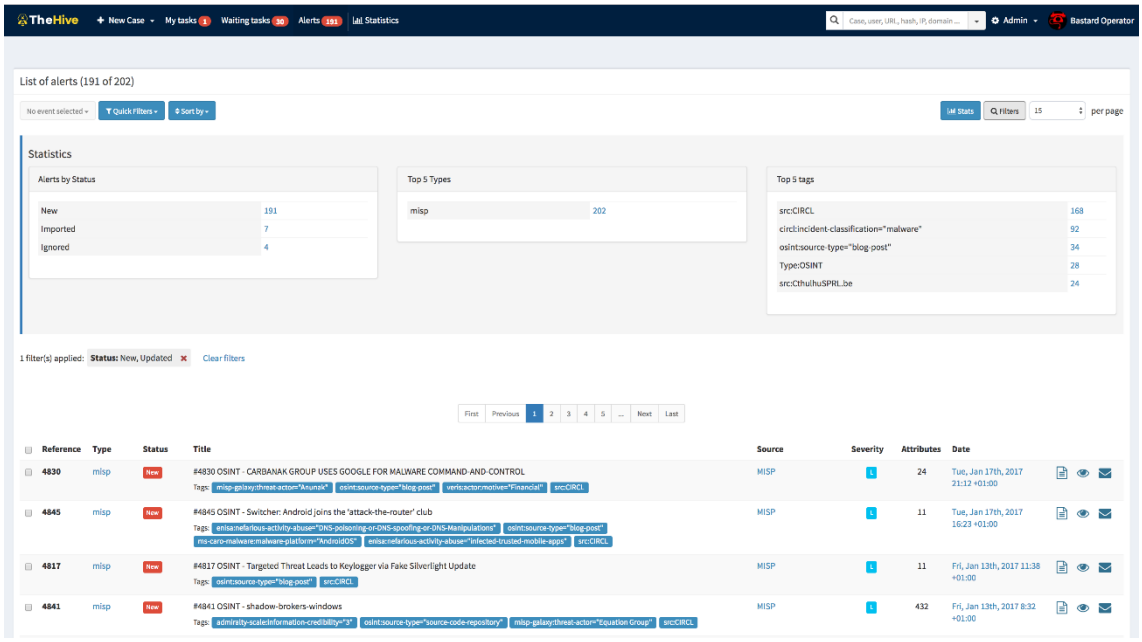

Figure 27: TheHive Alert Panel [189]

Includes a template-based case creation engine, analysis, response and tasking automation mechanisms. It also allows filtering and searching of alerts, evidence logging and TLP tag association. All these capabilities facilitate the work of security investigators and make it possible to easily identify the activities in which most effort is invested.

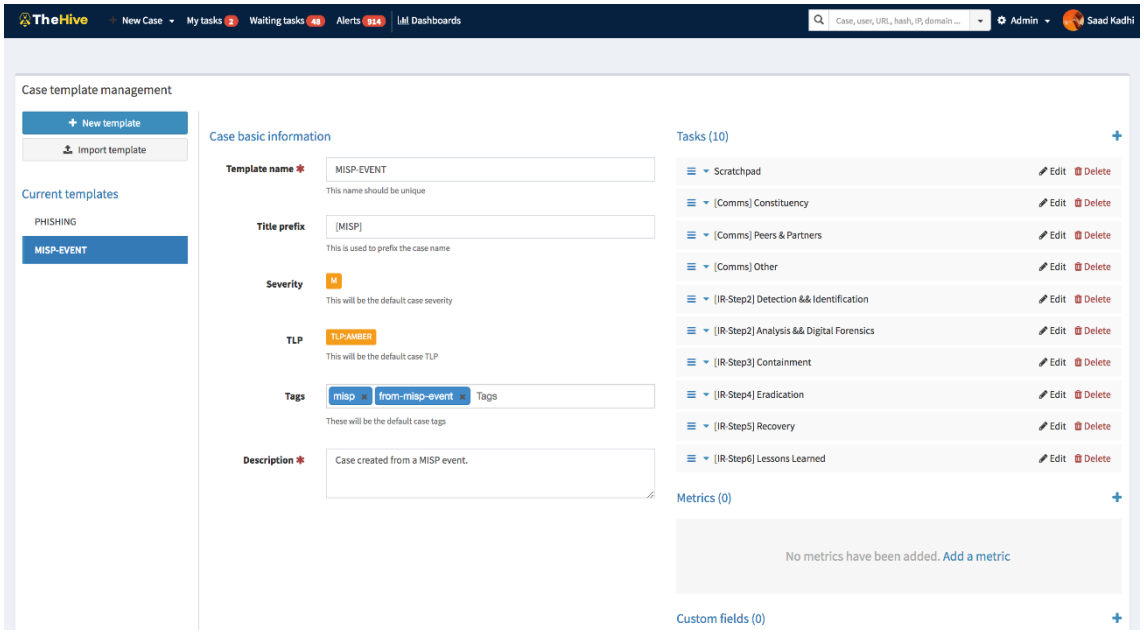

Figure 28: TheHive Case Template Management [190]

In addition, TheHive has native integration with a long list of third-party applications, such as Watcher, DigitalShadows, Synapse, StackStorm and Zerofox among others.

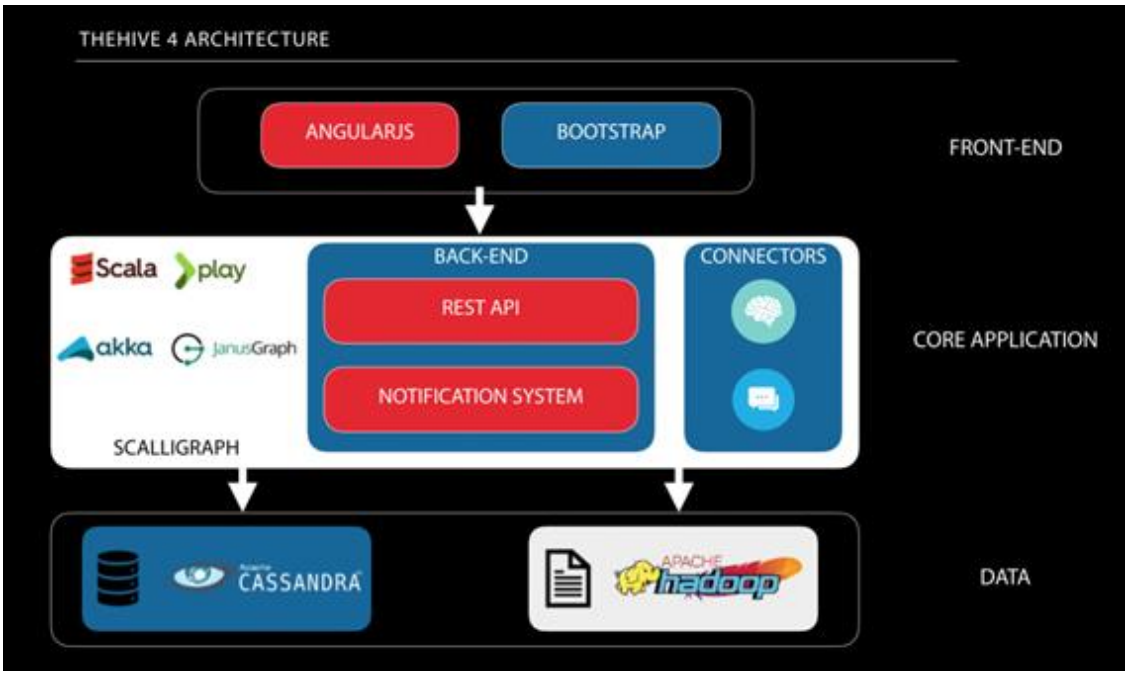

Figure 29: TheHive 4 New Architecture [191]

MISP

Mainly, MISP (Malware Information Sharing Platform) is a threat intelligence platform created to share, store and correlate Indicators of Compromise (IoCs) of targeted attacks, thus allowing different organizations to share information about Malware and its indicators. It has been developed by CIRCL (Computer Incident Response Center Luxembourg), the Belgian Defense team and NATO (NCIRC).

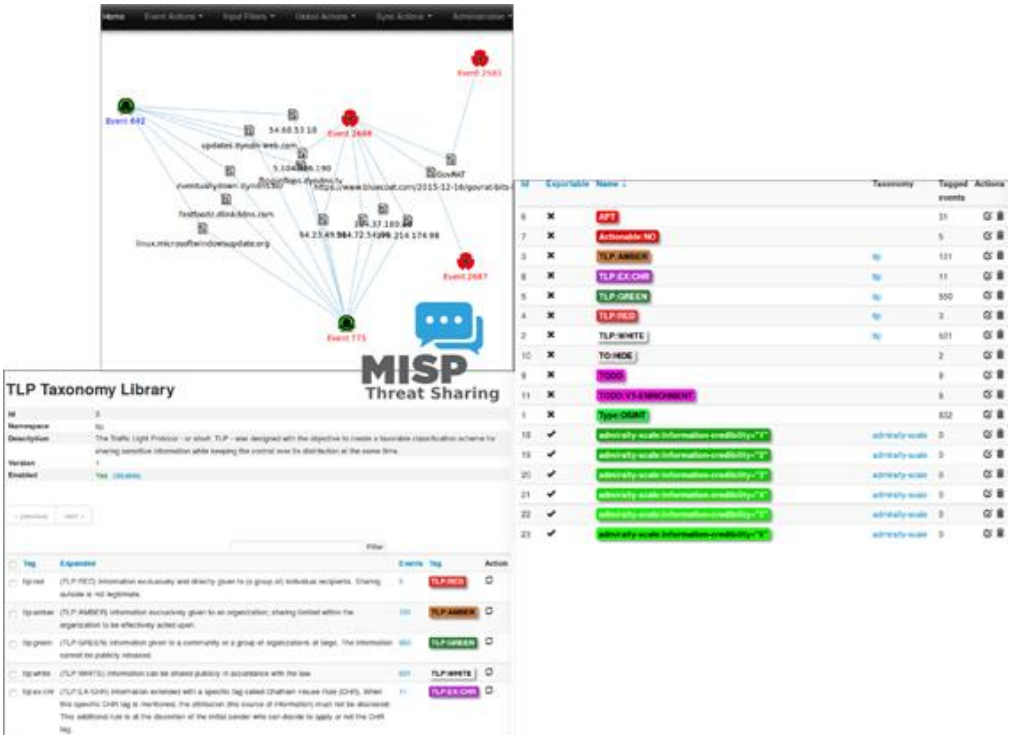

Figure 30: MISP views [192]

The objective of MISP is to foster the sharing of structured information within the security community and abroad in real time. MISP provides functionalities to support the exchange of information but also the consumption of said information by Network Intrusion Detection Systems (NIDS), LIDS but also log analysis tools, SIEMs. In addition to an extensive and efficient database, it includes a correlation engine between attributes and indicators, flexible data models and APIs. It supports STIX taxonomy and intelligence vocabularies called MISP galaxy, which can be grouped and linked with existing threat actors, malware, RAT, ransomware or MITRE ATT&CK. Consequently, all data can be exported and imported in multiple formats, such as CSV, JSON, MISP XML, among others.

One of the greatest strengths that MISP boasts is that numerous external open source and proprietary tools are fully integrated thanks to the MISP API. The additional software supported by the MISP project allows the community to have additional tools to support their daily operations and greatly expand their functionality. Relevant third-party tools include Elastic.co, Hybrid analysis, Splunk, TheHive, OpenCTI and Kaspersky Threat Feeds [193].

Cortex

Cortex is an independent analysis engine, open source and with a web interface, which aims to analyze, one by one or in bulk mode, different observables such as IP and email addresses, URLs, domain names, files or hashes. It can make use external analyzers like VirusTotal, PassiveTotal or DomainTools that can provide different scanning services. Thanks to its REST API, all of the above operations can be automated.

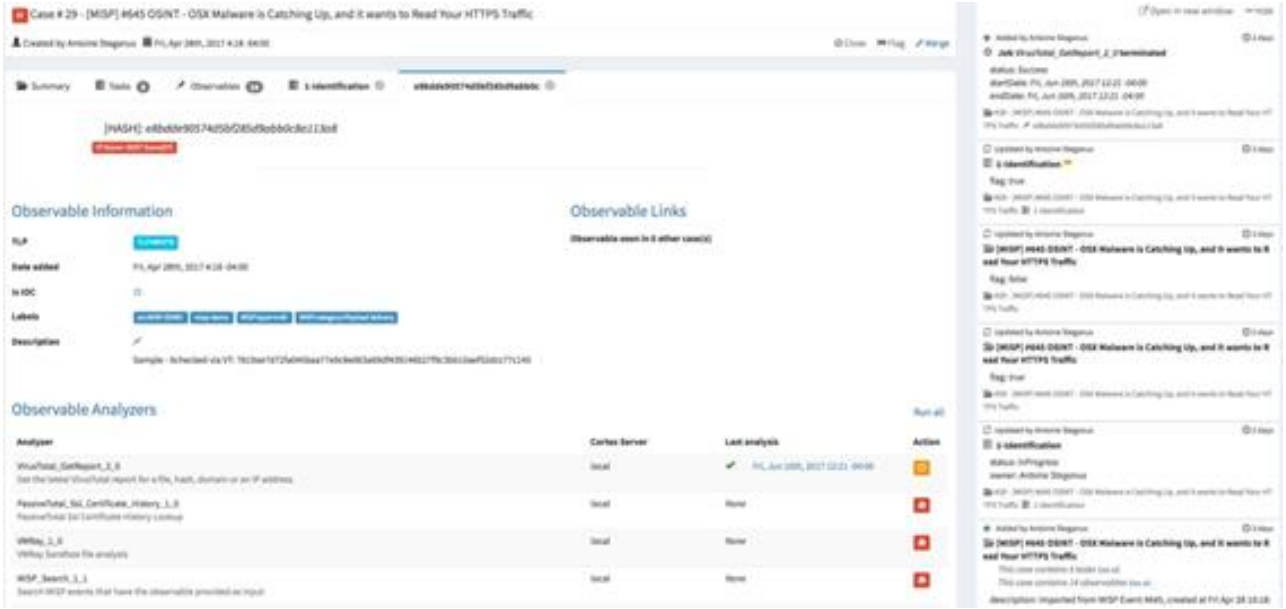

Figure 31: Cortex view [194]

TheHive Project can provide to the context of Project SPARTA a comprehensive threat database such as MISP, with frequently updated events, which together with the STIX2 standard would allow us to collaborate and exchange information in real time. The recommended hardware requirements for all components of the platform are:

Number of users	CPU	RAM
< 3	2 cores	4 - 8 GB
< 10	4 cores	8 – 16 GB

Number of users	CPU	RAM
< 20	8 cores	16 – 32 GB

Table 20: Cortex Requirements

The Table below summarizes the list with the strengths and weaknesses identified by the SPARTAN stakeholder of the tool, which were discussed and agreed assuming its application for assisting threat prediction.

Strengths	Weaknesses
• Open source. • Comprehensive and efficient data-base. • STIX2 standard. • Numerous analyzers. • Automated incident response.	• Difficult deployment process. • High hardware requirements.

Table 21: Strengths/Weakness Cortex

3.4.3 Alertflex

Alertflex is a SecOps (ITSecurity Operations) solution that helps assemble processes and technology to deal with IT security risks and threats. The solution functions as a security event manager and orchestrator for a distributed network of security sensors and scanners in hybrid IT infrastructure (on-premises and cloud). Alertflex can monitor different types of platforms e.g., Windows, Linux, Docker and provides a single user interface. Most of the products are free open-source software in the areas of IDS and DevSecOps, which can be unified by Alertflex.

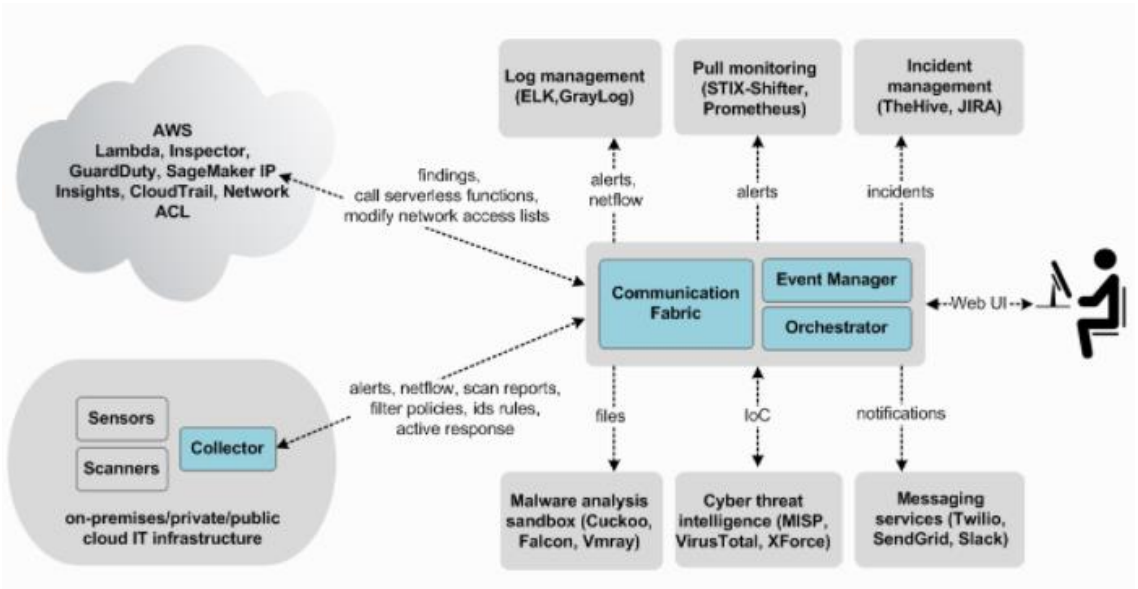

Figure 32: Alertflex high level design [195]

Alertflex implements security event management functions for a distributed network of security sensors such as Suricata NIDS, Wazuh and others. It collects the information via Alertflex collector, in streaming thanks to ActiveMQ, analyzes it with Alertflex controller, stores it in MySQL and allows access to it with the Alertflex controller and its management console. When a security alert matches a certain response profile, the controller sends the alert context to the management console to perform a response action, such as notifying the user or invoking an automation playbook.

The solution consists of distributed software components: Collector (Altprobe), Controller and Management Console. Alertflex Controller and Management Console build up the Central node (Cnode).

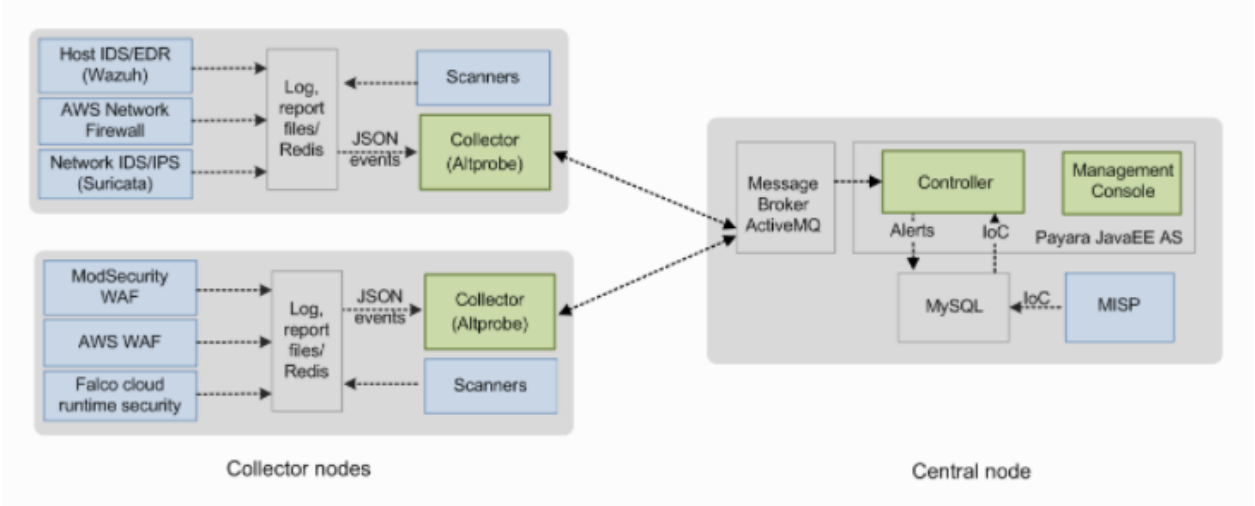

Figure 33: Alertfelx low level design [196]

The following screenshot shows the web interface of the Alertflex management console. It highlights that MISP is tightly integrated with Alertflex through MySQL and can be installed from a Docker image.

Figure 34: Alertflex Management Console [197]

Requirements:

- Cnode

Operating system	RAM	Disk space
Ubuntu 16.04 Ubuntu 18.04 Centos 7	≥ 8 GB	≥ 40 GB

Table 22: Cnode Requirements

Next TCP/UDP ports should be open for interaction between solution components and users:

Port	Application	Type
22	SSH	Linux console
443	Alertflex	web console
1514	Wazuh	Wazuh agent (only for appliance)
8443	MISP	admin ui (optional)
61617	ActiveMQ	altprobe (for remote collectors)

Table 23: Port should be open in Cnode

- Altprobe

Operating system	RAM	Disk space
Ubuntu 16.04 Ubuntu 18.04 Ubuntu 20.04 Centos 7	≥ 2 GB	≥ 20 GB

Table 24: Altprobe Requirements

The Table below summarizes the list with the strengths and weaknesses identified by the SPARTAN stakeholder of the tool, which were discussed and agreed assuming its application for assisting threat prediction.

Strengths	Weaknesses
• Open source. • GitHub community. • Developed since 2016. • High integration (MITRE, MISP, Suricata …).	• Difficult deployment process. • Few documentation.

Strengths	Weaknesses
• Cyber security operations automation and response. • Alerts filtering, prioritization, and visualization. • Detection threats, misconfigurations, vulnerabilities.	

Table 25: Strengths/Weakness Altprobe

3.4.4 Shuffle

Shuffle originated as one of the first open-source SOAR solutions. Although nowadays it offers many different subscription plans, an open-source version is still available. This solution acts as a SOAR using two main tools: Apps and Workflows. Apps are plug-and-play functionality that offer integration with other applications. They mainly rely on OpenAPI, a Web API standard. Currently Shuffle has apps for tools like The Hive, Cortex, VirusTotal and many others.

The second one, Workflows is where everything comes together. Combining Apps, Triggers and Variables, workflows act as playbooks to react to possible threats on the system, or even to prevent possible threats.

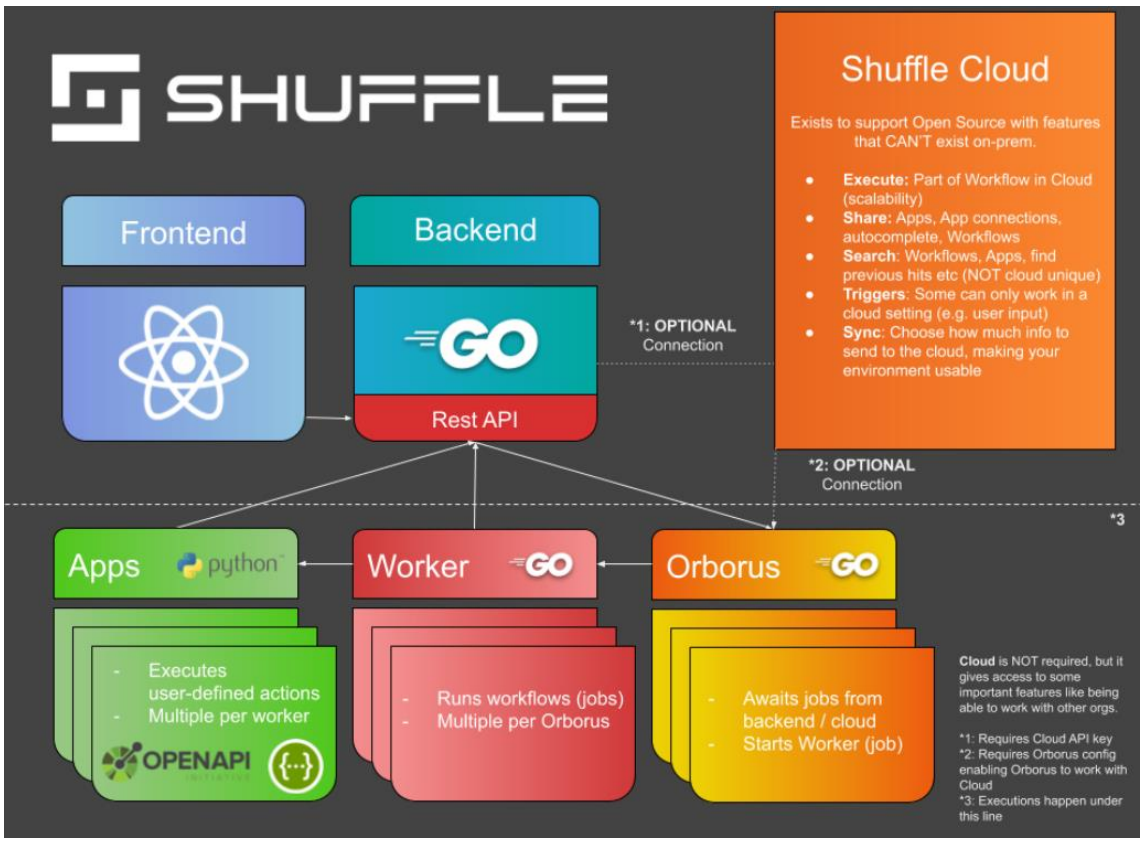

Figure 35: Shuffle View [198]

The Table below summarizes the list with the strengths and weaknesses identified by the SPARTAN stakeholder of the tool, which were discussed and agreed assuming its application for assisting threat prediction.

Strengths	Weaknesses
• Open source. • Highly documented. • Customizable. • Many integrations. • Available on Docker. • Simple to use.	• It's in Beta.

Table 26: Strengths/Weakness Shuffle

3.5 Prediction Framework

In this section we will summarize some popular prediction frameworks that could be used for the implementation. These solutions will be analyzed using the SWOT method within the next section of this document.

The prediction framework is a keystone for the threat prediction and risk analysis as it is the component which analyzes the data through different data model approaches. The use of frameworks helps implementing and accelerating the predictive process.

3.5.1 AIS

Automated Indicator Sharing (AIS) is a CISA capability that enables the real-time exchange of machine-readable cyber threat indicators and defensive measures to help protect participants of the AIS community and ultimately reduce the prevalence of cyberattacks. It uses open Standards STIX and TAXII for machine-to-machine communications.

3.5.2 WEKA 3

It is an Open-source technology. WEKA 3 is designed to be fast and flexible analyzing different datasets. It provides a repository of packages with implementations for a large set of classical algorithms variations, and it is interoperable with other scientific software (such as Matlab). In addition, it includes a wide variety of learning algorithms, as well as an extended range of preprocessing tools. Moreover, no programing needed, allows writing in Java and it is a nice and clean graphical output. Finally, several data mining techniques are available like clustering, classification, regression.

3.5.3 TENSORFLOW

Tensorflow is an open-source platform orientated on machine learning, it consists of various libraries and tools that help developers building machine learning models such as neural networks. Tensorflow works with ease in python notebooks, having available libraries for javascript and for mobile deployments. It has a big community supporting it, as well as a really complete documentation, making it the primary choice for many people when trying to build a machine learning model. It has the advantage of being scalable; while other frameworks get worse in performance when we increase the processor parallelism, TensorFlow continues to improve times even with 32 threads running in parallel.

3.5.4 Pytorch

Pytorch is an open-source framework that has the advantage of being supported by Meta. It supports dynamic graphs and is therefore a more mature technology in this respect compared to other similar frameworks while also having native support for asynchronous execution in Python. It has great support for distribute machine learning thanks to a recently released library called moolib. It has an active developer community, providing better and faster support for CUDA latest versions than other competitors, while also having complex models developed by the community itself.

3.5.5 Scikit-learn

Scikit-learn is a widely used open-source library for python, it is built on top off libraries such as NumPY, SciPY and MatPlotLib, some cores being built using Python to improve the model performance. Scikit-learn provides us with classification, regression and clustering algorithms among others. It also has a great community in which is easy to find help and support, also making it improve quickly.

3.5.6 Caffe

Caffe's greatest strength is its speed. It can process over 60 million images a day with an appropriate GPU. This consists of a rate of inference of 1ms per image and learning rate of 4ms per image. It is one of the fastest solutions available.
Being open source, it allows community contributions. Thanks to these contributions and its popularity, state-of-the-art frameworks have been developed. Available on Python, Matlab and C++. Expressive architecture to define and optimize models without needing to hard-code.

3.5.7 H2O.ai

It is a cloud solution supported in Java, R and python programming languages provides feature transformation, machine learning, explainable AI, machine learning operations and an AI app store. It is distributed in memory processing, and it is an Easy ingest data directly from HDFS, Spark, S3, Azure Data Lake and other formats from Big Data infrastructures. Furthermore, it is well documented and suitable for fast training, and it has a powerful Auto ML tools.

3.5.8 Deep Learning Toolbox

DLT (Deep Learning Toolbox) is a MatLab framework that provides designs and implementations for deep neural networks. DLT can exchange models with other technologies, such as TensorFlow and PyTorch through ONNX, and import models from TensorFlow-Keras and Caffe; furthermore, supports transfer learning with many pretrained models, such as ResNet-50, NASNet or DarkNet-53. The performance can be speeded up on a single or multiple GPU workstation (using Parallel Computing Toolbox) or scaled up to clusters/clouds like NVIDIA GPU Cloud or AWS EC2 GPU instances (using MATLAB Parallel Server).

3.5.9 Chainer

Chainer is a powerful, flexible, and intuitive deep learning framework [199] that supports CUDA computation, also running in multiple GPUs simultaneously making it faster than most other popular frameworks. Chainer supports several network architectures like as recursive networks, recurred networks or feed-forward networks. Furthermore, the code is very intuitive because off it can include any statements of python without lacking the backpropagation ability. In addition, the code is very easy to debug.

3.5.10 GNU Octave

GNU Octave is an open-source scientific programming language, it is similar to Matlab being the syntax very similar. In addition, some of Octave's scripts are compatible with Matlab. Furthermore, it is developed in C++ thus supporting many of the functions of the standard libraries. It allows large number of operations, including the visualization of data in 2D or in 3D. Moreover, GNU Octave provides the object-oriented. It runs in Windows, GNU/Linux, macOS and BSD. Finally, the framework is very useful and easy to use.

3.5.11 RapidMiner

It is a framework that allows the analysis and the data mining. Furthermore, it is developed by Java although includes R and python modules. Moreover, RapidMiner is a cross-platform environment, and it includes graphics and tools than allow to visualize data. Other characteristic of this tool is that

the representation of internal data analysis is in XML files. Furthermore, it can be used of several ways like as through of GUI, in command line, in both or reference a library from another program.

3.6 SWOT Prediction Framework Analysis

The Strengths, Weaknesses, Opportunities and Threats analysis (SWOT) evaluates the internal strengths and weaknesses, and the external opportunities and threats in a competitive environment. The internal analysis is used to identify resources, capabilities, core competencies, and competitive advantages inherent to the methodology under review. The external analysis identifies market opportunities and threats by looking at competitors' resources, the industry environment, and the general environment. The name is an acronym for the four parameters the technique examines:

- **Strengths**: characteristics of the object of study that gives it an advantage over others.
- **Weaknesses**: characteristics of the object of study that places it at a disadvantage relative to others.
- **Opportunities**: elements in the environment that the object of study could exploit to its advantage.
- **Threats**: elements in the environment that could cause trouble for the object of study.

In the next point, the prediction frameworks have been analyzed using the SWOT method in order to evaluate if their use it appropriate for this project.

3.6.1 SWOT Matrix + Requirements Collection

In this table we find SWOT's analysis for the tools defined in section 3.5.

	Strength	Weakness	Opportunities	Threats
AIS	Automated Indicator Sharing (AIS), a CISA capability, enables the real-time exchange of machine-readable cyber threat indicators and defensive measures to help protect participants of the AIS community and ultimately reduce the prevalence of cyberattacks. Use open Standards STIX and TAXII for machine-to-machine communications.	AIS is a free service but to participate in AIS between the steps to be completed it: • Obtain a PKI certificate from Federal Bridge Certificate Autority (having to be purchased if you do not already have it). • Sign an interconnection agrement and provide your IP address to CISA.		
WEKA 3	Designed to be fast and flexible analyzing different datasets. Repository of packages (https://weka.sourceforge.io/packageMetaData) with implementations for a large set of classical algorithms variations. Interoperaable with other scientific software (such as Matlab) Open Source technology It includes a wide variety of learning algorithms, as well as an extense range of preprocessing tools. No programing needed. Written in Java. Nice and clean graphical outputs. Serveral available data mining techniques like clustering ,classification ,regression.	Limited literature available and online support in case of issue. Learning complexity Previous knowledge in java required for users No sequential or multi-raltional model supported Uncustomizable analysis	Reproducible result on different softwares Runnable in almost every platform Youtube free online courses Several packages for extra utilities availabilities FAQ Blog available on the university of Waikito mail list Continuously developed and parched	It is not easy to connect Weka with a separately developed user interface Unable to run on certain IDEs Unpopular file format. No CSV supported

	Strength	Weakness	Opportunities	Threats
TENSORFLOW	Developed by the largest Internet company in the world, Google, that is, it has the support of this great company which has driven its rapid development and has also established important improvements that make it easier to implement and use as is the case of the Google cloud (allows accelerated performance) or an online hub (allows sharing models created with the framwork). TensorFlow has the advantage of being scalable; while other frameworks get worse in performance when we increase the processor parallelism, TensorFlow continues to improve times even with 32 threads running in parallel. Great user interface and architecture which allows easy implementation of computations on multiple platforms such as CPU, GPU and others. In addition, this architecture is flexible and has support for other languages such as C++, python and R and can be used directly to create Deep Learning models. Open source Keras built-in high-level API enable fast development of DL models. Visualizing training with Tensorboard.	Deprecated and legacy methods, style guide. Debugging method. Hard to make quick changes. Does not provide the best suppoort for distribute training. Not suitable for sharing dedicated hardware such as GPUs since TF sessions block them.	One of the best places to start your deep learning journey as a beginner, TF is a platform that allows you to easily implement or generate OA models. Create customized tools for different types of industry.	TensorFlow 2 has a known incompatibility with Python pickling. It is possible to find it if you use PySpark, HorovodRunner,Hyperopto or any other package that depends on pickling. The solution is to explicitly import TensorFlow modules inside functions.

	Strength	Weakness	Opportunities	Threats
	Production-ready thanks to TensorFlow serving. Easy mobile support via Tensorflow Lite. Easy web support via Tensorflow JS. Production-ready thanks to TensorFlow serving. Edge computing support via extensions Tensorflow Lite and RT for model shrinking/quantization Good documentation and community support.			
Pytorch	• Supported by Meta (Facebook). • It originally supports dynamic graphs and is therefore a more mature technology in this respect compared to others (eager TF). • Native support for asynchronous execution from Python. • Great support for distribute ML via recently released moolib. • Large availability of complex models open sourced by community. • Active developer community, provides better and faster support for CUDA latest versions than competitors.	• Not production ready. It does not provide native extensions to productionize models. • Relatively new, less mature suite of products.	• Since it attracted a large part of the scientific community that was mostly developing in theano. Many of the latest published models are realized under this framework. • State of the Arts tools for scalable reinforcement learning.	• Dependency on third party software for visualizing (i.e. Tensorboard, MLflow, etc) • No high level API provided for fast prototyping, it implies a slower learning curve for new adopters.

	Strength	Weakness	Opportunities	Threats
Scikit-learn	The project scikit-learn started in 2007 and is currently maintained by a team of volunteers. Mainly developed in Python, some cores are developed using Cython to improve the models performance, such as the SVM, SMVL and the logistic regression module.	Under certain conditions, the performance of the models created using scikit-learn can be defective as the performance is related to the hardware running the models. Aldo, the use of this library requires basic python programming skills.	As scikit-leanr is under continuos development, it is constlantly imporoving and adding more characteristics, options and models. Furthermore, more capabilities can be added by interacting with other frameworks or libraries, like the plotting capabilitie adquired by using the library Matplotlib or the scikit-image functions, which require Pandas or seaborn (depending on the used function).	As this library is under development by a team of volunteers, the maintenance isn't actually asured.
Caffe	Caffe's greatest strength is its speed. It can process over 60 million images a day with an appropriate GPU. This consists of a rate of inference of 1ms per image and learning rate of 4ms per image. It is one of the fastest solutions available. Being open source, it allows community contributions. Thanks to these contributions and its popularity, state-of-the-art frameworks have been developed Available on Python, Matlab and C++ Expressive architecture to define and optimize models without needing to hard-code	Caffe is focused mostly on image processing. Therefore, convolutional networks are the most common model. Data processing is not possible. Some type of networks like recurrent networks are not available. Few input and output formats	Spark integration available and backed by Yahoo GPU clustering possible for maximum efficiency Already being used in academic research projects, start-ups and even industrial applications	Uses may 3 party packages. Depending on the compatibility between them updating can be a problem since uptades are done asynchronously.

	Strength	Weakness	Opportunities	Threats
H2O.ai	• Cloud solution supported in Java, R and python programming languages provides feature transformation, machine learning, explainable AI, machine learning operations and an AI app store. • Distribute in memory processing. Easily ingest data directly from HDFS, Spark, S3, Azure Data Lake and other formats from Big Data infrastructures. • Well documented and suitable for fast training • Powerful Auto ML tools.	• No support for edge computing • Lack of collaborative features • Expensive for small-sized organizations and will be a deterrent for early adoption	• H2O is much easier for scaling and productionizing than scikit learn	• Since the technology performs better in Big Data scenarios it has been promoted as a cloud solution, few people consider it as a stand-alone tool to implement without H2O clusters.
Deep Learning Toolbox	DLT is a MatLab framework that provides designs and implementations for deep neutal networks. DLT can excange models with other technologies, such as TensorFlow and PyTorch through ONNX, and import models from TensorFlow-Keras and Caffe; furthermore, supports transfer learning with many pretrained models, such as ResNet-50, NASNet or DarkNet-53. The performance can be speeded up on a single or multiple GPU workstation (using Parallel Computing Toolbox), or scaled up to clusters/clouds like NVIDIA GPU Cloud or AWS EC2 GPU instances (using MATLAB Parallel Server).	DLT is not an open-sorce solution; hence, DLP licenses are a paid service, with plenty of psicing/licensing options available depending on the purpose, the users that will access and the periodicity.	As DLT Framework is a solution developed and maintained by MathWorks, the company offers support to the users, along many courses and resources to use the framework.	Since the MatLab performance ban be linked to he hardware running it, the use of a GPU workstation or a cloud/clutch solution is almost essential, such as the dependencies corresponding. Hence, the cost of the implementation can be highly increased.

	Strength	Weakness	Opportunities	Threats
Chainer	Chainer can make use of CUDA/CuDNN computation allows GPU management to ensure the best performance Easy to learn and use using define-by-run approach. Faster than many popular frameworks	The biggest weakness is the lack a strong community in comparison with other frameworks. This means that finding documentation, support and addons can be harder.	Chainer is available on Docker which can make integration much easier.	Chainer is currently in a maintenance phase, so no development is being made. Therefore, state-of-the-art functionality can only be available by community updates and addons.
GNU Octave	GNU, free equivalent to MATLAB Includes an Interpreter that can be used via shell, script and GUI Capable of resolving lineal and no lineal numeric problems Developed in C++ Different functions can be added by dinamic modules Linux and Windows compatible Different languages help resources, including spanish	Own programing language, add dificulties to users Octave does not integrate CAD and PLM well Dificulty to master Windows version has library path issues Slow runtime for relatively large input	Language is similar to MATLAB's, this enhances the user learning It supports most of the functions of the standard C library As it is licensed under the GNU General Public License, it can be shared and used freely. Suitable for bash lovers.	Not all MATLAB's functionalities are included, the user has to write them out No written modules/functions/libraries available for ML programs Limited libraries and toolbox Similarity to MATLAB can resolve in a continuous comparative
RapidMiner	RapidMiner offers a visual workflow, easy to use and learn, with a drag&drop approach, with plenty of predictive models. It's a unified open source platform, with plenty of connectors available and a community of +250K data scientists.	High storane and processing requirements, as well as high implementation/customisation cost. Some customisations are limited and doesn't allow changes on predefined models.	As it is a drag&drop easy-to-use solution, even non data-scientists are able to create models reproducibles on different softwares.	As some customizations are limited, other solutions could be more appropiate in order to apply some models.

Table 27: SWOT Analysis

The main features that describe this table are the function that provides the tools to complete the requests of the project. For example, the programming language and advantages that are required to use it in the project, other aspect that are comment are the disadvantages that they contain. Other features that are necessary mention are the security of the tools and the amount of human detection. In addition, it is important know that the mean of the character ~ is that the features can be adapted.

	WEKA	Tensor Flow	PyTorch	Scikit-learn	Caffe	H20.ai	Deep Learning Toolbox	Chainer	GNU Octave	RapidMiner
Extensibility, Expandability and Multi-level scalability by design	NO	YES	YES	YES	YES	YES	~	YES	NO	YES
Non-Stationarity	NO	YES	YES	YES	YES	N/S	YES	YES	N/S	NO
Uncertainty	YES	~	~	~	~	~	YES	~	~	~
Open solutions and COTS reliant	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Privacy by design	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Security by design	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Ethics and trustworthy AI by design	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Human-centred by design	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Full-spectrum cyber security threats	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Threat projection at different time horizons	~	YES	~	~	~	~	~	~	~	~
Non-Determinism	~	YES	~	~	~	~	~	~	~	~
Reduce the human intervention	N/S	N/S	N/S	N/S	N/S	~	N/S	N/S	N/S	~
Notification and Information sharing	~	~	~	~	~	YES	~	~	~	YES

Table 28: Requirements

Chapter 4 Side-wide Design Decisions

Based on the revision of the state of the art described above, this section delves into the proposal design principles, hence describing its objectives, assumptions, requirements, limitations and architecture.

4.1 Framework Objectives

The main purpose of the present research (SPARTA deliverable D4.4) is to introduce a cybersecurity threat prediction framework, which includes the definition of approaches, methods to be used and process organization to provide comprehensive prediction of full-spectrum cybersecurity threats. Taking this into consideration, the framework is complemented by a novel architecture for anticipatory cyber response for emergent communication/information environments. As secondary targets, the proposal provides an adaptive solution able to deal with non-stationarity and data source heterogeneity [200], taking advantage of the self-similarity properties of the analyzed cyber activities [201] and an emerging ICT infrastructure. This includes a detailed presentation of each framework component, illustrative use cases within the joint SPARTA project demonstrators, and an analysis of the evaluation methodologies (procedures, performance indicators, datasets, etc.) suitable for assessing operational feasibility. The research concludes with a discussion on how each requirement was embraced, implementation considerations and lessons learned.

4.2 Initial Assumptions

In order to restrict and lay the foundations of the performed research the following premises have been assumed:

- **Assumption 1**. The design and implementation of an effective framework for anticipatory threat prediction on full-spectrum dimensions and the emergent communication landscape is possible, being this the *alternative hypothesis* of the performed research. The opposite constitutes our research *null hypothesis*.
- **Assumption 2**. The cyberspace as a whole displays non-stationary characterization, so an adequate anticipation of threat situations suggests dealing with the concept drift [202].
- **Assumption 3**. The self-similarity property will be present in emerging environments like 5G communication networks, but its significance varies according to the use-case instantiation and the operational needs [203].
- **Assumption 4**. Attacker may exploit concept drift situations in order to gain effectiveness or evade intrusion detection and prevention systems [204].
- **Assumption 5**. A proper projection of the cyber situations demands a continuous consultation and update on heterogeneous data sources, including Cyber Threat Intelligence (CTI) providers [205]

4.3 Requirements

The following requirements aim on ensuring the effectiveness of the operation and instantiation of the proposed framework at different scenarios.

- **Req 01 - Extensibility, Expandability and Multi-level scalability by design**. The dynamism required for the proper instantiation of the proposal lies on three major properties: extensibility, expandability and multi-level scalability. The first of them adopts the extensibility by design paradigm, which is achieved through proving a modular design that allow third parties to create/modify their threat prediction frameworks. Similarly, the deployment must be expandable, in this way allowing the inclusion of new technologies and resources. With

this purpose. In addition, the proposal also must be multi-level scalable, which is crucial for dealing with the numerous and unexpected management issues on large scale mobile communication environments. As described in the following sections, this is achieved by a vertical architectural hierarchization.

- **Req 02 - Non-Stationarity.** Most of the machine learning and datamining methods, especially those that are oriented to anomaly recognition, assume the premise that the collections of reference data present stationary distributions. They also assume that the information they analyze is gathered from a monitoring environment of static characteristics, a situation that is not always satisfied in their deployment at real use cases. To deal with this variability poses a mandatory feature for properly addressing analytics on the emerging communication scene, so the proposed anticipatory solution must be able to operate on non-stationary factual knowledge.
- **Req 03 - Uncertainty.** Conventional logic was grounded by exact reasoning, where it is assumed that the perfect knowledge always exists. But given the dynamism inherent in real cyber situations, as well as the effectiveness expected from the existing solutions for situational awareness and cognitive management, uncertainty is an essential part of the knowledge that will be processed and managed [206]. Consequently, the proposal must adapt to the probabilistic reasoning approaches commonly implemented for adapting the cyber decision loops as more information of the situation is revealed.
- **Req 04 - Open solutions and COTS reliant**. A feasible implementation of the framework shall prioritize economic and temporal efficiency by maximizing the adoption of existing prebuilt components and prototypes (COTS, open-source solutions, OSINT studies). Hence this shall assume and facilitated by design, which not only affects software but pre-built AI models, datasets, etc.
- **Req 05 - Privacy by design**. Security and privacy guaranties must be provided, by way of the GDPR or equivalent ones, to build a sustainable ecosystem of benefit to all the stakeholders. Transparency of practices, user control over their data, and accountability are key requirements to build this trust and to be compliant with the EU ethics requirements and current legislation, which will be assumed by the threat prediction framework instantiations by design.
- **Req 06 - Security by design**. The framework once instantiated will be part of the ICT to be deployed in real operational environments, being also susceptible to attacks, failures, and misconfigurations. With the purpose of minimizing the potential attack surfaces, the framework will assume the design of hardening and self-protective capabilities by design, which will be explicitly covered during the effort dedicated to establishing Common principles for Audition and Self-Protection.
- **Req 07 - Ethics and trustworthy AI by design**. The framework will heavily rely on advanced AI capabilities, as is the case of machine learning algorithms. Bearing in mind the increasing efforts of the EC towards fostering and securing ethical and robust AI within the European Markets, which also impacts the military industry. The guidelines published in the EC Ethics Guidelines for Trustworthy AI will be assumed by design, as well as other ongoing EU-level.
- **Req 08 - Human-centred by design**. In order to break the adoption barriers of the AI-driven automatism and better fit the human operational needs (human in the loop), the framework will involve the human perspective in all steps of the problem-solving process when instantiated (observing the problem within context, brainstorming, conceptualizing, developing, and implementing the solution). The overall threat prediction approach will from the outset be geared towards assisting humans and not the other way around, thus prioritizing ease of use, explainability, interpretability, etc.
- **Req 09 - Full-spectrum cyber security threats**. The proposal shall be instantiable for dealing with all cyber threats of all dimensions, including hybrid cyber activities. In particular disinformation campaigns, fake news in combination with targeted cyber operations, varying in

scope, scale, duration, intensity, complexity, sophistication and impact, are getting increasingly common and demand a mobilization of the full range of intelligence tools and instruments, and new ways to respond, where prediction/anticipation became essential for early response.

- **Req 10 - Threat projection at different time horizons**. The framework shall be able to combine predictors for projecting the situation at short, mid, and long terms; thus suggesting a fast way for early anticipate imminent cyber situations, but also resolving estimations about how they may evolve at large observation periods.
- **Req 11 - Non-Determinism**. As complement of the above, the projection framework shall be compatible with non-deterministic forecasting processes. In this way, if there are different prognosis for threat situations, decision-makers are allowed to identify, assess and plan countermeasures for each of those regarding their likelihood of occur.
- **Req 12 - Reduce the human intervention**. The framework shall prioritize to reduce the human participation in the prediction process (if allowed), thus exploring supervised, semi-supervised and unsupervised automation.
- **Req 13 - Notification and Information sharing**. The proposal shall be able to communicate in an understandable and harmonized way the predicted threats, thus allowing triggering SOAR reactions against them. The notifications should include information about the projection time, likelihood, alternative prognosis (see non-determinism requirement), nature of the threat, how it was inferred, etc.

4.4 Limitations

Due to different circumstances, the following issues have not been taken into account, so they were postponed for future work.

- **Limitation 1**. Some of the implemented machine learning solutions demand reference and validation datasets, that must be complete, significant and property labelled. Given the difficulties inherent in produce/share threat related information, the presence of suitable datasets for all threat situations cannot be guaranteed.
- **Limitation 2.** The prediction of threat situations that demand to build models of the normal usage of the protected system (e.g., anomaly based intrusion detection) may require a continuous upkeep of the related datasets. It is up to the framework instantiates and operators to collect and provide this information.
- **Limitation 3.** Although the selection of a proper data granularity usually leads to considerably improve the accuracy of the performed analytical tasks [207], the proposed anticipatory solution does not delve into the existing methods; instead, the data granularity was directly provided by the use case specifications.
- **Limitation 4.** The actuation of elements/agents capable of preventing access to the traffic circulating through the monitored environment, the obfuscation of their contents, and the generation of noise, are frequent at the current communication landscape [208], which may substantially hinder the analytic processes. In order to improve the understanding of the performed research, these kinds of circumstances were not taken into consideration.

4.5 Other early considerations

Considering the previous requirements and limitations, a prediction framework was developed. In order to do so, the following considerations were heeded:
One of the main strengths in a prediction model resides in the quality of the ingested data, therefore the first important step is to obtain valid data sources. The main challenge while collecting data is to obtain data which accomplish the desired quality, which means non-stationary and heterogeneous data. In order to do so, both internal and external data sources are considered, and both will be described later in this document.

As all data sources might not provide normalized data, an intermediate step should be considered in order to digest the data, similarly to the Transformation step in a ETL process. This data processing will digest the ingested information with the objective to prepare the data for the subsequent analysis.
Finally, the data will be analyzed using prediction models in order to achieve trustable predictions and decisions. As the prediction modelling offers statistical outputs to predict and forecast likely future cyber-events, the implementation of multiple models should be considered as its performance is yet to be evaluated; different ML/AI areas and models will be later introduced and analyzed.
It is appropriate to keep in mind that the performance is strictly related to the data sources, thus the predictions and forecast likelihood will likely improve as the data ingested increases, and that the initial performance might not be as accurate as it would be desired.

Chapter 5 Situational Awareness prediction framework

This section focuses on explaining the most relevant aspects that define the Situation Awareness prediction framework, such as architecture design, APIs and interfaces, knowledge representation and data models, concepts of execution and external data sources.

5.1 Architecture design

In order to facilitate the understanding of the functioning of the Situational Awareness prediction framework we proceed to describe in detail its architecture, as well as each of its component blocks and their interaction with the rest of the system. The division of the framework into different blocks has been made according to their purpose or functionality within the overall system as a whole.

As described in section “Side-wide Design Decisions”, for the design of the framework, the initial limitations and assumptions have been considered, as well as all the necessary aspects that will make it possible to achieve the established objectives and ensure compliance with each of the project requirements.

As a result, the Situation Awareness prediction framework has been obtained, which is composed of 5 main blocks: Data Source, Data Processor, Data Storage, Data Analyzer and Data Manager. The following figure shows a schematic representation of the design architecture and how its different components are related to each other.

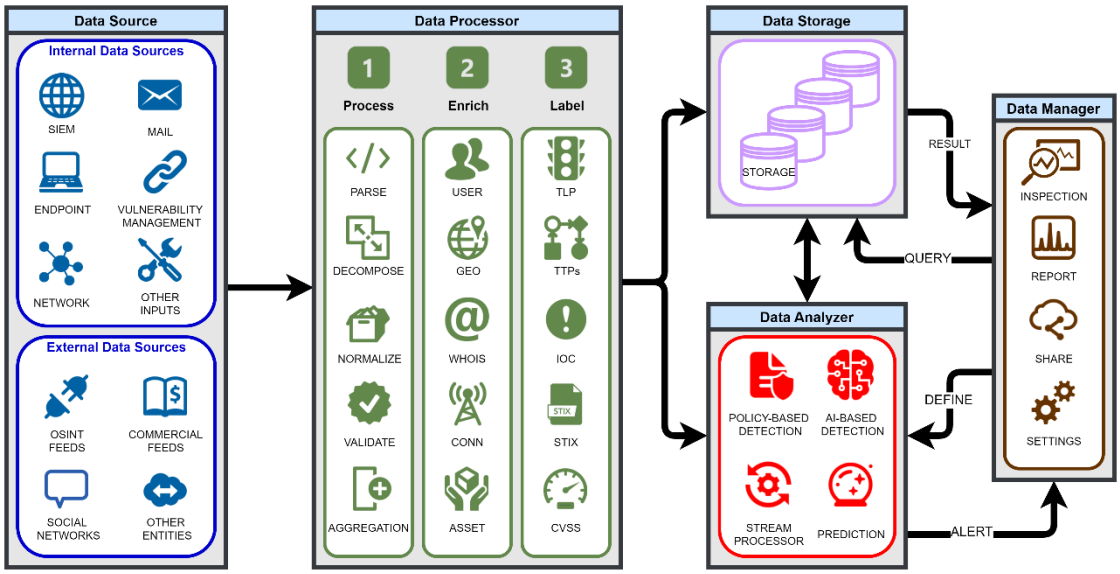

Figure 36: Prediction Framework Architecture

5.1.1 Data Source

Within the scope of Cyber Threat Intelligence (CTI) we find that all feeders provide contextually enriched information concerning actors, threats, and vulnerabilities presented to enhance the decision-making process and heighten the consumer’s security posture. The main challenges to be faced when collecting all these data are closely related to two qualities that characterize the sources of information consulted: non-stationarity and heterogeneity.

Generally, the collection of threat-related information takes place in dynamic environments, where new technologies, techniques and actors are constantly emerging. As a consequence of this variability, it can be assumed that the data collections that will serve as reference for the training of

machine learning and data mining models oriented to anomaly recognition present non-stationary distributions.

The objectives of the project are to cover the full spectrum of cyber security threats and all their dimensions, giving rise to a wide variety of different types of information-generating elements. From each of these information-generating elements, data sets are obtained which, depending on the source, may differ from each other in format, structure, organization, complexity, value, validity, , scope and representation, among many other aspects. This heterogeneity of information sources implies that once the raw data is collected, it will need to be processed before it is suitable for processing.

The Data Source component has the sole objective of ingesting and compiling information from numerous sources of different nature and origin, which is necessary to provide comprehensive prediction of full-spectrum cybersecurity threats effectively. The implementation of this component is intended to provide an adaptive solution for framework data acquisition capable to cope with non-stationarity and heterogeneity of information sources, taking advantage of the self-similarity properties of cyber-activities analyzed by other external CTI infrastructures.

Within the data sources that will feed the Situation Awareness prediction framework, a division has been implemented establishing as a distinction criterion the origin of the information generation, from which two different types of data sources can be distinguished: Internal Data Sources and External Data Sources.

Internal Data Sources:

Internal data sources are those elements that generate organized data sets potentially relevant for the anticipation of security threats, which are part of the same network infrastructure in which the Situation Awareness prediction framework is located. These are different agents distributed throughout the system that have very diverse tasks not necessarily related to security, but whose common denominator is that their activity provides information that must be analyzed to improve the decision-making process.

Among the internal information sources, the following can be highlighted as the most important, significant, and enriching for the Situation Awareness prediction framework[1]:

- **SIEM:** Security Information and Event Management is a key data source because it is an entity specifically dedicated to the collection and correlation of security-related events that facilitates the detection of unusual trends and patterns that may pose a threat.
- **Mail:** One of the main attack vectors within the IT systems of any company is email. Lately, attacks that include BEC (Business Email Compromise) patterns in combination with phishing techniques based on social engineering have been proliferating, significantly increasing the effectiveness of phishing campaigns aimed at stealing confidential information. For this reason, it is crucial to collect as much data as possible generated from the email exchange, process and analyze it.
- **Endpoint:** There are numerous dedicated devices within a company's infrastructure, such as Antivirus, HIPS, HIDS, Firewall, DDoS or user equipment, which store the information generated from the operations they perform in system log files. The data stored in these log files is very valuable in the process of detecting and predicting cyber security threats.
- **Vulnerability management:** Vulnerability management is the process of identification, analysis, classification and treatment of risks and threats. The purpose of this process is to correct weaknesses, apply controls and minimize impacts, among others, to guarantee an adequate

[1] This when assuming a general purpose application of the framework, and based on the commonalities suggested by the stakeholders. Contextual-based or use-cased instantiations may rely on specific data sources in a greater way, but for obvious reasons it is out of the conducted study to explore every possible framework application and context.
This holistic cyber situation prediction approach will be a constant during the rest of studies, designs and decisions.

level of project security. The information gathered and generated in the development of this process constitutes a fundamental pillar in the knowledge base that the Situation Awareness prediction framework will use in the construction of threat prediction.

- **Network:** The devices that manage the traffic within a network infrastructure and the recording of the message exchange activity by members of that network become a fundamental source of data that allows the collection of evidence of what happened in the communication process.

Other inputs: Any data source that can provide relevant information in the field of security threats should be considered as a candidate to be part of the information gathering process. Examples of other inputs include CCTV systems, fire detection systems, sensors, risk assessment, penetration testing, security audits and SecDevOps procedures. Internal sources are obtained locally, at the network itself, and a new division can be made in two categories: low-level and higher-level.

In the category of low-level internal sources, we include those that obtain data directly from hosts or the network, without much processing or correlation. Some examples of such data are:

- Host data, e.g., syscall traces obtained with ptrace;
- Host performance data, e.g., obtained with Windows performance counters or mpstat;
- Events, e.g., syslog events or Windows events;
- Network packets or messages, e.g., obtained with libpcap or tshark;
- Flows, e.g., Netflow or Sflow;
- Network statistics from SNMP/RMON.

In the category of higher-level internal sources, we include those that involve more processing, possibly even correlation of data from several sources. Examples are:

- Intrusion detection systems and SIEMs;
- Antiviruses and other malware detectors;
- Firewalls and other middleboxes;
- Vulnerability scanners, port mappers.

External Data Sources:

The importance of sharing information regarding cybersecurity incidents/events among community members for the anticipation of security threats is well known. External data sources are those from reporters located outside the Situation Awareness prediction framework network infrastructure.

A cataloguing of external information sources can be made, where the most relevant ones for the Situation Awareness Prediction Framework are included:

- **OSINT Feeds:** OSINT feeds are publicly available threat intelligence data sources provided by organizations and individuals. OSINT includes information gathered from the internet, mass media, specialist journals and research, photos, and geospatial information. The list of the most suggested OSINT feeds between the participant stakeholders is as follows:
    - **@abuse.ch:** Ransomware Tracker collects data related to ransomware attacks so that security teams can check IP addresses and URLs against those that are known to be involved in attacks. The tracker provides detailed information on the servers, sites, and infrastructure that have been exploited by ransomware actors, as well as recommendations for preventing attacks.
    - **AlienVault:** Multiple sources including reputation indicators and large honeynets that profile adversaries.
    - **Cisco Talos Intelligence:** This feed provides information on known threats, new vulnerabilities, and emerging dangers. Talos also provides research and analysis tools.
    - **Facebook ThreatExchange:** Facebook created the ThreatExchange platform so that participating organizations can share threat data using a convenient, structured, and

easy-to-use API that provides privacy controls to enable sharing with only desired groups.

- **Google Safe Browsing:** This service identifies dangerous websites and shares the information to raise awareness of security risks.
- **SurfWatchLabs:** Insights tailored to specific projects.
- **SANS Internet Storm Center:** The Internet Storm Center, formerly known as the Consensus Incident Database, uses a distributed sensor network that collects more than 20 million intrusion detection log entries per day to generate alerts about security threats. The site also offers analysis, tools and forums for security professionals.
- **SecureWorks:** Provides feeds and instruments networks.
- **Symantec:** DeepInsight feeds on a variety of topics including reputation.
- **Spytales:** Complete information on the past, present and future of spies.
- **Team-Cymru.com:** Threat intelligence plus bogon lists.
- **TheCyberThreat:** High level but comprehensive and curated feed.
- **The Spamhaus Project:** Spamhaus is a European non-profit organization that tracks cyber threats and provides real-time threat intelligence. It has developed comprehensive block lists for known spammers and malware distributors, which they provide to ISPs, email service providers and individual organizations.
- **ThingsCyber:** The most critical lessons in cyber conflict and cyber defense.
- **ThreatIntelligenceReview:** Updated reviews of threat intelligence sources.
- **ThreatTrack:** Stream of malicious URLs, IPs and malware/phishing related data.
- **Virustotal:** It uses dozens of antivirus scanners, blacklisting services and other tools to analyze and extract data from files and URLs submitted by users. The service can be used to quickly check for incidents such as suspicious phishing emails, and each submission is retained in its database to build an overall picture of cyber threats.
- **VirusShare:** is an online repository of malware created and maintained by J-Michael Roberts, a digital forensics examiner. The site gives researchers, incident responders, and forensic investigators access millions of malware samples.
- Government sources of Cyber Threat Intelligence:
  - **The Defense Cyber Crime Center (DC3):** Through the newsletter and its Twitter feed, provide daily context on cyber threat and incidents.
  - **US Computer Emergency Response Team (US-CERT):** Facilitates response to major incidents and analyzes threats. Shares vulnerability information through alerts and announcements, delivering a wealth of tips and awareness items useful to your cyber threat intelligence program.
  - **Department of Homeland Security:** Private companies can report cyber threat indicators to DHS, which are distributed through the Automated Indicator Sharing website. This database helps reduce the effectiveness of simple attacks by exposing malicious IP addresses, email senders, etc.
  - **European Union Agency for Network and Information Security (ENISA):** Numerous references, publications, media.
  - **FBI Cyber Crime:** This feed provides information relevant to 16 critical infrastructure sectors. Public and private sector organizations can share security information and events. The FBI also provides information on cyber-attacks and the threats they are tracking.
  - **National Council of ISACs:** While some ISAC feeds are quite expensive, others are free. The National Council of ISACs provides a comprehensive list.
  - **StopThinkConnect:** Not much operational, but good background for the workforce. Striving to make cyber security understandable by people. Good tips for corporate and personal awareness programs.

- **Commercial Feeds:** Although they come at a cost, they have some advantages over OSINT feeds. Commercial threat intelligence vendors strive to ensure threat validation, checking

each indicator at least daily and updating the feeds hourly with any new data, thus achieving a rich data history. In addition, commercial threat intelligence vendors offer customization of specific threat feeds to meet the particular needs of their customers and present each threat stream in a standard format, or structured formats, that allow it to be directly connected to different security platforms and systems. Among the most suggested commercial feeds are the following:

- **Cisco AMP Threat Grid:** Cisco Threat Grid crowd-sources malware from a closed community and analyzes all samples using proprietary, highly secure techniques that include static and dynamic (sandboxing) analysis.
- **CrowdStrike:** CrowdStrike Falcon is highly targeted to prevent damage from targeted attacks, detect and attribute advanced malware and adversary activity in real-time, and effortlessly search all endpoints reducing overall incident response time.
- **Fox-IT:** Provides an intelligence database with global incident data to its customers, including government, defense, law enforcement, critical infrastructure, banking and commercial enterprises.
- **iDefense:** This is a contextual, timely and actionable security intelligence, enabling businesses and governments to make smarter decisions to defend against new and evolving threats.
- **FireEye iSIGHT**: FireEye iSIGHT Threat Intelligence is a proactive, forward-looking means of qualifying threats poised to disrupt your business based on the intents, tools and tactics of the attacker.
- **Kaspersky Threat Intelligence:** Kaspersky Threat Data Feeds contain information about suspicious and dangerous IPs, URLs and file hashes in existing security systems such as SIEM, SOAR and Threat Intelligence Platforms. It allows security teams to automate the initial alert triage process while providing them with sufficient context to immediately identify alerts that need to be investigated or escalated to incident response teams for further investigation and response.
- **Mr. Looquer IOC:** Threat feed focused on dual-stack systems (IPv4 and IPv6 protocols) that provides an extensive database of indicators of compromise (IOCs).
- **RecordedFuture:** Real-time threat intelligence from the web.
- **ReversingLabs:** Database that provides more than 50 reputation information services, threat feeds and malware feeds continually reanalyzed for the most up-to-date status.
- **ThreatConnect:** Implemented and maintained by Cyber Squared, it focuses on information exchange.
- **ThreatFabric:** Feed specialized in Mobile Threat Intelligence allows banks to track mobile banking malware campaigns targeting their banking applications and import information reports on malware families, their capabilities and their evolution.
- **ThreatStop.com:** Block Botnets by IP reputation.
- **Verisigninc.com:** iDefense feeds highly regarded by some key institutions.
- **ZeroFOX:** This company offers a highly contextualized threat feed comprised of social and digital based campaigns and indicators, pulled directly from social and digital data sources and analyzed by ZeroFOX's analysis engines.

- **Social Networks:** Social media profiles on sites such as **LinkedIn**, **Facebook**, **Twitter, Telegram** and **Instagram** are considered to be the most effective data source for gathering information about individuals. SOCMINT (Social Media Intelligence) is a type of Intelligence through which large amounts of information obtained from social networks can be collected, integrated and shared. SOCMINT allows to analyze conversations, personal interests, preferences and behaviors in online social interactions, so that a profile of the individual can be created, and their behavior pattern can be generated based on this knowledge.

- **Other entities:** Any external entity that can provide the Situation Awareness prediction framework with relevant security threat information is likely to be used as a feed. By leveraging the information shared by the threat management community, it can be used to increase the knowledge base managed by the framework. Very clear examples of this type of external data sources are any of the platforms explained in point “2.6 Relevant existing solutions” of this document, such as SIEM, CTIP and SOAR, managed by any external entity and reporting their findings to the community members.

5.1.2 Data Processor

Each tool and systems mentioned in the “Data Source” section generates a different type of event report, and these reports do not have to resemble each other since each manufacturer uses a different type of message, or even different standards for notification of events.

However, in order to process all event types received, with the purpose of obtaining a prediction of possible threats that may be happening, it is necessary to speak a common language, establishing a reliable standard where can reflect all the information necessary to describe the event which must be treated.

That is precisely the task of “Data Processor” module. It is a three-step-task, where the input will be all received events from internal data sources as SIEM, email, etc. or external data sources as OSINT Feeds, social networks, etc. and the output will be a standardized message with all the event information.

It is important to state that the processing pipeline may vary depending on the different products that are meant to feed data into the system or act as a part of it, and the rest of information sources. Depending on the product, most or part of this processing pipeline might be already automated and only requires of little tweaks and adaptations. On the contrary, relevant data sources of public non-structured or custom feeds might require the complete processing pipeline to be coded from scratch in an ad-hoc solution.

The three-step-tasks are as follows:

Process: This is the first step thought to prepare the data collecting and preprocessing information, in order to facilitate the subsequent work by establishing a standard message.
Enrich: The second step will enrich the standardized message with some relevant information like geolocation data.
Label: Third step aggregates a labelled system in order to expand information about possible event risks.

As a final result, a complete message will be obtained with all the relevant information about the detected event. The tasks contained in each of the steps are detailed below.

Process

Step “Process” have the responsibility to build a standardized message that has all the important information to describe the detected event.

The ingested information may be of different nature. There are different types of data according to the degree of suitability or adequacy to the objectives. Depending on this degree, the incoming data might be qualified among the following:

- Unstructured data: Raw data that needs to be processed and adapted to the customer needs and is not appropriate as it is.
- Semi-structured data: Data that has some internal structure, semantics or labelling but it is not directly operable without an intermediate processing.
- Structured data: Data that is suitable to the needs of the process and can be managed seamlessly.

The objective for the processing tasks is, to gather any of the previous types of data and generate structured data out of them, “cleaning” and curating the information in a way that it suits the requirements for further analysis, storage and visualization of it. For this, it has different work areas or tasks that must be fulfilled.

Figure 37: Process: First step in Data Processor

- **Parse**. Process step must have a subtask that could read the event from any raw data source and parse the information to do whatever subsequent process. A parser is generally a piece of software or middleware of little to very high complexity that acts as a data filter, cleaning all the non-relevant information of a source and fetching the initial information to feed the next procedure. It can be implemented as well as the first error controller, excluding out of the process any information that has not a valid structure or incorrect fields, if necessary.
- **Decompose**. The next task, that is almost a final step of the parser, is to take parsed information and extract and identify the relevant data for further normalization. This is a step that adds a semantic meaning to the data ingested.
- **Normalize**. Once the unordered information is obtained, it is necessary to normalize it following a previously defined standard that groups all the requirements to define the event. This step will gather the previous information and adequate to the customer needs, rules and shape.
- **Aggregation**. In addition, the message must have all the fields to define an event, not only the information collected from raw data. For this reason, all necessary fields to complement raw information must be available in order to be completed. This process will allow to reconstruct complex objects/entities and their attributes in a meaningful way.
- **Validate**. To validate the message obtained it is necessary to have a template or schema that indicates what kind of information can appear, the data type and the order that must be presented.

Usually, last three task take place at the same time, since when the normalized message is created, it can be verified that it complies with the proposed schema. All the process starts with the initial parsing of the data and might be chained all along in a full routine to output the curated data.

As for example, we can imagine a simple message containing a raw log file with a set of certain events gathered and labelled on different degrees of error -from warning to critical-:

- The parser will understand the format of the log and extract the relevant information -fields- out of it, ignoring the rest of the unnecessary information.
- Decompose: The parser will deep inspect and extract the relevant fields of the filtered information, for example, a timestamp in Linux format and a variety of fields relevant to the issue.
- After the significant fields are extracted, such as alert degree, timestamp and, for example, an IP address, the normalization process, for example, it can turn the timestamp into human-readable date.
- The aggregation process will gather the previous object and will recompose this raw information as an object or entity suitable for the needs of the project. For example, it can label the event as a “Unauthorized access”.

Validation proposal: all the previous steps may be carried out according to a predefined set of rules and schemas previously defined, where the achieved results can be compared against expectations.

Enrich

Next step is “Enrich”, which consists of adding relevant information to the event that was not contained in the initial message, like user information or type of connection information.

To carry out this enrichment, several independent tasks will add different types of information to the message.

Figure 38: Process: Second step in Data Processor

The information used to enrich an event may vary depending on the source of information, whether the event was generated from outside the organization or internally; or depending on the nature of the information, such as a generic cyber-intelligence feed or an ad-hoc network IDS on the self LAN.

- **User:** Depending on the event, it might be possible to attach to it an entity identifying, with a certain degree of confidence, the author of it. It can be a local user of our self- organization, identified with its username or email address, or employee id or simply with the user’s host-name. In other cases, such intelligence feeds, an event might be enriched with an actor id, such an APT name, email address of a spam source, etc.
- **Geo:** Events that are including an IP address are likely to be traced back to its geo location origin. Despite the Ip address or other significant information of a host can be spoofed with the usage of VPNs, the TOR network, etc. it can be valuable information for generating TTPs of a certain type of event. Also, sometimes the events can be labelled not with coordinates but, generally, with a country origin.
- **Whois:** Information related to a domain name that can be used to enrich an event and might include several information about the owner of a host. it might include the address and full name (could be fake) of an author or organization, and also the name servers and contact email of the host.
- **Conn:** Other relevant information regarding a connection, such as the public IP, Internet Service Provider or MAC address.
- **Asset:** After the process of enriching the information and normalization of the previous phase, the output is a manageable, structured resource that can be operated with. A structured object can be included seamlessly in a database (meaning it has attributes with meaning, and typed value) and can be visualized and operated (CRUD operations) within a system.

Label

Labelling is the next and final step of processing raw data and, as a result, labelling can turn the previous, structured and enriched data, into Threat Intelligence. This information will be then shareable and compliant with Cyber Threat Intelligence share standards such as MISP or STIX.

Figure 39: Process: Third step in Data Processor

Labelling is the process that adds meta-information or metadata to an event being feed and categorizes it for better management or treatment of the event, either in the following scenarios:

- Pseudo real-time scenarios where the labelled data is visualized and/or automated software performs pre-programmed actions over the labelled incoming events.
- Digital investigations and forensics: where the findings need to be classified and better organized.
- Threat Intelligence community: where curated, labelled data is classified and shared accordingly.

Part of the enriching process could have done this job partially already; but in any case, it is relevant to remark the following categories:

- **TLP:** The Traffic Light Protocol is a scheme created to promote a better exchange of sensitive (but unclassified) information in the field of information security. Through this scheme, in an agile and simple way, the author of an information indicates how far the information can be circulated beyond the immediate recipient, and the latter must consult the original author when the information needs to be distributed to third parties. In this protocol, there are four categories –labels- on which a piece of information can be graded:
  - **TLP:White:** should be used when the information does not pose any risk of misuse, within the rules and procedures established for its public dissemination. TLP:White information may be distributed without restriction, subject to copyright controls.
  - **TLP:Green:** should be used when the information is useful to all participating organizations, as well as to third parties in the community or industry. Recipients may share information listed as **TLP:GREEN** with affiliated organizations or members of the same sector, but never through public channels.
  - **TLP:Amber:** should be used when information requires limited distribution, but poses a risk to privacy, reputation or operations if shared outside the organization. Recipients may share information indicated as **TLP:AMBER** only with members of their own organization who need to know it, and with customers, suppliers or associates who need to know it to protect themselves or avoid harm. The sender may specify additional restrictions on sharing this information.
  - **TLP:RED:** should be used when the information is limited to specific individuals, and could impact privacy, reputation or operations if misused. Recipients should not share information designated as **TLP:RED** with any third party outside the scope where it was originally exposed.
- **TTPs:** The Techniques Tactics and Procedures allows identifying common patterns and behaviors observed in an antagonist. This way, when can model the usual behavior of an actor.
  - A **tactic** is the highest-level description of this behaviour.
  - **Techniques** give a more detailed description of behaviour in the context of a tactic. Keeping with the example: “Uses ad-hoc malware covert on office documents resembling those of the legit company.”
  - **Procedures** an even lower-level, highly detailed description in the context of a technique.

For example, when enough significance is gained, a malicious APT could include, as part of the TTP list, a statement such as “Spreads certain malware by means of spear phishing and other social engineering strategies, like phishing websites.”, “The malware is delivered via a cover office document resembling a legit one“, “The malware consists of an ad-hoc development in a delivery process of two stages, using a dropper written in PowerShell that downloads a ransomware from a set of contacted domains.”

Analysis of TTPs aids in counterintelligence and cybersecurity operations by articulating how threat actors perform attacks. Actions related to TTP maturation include, but are not limited to:

- Rapid triage and contextualization of an event or incident by correlating it to TTPs of known actors or groups potentially related to an attack.
- Supporting the investigative process by providing probable paths for research and focus, based upon former TTPs used in a campaign or attack.
- Supporting identification of possible sources or vectors of attack.
- Supporting the incident response and threat identification and mitigation processes by helping identify which systems are likely to be compromised.
- Supporting threat modelling exercises by assisting with controls analysis and integration to defend against known threat actor TTPs.

- **IOC:** The **Indicators** of Compromise are, in digital forensics and threat Intelligence, artefacts observed on a network or system that, with high confidence, indicates an attack or intrusion.

  The intent of an indicator of compromise is to outline the information that is received or extracted during the analysis of an incident in such a way that it can be reused by other investigators or affected parties to discover the same evidence on their systems to determine whether or not they have been compromised either from a threat monitoring or forensic analysis standpoint. For example, IP addresses observed, domain names contacted, hash signatures of certain pieces of suspicious malware, files created, registry entries modified, new processes or services are identified, etc. The underlying idea is that if a cybersecurity analyst finds the details contained in a particular IoC within a system, it is likely that is dealing with an infection caused by the malware referred to in that indicator of compromise.

  Indicators of Compromise enable a simple and practical exchange of information for the purpose of intrusion detection from forensic analysis, incident response or malware analysis. For this reason, IoCs are collected, stored and distributed by so-called Threat Intelligence Platforms (TIPs) that enable the management of IT threat intelligence and associated entities such as actors, campaigns, incidents, signatures, bulletins and bulletins and the aforementioned TTPs.

- **STIX:** Structured Threat Information Expression (STIX). JSON structured language for describing cyber threats in a format that can be shared, stored and analyzed consistently. It provides an organization of information in a highly structured and interrelated manner for high readability and easy understanding. STIX allows representing the information in a visual, graph-based format to provide a very intuitive representation of objects and relationships between objects. This format might be very useful for high-level aspects such as who is behind the malware and where it has been used. It allows to take advantage of the information described in other formats, for example OpenIOC, Snort rules (network intrusion detection system) and YARA rules (identify malware based on text or binary patterns and the use of Boolean expressions to determine their logic) among others.
- **CVSS:** The Common Vulnerability Scoring System (CVSS) provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity. The numerical score can then be translated into a qualitative representation (such as low, medium, high, and critical) to help organizations properly assess and prioritize their vulnerability management processes. In other words, is an open and universally used framework

that establishes metrics for communicating the characteristics, impact and severity of vulnerabilities affecting elements of the IT security environment. The responsible organization, the Forum of Incident Response and Security Teams (FIRST) has presented an update of the standard with the CVSS 3.0 version.

A CVSS score is composed of three main groups of metrics: Baseline, Temporal and Environmental. Each of these groups is in turn composed of a set of metrics.

- **Base Group:** It encompasses the intrinsic qualities of a vulnerability that are independent of time and environment. The metrics evaluated in this group are:
  - Access Vector (AV). Values: [L,A,N] (Local, Adjacent, Network)
  - Access Complexity (AC). Values [H,M,L] (High, Medium, Low)
  - Authentication (Au). Values [M,S,N] (Multiple, Single, None)
  - Confidentiality Impact (C). Values [N,P,C] (None, Partial, Complete)
  - Integrity Impact (I). Values [N,P,C]
  - Availability Impact (A). Values [N,P,C]
- **Temporal Group:** Vulnerability characteristics that change over time. Three metrics are applied:
  - Exploitability (E). Values: [U,POC,F,H,ND] (Unproven, Proof-of-Concept, Functional Exploit, High, Not Defined).
  - Remediation Level (RL). Values: [OF,TF,W,U,ND] (Official Fix, Temporary Fix, Workaround, Unavailable, Not Defined).
  - Report Confidence (RC). Values: [UC,UR,C,ND] (Unconfirmed, Uncorroborated, Confirmed, Not Defined).
- **Environmental Group:** The characteristics of the vulnerability related to the user's environment. In this case the factors evaluated are:
  - Collateral Damage Potential (CDP). Values: [N,L,LM,MH,H,ND] (None, Low, Low Medium, Medium High, High, Not Defined).
  - Target Distribution (TD). Values: [N,L,M,H,H,ND]
  - Security Requirements – CR, IR, AR (Confidentiality Requirements, Integrity Requirements, Availability Requirements). Values: [L,M,H,ND]

Once the values of each metric have been assigned, the formulas (equations) contained in the CVSS specifications will be applied and will result in a numerical value between 0.0 and 10.0 for each group. This total numerical result scores and quantitatively determines the final impact of a vulnerability. The final numerical value is accompanied by a text string, called a vector where the syntax (metric:[value]) (each group of metrics evaluated) is specified.

The base metric is mandatory and optionally the temporal and environmental factors can be evaluated.

A typical CVSS example looks as follows:

Heartbleed CVE-2014-0160

CVSS v2 Base Score: 5.0 (AV:N/AC:L/Au:N/C:P/I:N/A:N)

- **AV:N** --> Access Vector: Network
- **AC:L** --> Access Complexity: Low
- **AU:N** --> Authentication: None
- **C:P** --> Confidentiality: Partial
- **I:N** --> Integrity:None
- **A:N** --> Availability:None

5.1.3 Data Storage

Seen in the previous “Data Processor” section the process of collection and processing of information, in this section we will review what are the main methods that we will have for the storage of information that has previously been collected and processed.

Figure 40: Data Storage in Prediction Framework

In most cases we will have to face regulatory frameworks that will condition the system of retention and presentation of this information. This implies that during the design phase of the desired architecture until the implementation phase, we must take into account regulations such as GDPR (General Data Protection Regulation), PCI/DSS (Payment Card Industry Data Security Standard), HIPAA (Health Insurance Portability and Accountability Act) and SOX (Sarbanes Oxley) that require that the information we are going to treat must be stored between 1 and 7 years.

For example, PCI/DSS requires to have the logs available for at least one year, and the last 3 months must be easily accessible and removable, that is, no tapes or cabin systems. Large corporations have a considerable flow of logs and events being stored in real time.

A large company installation can easily exceed 20.000 events per second on average (EPS). Taking into account that it is an average and there will be peaks of more than 40.000 EPS and valleys where we barely reach 5.000 EPS.

It is estimated that 1 event supposes on average about 300 bytes, which means 0,006 GB per second, being a total of 500 GB per day of storage of necessary information.

Therefore, the greater the volume of information intake, the greater the costs associated with:

- Licensing of the software necessary for the treatment of events.
- Hardware needed to carry out the processing.
- Hardware and software required for storage.

Once we are clear that among the factors to take into account for the storage of information, we have the regulations that could affect us and another of the elements to take into account is the estimation of how many EPS/GB we will need during the definition of the architecture and the implementation of the solution, refining the sources so that we only collect the events that we really need to be processed first and later stored and with it:

- Have an assumable cost for the storage of events.
- To be able to build indexed on the data.
- Make it easily accessible.

In order for the stored information (logs) to be exported, if necessary **(accessible)** and easily consultable **(indexed)**, without having to allocate a large number of resources (hardware/disks), it is necessary that the above characteristics are met.

We have already referred to the **Volume of Data** than an organization can generate, but we must also attend to the **Variety of data** sources that usually any organization must handle and subsequently save for treatment. Among these data sources the most common are the following:

- **Structured data**, which are most of the traditional data, with schema or fixed format, from relational databases, spreadsheets or files with a fixed format.

- **Semi-structured data**, fixed raw data, which contains tags or markets that allow elements to be formatted or separated, such as Web logs, logs of internet connections or any other text tagged with XML or HTML.
- **Unstructured data**, those that do not have any predefined type. They are stored in objects or documents without any structure such as text data, photographs, videos or audios.

As reflected above, enterprise systems store more and more heterogeneous data. This implies that storing this data in a single type of warehouse is not the best option. Typically, different types of data are stored in different stores, each specialized or oriented to a specific workload use or pattern. The term “Polyglot Persistence” [209] it is used to describe solutions that employ a combination of data storage technologies, so it is necessary to take into account the main storage models and their disadvantages, to choose the most appropriate architecture for our needs. There are, for example, implementations between SQL and NoSQL databases or data stores that are classified according to the way the data is structured or the types of operations they support. Not all data stores in a given category provide the same feature set.

Most data stores provide server-side functions for querying and processing data. Sometimes these functions are built into the data storage engine. In other cases, data storage and processing functionalities are separated and there may be several options for processing and analysis. Datastores also support different programming and management interfaces.

The first thing we must take into account is which storage model best suits our requirements, taking into account among other factors such as the set of features, cost and ease of administration to opt for a given data warehouse.

Here are some of the most common data storage models:

Relational database management systems organize data as a series of two-dimensional tables with rows and columns. Most providers provide the dialect of Structured Query Language (SQL) to retrieve and manage data. Typically, a relational database management system implements a transactionally consistent mechanism that conforms to the ACID (atomicity, consistency, isolation, durability) model to update information.

An RDBMS typically supports a schema-on-write model, where the data structure is defined ahead of time, and all read or write operations must use the schema.

This model is very useful when strong consistency guarantees are important where all changes are atomic, and transactions always leave the data in a consistent state. However, an RDBMS generally can’t scale out horizontally without sharing the data in some way. Also, the data in an RDBMS must be normalized, which isn’t appropriate for every data set.

- **Workload**
  - Records are frequently created and updated.
  - Multiple operations have to be completed in a single transaction.
  - Relationships are enforced using database constraints.
  - Indexes are used to optimize query performance.
- **Data Type**
  - Data is highly normalized.
  - Database schemas are required and enforced.
  - Many-to-many relationships between data entities in the database.
  - Constraints are defined in the schema and imposed on any data in the database.
  - Data requires high integrity. Indexes and relationships need to be maintained accurately.
  - Data requires strong consistency. Transactions operate in a way that ensures all data are 100% consistent for all users and processes.
  - Size of individual data entries is small to medium-sized.
- **Examples**
  - Inventory management.

- Order management.
- Reporting database.
- Accounting.

Time series databases is a set of values organized by time. Time series databases typically collect large amounts of data in real time from many sources. Updates are rare and deletes are often done as bulk operations. Although the records written to a time-series database are generally small, there are often a large number of records, and total data size can grow rapidly.

- **Workload**
  - Records are generally appended sequentially in time order.
  - An overwhelming proportion of operations (95%-99%) are writes.
  - Updates are rare.
  - Deletes occur in bulk and are made to contiguous blocks or records.
  - Data is read sequentially in either ascending or descending time order, often in parallel.
- **Data type**
  - A timestamp is used as the primary key and sorting mechanism.
  - Tags may define additional information about the type, origin, and other information about the entry.
- **Examples**
  - Monitoring and event telemetry.
  - Sensor or other IoT data.

A **key/value store** associates each data value with a unique key. Most key/value stores only support simple query, insert, and delete operations. To modify a value (either partially or completely), an application must overwrite the existing data for the entire value. In most implementations, reading or writing a single value is an atomic operation.

An application can store arbitrary data as a set of values. Any schema information must be provided by the application. The key/value store simply retrieves or stores the value by key.

Key/value stores are highly optimized for applications performing simple lookups, but they are less suitable if you need to query data across different key/value stores. Key/value stores are also not optimized for querying by value.

A single key/value store can be extremely scalable, as the data store can easily distribute data across multiple nodes on separate machines.

- **Workload**
  - Data is accessed using a single key, like a dictionary.
  - No joins, lock, or unions are required.
  - No aggregation mechanisms are used.
  - Secondary indexes are generally not used.
- **Data type**
  - Each key is associated with a single value.
  - There is no schema enforcement.
  - No relationships between entities.
- **Examples**
  - Data caching
  - Session management
  - User preference and profile management
  - Product recommendation and ad serving

A **Document Database Stores** a collection of documents, where each document consists of named fields and data. The data can be simple values or complex elements such as lists and child collections. Documents are retrieved by unique keys.

Typically, a document contains the data for single entity, such as a customer or an order. A document may contain information that would be spread across several relational tables in an RDBMS. Documents don't need to have the same structure. Applications can store different data in documents as business requirements change.

- **Workload**
  - Insert and update operations are common.
  - No object- relational impedance mismatch. Documents can better match the object structures used in application code.
  - Individual documents are retrieved and written as a single block.
  - Data requires index on multiple fields.
- **Data type**
  - Data can be managed in de-normalized way.
  - Size of individual document data is relatively small.
  - Each document type can use its own schema.
  - Documents can include optional fields.
  - Document data is semi-structured, meaning that data types of each field are not strictly defined.
- **Examples**
  - Product catalog
  - Content management
  - Inventory management

A **Graph Database stores** two types of information, nodes and edges. Edges specify relationships between nodes. Nodes and edges can have properties that provide information about that node or edge, similar to columns in a table. Edges can also have a direction indicating the nature of the relationship.

Graph databases can efficiently perform queries across the network of nodes and edges and analyze the relationships between entities. The following diagram shows an organization's personnel database structured as a graph. The entities are employees and departments, and the edges indicate reporting relationships and the departments in which employees work.

For large graphs with lots of entities and relationships, you can perform very complex analyses very quickly. Many graph databases provide a query language that you can use to traverse a network of relationships efficiently.

- **Workload**
  - Complex relationships between data items involving many hops between related data items.
  - The relationship between data item is dynamic and change over time.
  - Relationships between objects are first-class citizens, without requiring foreign-keys and joins to traverse.
- **Data type**
  - Nodes and relationships.
  - Nodes are similar to table rows or JSON documents.
  - Relationships are just as important as nodes and are exposed directly in the query language.
  - Composite objects, such as a person with multiple phone numbers, tend to be broken into separate, smaller nodes, combined with traversable relationships.
- **Examples**
  - Organization charts.
  - Social graphs.
  - Fraud detection.
  - Recommendation engines.

Data analytics stores provide massively parallel solutions for ingesting, storing, and analyzing data. The data is distributed across multiple servers to maximize scalability. Large data file formats such as delimiter files (CSV), parquet, and ORC are widely used in data analytics.

- **Workload**
  - Data analytics.
  - Enterprise BI.
- **Data type**
  - Historical data from multiple sources.
  - Usually denormalized in a “star” or “snowflake” schema, consisting of fact and dimension tables.
  - Usually loaded with new data on a scheduled basis.
  - Dimension tables often include multiple historic versions of an entity, referred to as a slowly changing dimension.
- **Examples**
  - Enterprise data warehouse

Column-Family Databases organizes data into rows and columns. In its simplest form, a column-family database can appear very similar to a relational database, at least conceptually. The real power of a column-family database lies in its denormalized approach to structuring sparse data.

You can think of a column-family database as holding tabular data with rows and columns, but the columns are divided into groups known as column families. Each column family holds a set of columns that are logically related together and are typically retrieved or manipulated as a unit. Other data that is accessed separately can be stored in separate column families. Within a column family, new columns can be added dynamically, and rows can be sparse (that is, a row doesn’t need to have a value for every column).

Unlike a key/value store or a document database, most column-family databases store data in key order, rather than by computing a hash. Many implementations allow you to create indexes over specific columns in a column-family. Indexes let your retrieve data by columns value, rather than row key.

Read and write operations for a row are usually atomic with a single column-family, although some implementations provide atomicity across the entire row, spanning multiple column-families.

- **Workload**
  - Most column-family databases perform write operations extremely quickly.
  - Update and delete operations are rare.
  - Designed to provide high throughput and low-latency access.
  - Supports easy query access to a particular set of fields within a much larger record.
  - Massively scalable.
- **Data type**
  - Data is stored in tables consisting of a key column and one or more column families.
  - Specific columns can vary by individual rows.
  - Individual cells are accessed via get and put commands.
  - Multiple rows are returned using a scan command.
- **Examples**
  - Recommendations.
  - Personalization.
  - Sensor data.
  - Telemetry.
  - Messaging.
  - Social media analytics.
  - Web analytics.
  - Activity monitoring.

- Weather and other time-series data.

A **Search Engine Database** allows applications to search for information held in external data stores. A search engine database can index massive volumes of data and provide near real-time access to these indexes.

Indexes can be exact or fuzzy. A fuzzy search finds documents that match a set of terms and calculates how closely they match. Some search engines also support linguistic analysis that can return matches based on synonyms, genre expansions, and stemming (matching words with the same root).

- **Workload**
  - Data indexes from multiple sources and services.
  - Queries are ad-hoc and can be complex.
  - Full text search is required.
- **Data type**
  - Semi-structured or unstructured text.
  - Text with reference to structured data.
- **Examples**
  - Products catalogs.
  - Site search.
  - Logging.

Object storage is optimized for storing and retrieving large binary objects (images, files, video and audio streams, large application data objects and documents, virtual machine dis images). Large data files are also popularly used in this model, for example, delimiter file (CSV), parquet, and ORC. Object stores can manage extremely large amounts of unstructured data.

- **Workload**
  - Identified by key.
  - Content is typically an asset such as a delimiter, image, or video file.
  - Content must be durable and external to any application tier.
- **Data type**
  - Data size is large.
  - Value is opaque.
- **Examples**
  - Images, videos, office documents, PDFs.
  - Static HTML, JSON, CSS.
  - Log and audit files.
  - Database backups.

Shared files. Sometimes, using simple flat files can be the most effective means of storing and retrieving information. Using file shares enables files to be accessed across a network. Given appropriate security and concurrent access control mechanisms, sharing data in this way can enable distributed services to provide highly scalable data access for performing basic, low-level operations such as simple read and write requests.

- **Workload**
  - Migration from existing apps that interact with the file system.
  - Requires SMB interface.
- **Data type**
  - Files in a hierarchical set of folders.
  - Accessible with standard I/O libraries.
- **Examples**
  - Legacy files.
  - Shared content accessible among a number of VMs or app instances.

To comply with this type of characteristics we will have three possible alternatives “Database”, “Warehouse” and “Data Lake”, although we could use SIEM’s own technologies for storage, if there were a deployed SIEM solution in the architecture in question.

We can also distinguish between the storage options available by their location, that is, by their physical distance from the data center. These will be differentiated between hot and cold, the elements closest to the Data Center and those that are accessed most regularly were located, literally, in storage facilities that were hot. Items farther away from the Data Center had slower load times, so it became the place to store the data you needed to access much less frequently. This type of storage was done differently than hot storage, usually using old drives or units that were off most of the time. These types of storage did not generate the heat generated by the nearest storage facilities.

Here’s a breakdown of the traditional hot vs cold categories, with warm data storage emerging as a third type:

	Hot Storage	Warm Storage	Cold Storage
Location	Close to the moment of computation	On a remote server or private network, usually a step away from users	In Cloud Services
Defining Characteristic	Very fast access	Medium speeds	Different pricing tiers for hot and cold options
Examples	Personal hard drives, SSDs, flash drives	Larger, cheaper, spinning drives	Cloud drives, AWS, Google Cloud Storage

Table 29: Traditional hot vs cold categories

Database Technology, it is a storage location that houses structured data. We usually think of a database on a computer-holding data, easily accessible in a number of ways. For organizations, the cases for databases include creating reports for financial and other data, analyzing relatively small datasets, automating business processes, auditing data entry. Most popular databases are:

- Oracle.
- PostgreSQL.
- MongoDB.
- Redis.
- Elasticsearch.
- Apache Cassandra.

Data Warehouse Technology, Data Warehouses are large storage locations for data that you accumulate from a wide range of sources. For decades, the foundation for business intelligence and data discovery/storage rested on data warehouses. Their specific, static structures dictate what data analysis you could perform. Data warehouses are popular with mid- and large-size businesses as a way of sharing data and content across the team- or department-siloed databases. Data warehouses help organizations become more efficient. Organizations that use data warehouses often do so to guide management decisions-all those “data-driven” decisions you always hear about. Popular companies that offer data warehouses include:

- Snowflake.
- YellowBrick.
- Teradata.

Data Lake Technology, it is a technology that allows us to store all types of data, from structures and formatted, to unstructured data an unrelated to each other. In a “Data Lake” the data is not sorted or indexed, it is simply stored and left accessible for consultation and extraction. They are usually presented as **Cloud environments** which allows us to easily adjust to the needs of the organization.

Key benefits & drawbacks of data storage types:

	Database	Data Warehouse	Data Lake
Data	Structures	Structured	Raw & unstructured
Processing	Schema-on-write	Schema-on-write	Schema-on-read
Cost	Free to €	€€€	€
Agility	Varies	Minimal	Maximum
Security	Immature	Mature	Immature
Users	Anyone	IT/business users	Data Scientists
Use Cases	Reporting, analysis & automation	Machine Learning	Data Science & research

Table 30: Key benefits & drawbacks of data storage types

On the other hand, the default repositories of SIEM technologies (QRadar, Splunk, AlienVault etc.) work in black box mode, that is, the technology or the specific way in which these technologies do it is not available or accessible, since they have their own storage systems, where in exchange for a higher cost in hardware and licenses, and with certain particularities of these technologies, information is stored, processed and structured.

With the advent of Big Data technologies, and with the aim of overcoming the limitations offered by traditional Data Warehouses, many organizations have adopted the Data Lake model as a storage model to solve the challenges in the management of the data obtained, since they offer scalability, speed and profitability necessary to help manage a large volume and type of data in their different analytics options: AI, BI, machine learning, streaming analytics, etc.

Initially there were local Data Lake solutions such as **Hadoop** (Open Source), but these had a series of limitations (elasticity, lack of security and governance, high maintenance costs) that with the arrival of cloud platform providers have made these limitations disappear and even expand even more the possibilities or advantages with the migration of Data Lakes to the cloud.

Among the advantages of migrating Data Lakes to the cloud we can find the following:

- Decoupled storage and computing.
- Built- in security and encryption.
- Transparent scalability.
- Flexible infrastructure on demand.
- Prices according to consumption.

Most providers that offer Data Lakes in the cloud differ in their features, functionalities or capabilities that they consider unique, but most of these are made up of the same key components, only varying in a series of attributes between different providers.

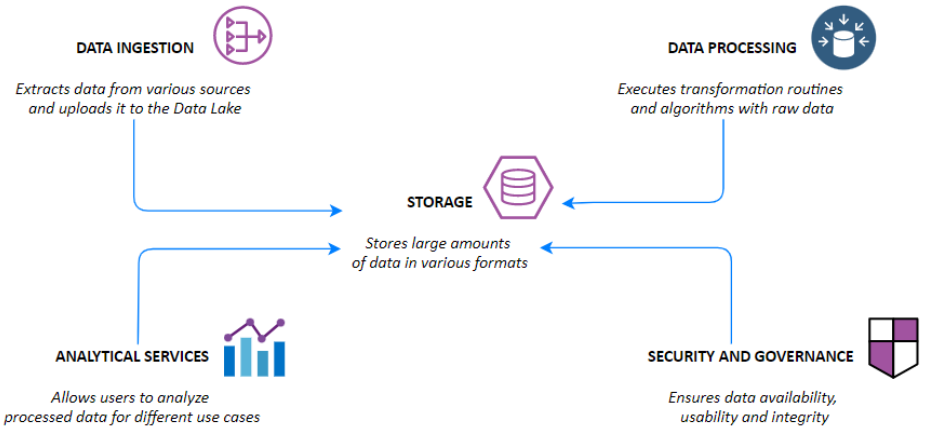

Figure 41: Series of attributes between different providers [210]

Here is a description of some top Cloud Data Lakes providers:

Specifications	aws	Microsoft Azure	Google Cloud Platform	CLOUDERA	databricks	snowflake
Primary storage service	Amazon S3	ADLS Gen 2	Google Cloud Storage	Cloudera Data Platform	Data Lake under AWS, GCS o ADLS	Snowflake cloud data platform
Processing engine	Amazon EMR	Azure HDInsight, Azure Synapse	Google Dataproc, Data Flow	CDP Data Engineering	Delta Engine	Snowflake
Hive and Spark support	yes	yes	yes	yes	Spark	N/A
Decoupled storage and compute	yes	yes	yes	yes	yes	yes
Channel service	AWS Glue	Azure Data Factory	Cloud Data Fusion, Data Flow	Cloudera Data Engineering	Delta Engine	Snowpark
SQL support	Amazon Athena, Redshift, Spectrum	Azure Synapse	Google BigQuery	Self-service analytics services for data warehouse	SQL analytics service	Snowflake
Multi-programming language support	yes	yes	yes	yes	yes	yes
catalog	AWS Glue	Azure Data Catalog	Google Data Catalog	Cloudera Data Platform	Partners Solutions	Partners Solutions
Lakehouse architecture	yes	Yes	yes	yes	yes	yes
Multicloud	no	no	no	yes	yes	yes

Figure 42: Series of attributes between different providers [210]

5.1.4 Data Analyzer

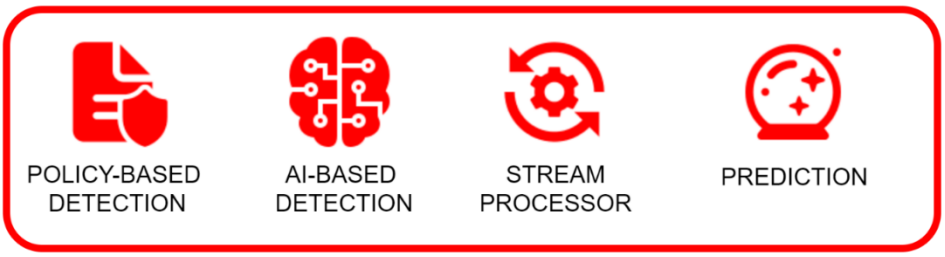

Figure 43: Data Analyzer Process in Prediction Framework

Stream processor

Given the heterogeneous data sources previously established the threat detection system requires continuous data ingestion. As established, the data will be presented in various possible modalities, be it databases, data warehouses or data lakes. Apart from data in the specific format provided by other such as SIEM sources. The stream processor module will be responsible for providing the means for continuous data processing based on available technologies. Through these techniques

it is possible to accelerate the speed at which value is obtained from the data, previously processed, enriched and labelled in the data processing unit and to generate actions to interact with the clients with low latency.

The streaming system will make data from the various sources available for sequential processing from infinite data streams without time limitation. Special attention shall be paid on the requirements based on:

1) Performance: The amount of simultaneous data with previously established peaks that can vary between 5.000 EPS and 40.000 EPS requires a high throughput technology, that is, a high rate of records processed per second. It must also be a scalable technology.
2) Delivery guarantee: The message delivery guarantee mechanisms can follow different policies based on the data ingest tolerances of the models. The delivery guarantee mechanisms can be classified according to their capabilities as: *at-least-once* delivery, which guarantees the availability of the data even in case of failure where duplicities may occur, therefore it is up to the models to discriminate duplicities; *at-most-once* delivery, in order to guarantee that duplicities do not occur some record may not be processed in case of failure, it can be applied in models that require uncorrelated data and have multiplexed inputs to cope with the absence of some of the input features; *exactly-once* delivery, is difficult to achieve in distributed systems and involves a compromise with performance, this option guarantees that it will be processed once and exactly once, even in case of failures, in SPARTA architecture it will be prioritized over the other policies in order to simplify the composition of threat prediction models.
3) Latency times: Shall be as small as possible in order to seek real-time processing, orders of magnitude between 10ms and a second.
4) Fault tolerance: Apart from replication mechanisms that improve the persistence and robustness of the service, the system must be able to re-establish itself in case of node or network-wide crashes by resuming the processing point from the streaming state storage. An example would be the checking of Kafka's offsets in Zookeeper or Samza's checkpoints.
5) Status management: If the technology allows maintaining a status storage that keep an updated record of some property that is required to be recorded, such as in the case of the detection of some anomaly or threat.
6) Advanced features: Some streaming technologies allow advanced features such as application of time windows, watermarking, aggregations, sessions, etc. In the use case involving threat detection one of the most desirable advanced features is the processing of logs based on the time they were generated at the source known as event time processing.

As previously mentioned, event streaming is an advantage given the decoupling between editor and subscriber. The data manager module is the intermediary that can manage the message queues, facilitating multiple configurations. On-demand models can be developed by configuring the data ingestion of each model independently and can be implemented through microservices.

Regarding the available open-source technologies, we have reviewed some candidate technologies to cover the requirements of the threat detection framework and will select one of those described below on the basis of the criteria previously described, the time of maturity and the popularity of the solutions:

Figure 44: Apache Storm Technology

The most established streaming technology by life time, and therefore, maturity. Storm is an open source, distributed, fault-tolerant system for processing real-time data streams with Apache Hadoop. Storm solutions can also provide guaranteed data processing, with the ability to replay data that has not been processed correctly the first time and multi-language.

Figure 45: Kafka Streams Technology

Kafka Streams is a stream processing library on top of Apache Kafka. Kafka Stream is a lightweight Java library that, in comparison with the rest of technologies contemplated on this comparison, cannot be considered as an standalone solution but an API over Kafka topics. It has exactly-once delivery guarantee methods, simple message reprocessing and microservices orientation.

Figure 46: Spark Streaming Technology

Spark is postulated as the successor to Hadoop in batch processing. Spark Streaming extends the functionality of Apache Spark to perform stream processing. It does this by grouping incoming data into small micro-batches and includes stateful and stateless operators. One of the advantages of the Spark ecosystem is that it is highly customized for analysis and predictive model generation through libraries such as MLlib.

Figure 47: Flink Technology

Emerged in 2011 and conceived as the successor of Storm, it is a distributed stream processing platform with high availability and scalability. Although it can perform batch and streaming processing, it has been designed with an architecture oriented to the processing of individual streaming events. As in the case of Spark streaming, Flink is compatible with Lambda architecture.

The following table highlights some of the most important features of each technology described above. These characteristics have been chosen because they are the ones, we think are interesting for the proposed technological solution. In addition, it is important study the answer of the system against unexpected failures, as well as the compatibility of the tools so they provide more flexibility to the system.

	STORM	SPARK STREAMING	KAFKA STREAMS	FLINK
Deployment	Standalone	Standalone, Yarn, Mesos	Java API library	Standalone, Yarn, Mesos, Kubernetes
Delivery	At least Once	Exactly once	Exactly once	Exactly once
Connectors	Multiple	Multiple	Kafka depending connectors	Multiple
Fault tolerant	Yes	Yes	Yes	Yes
State Management	No	No	Yes (RocksDb)	Yes (RocksDb)
Event delay	Yes	Yes	Yes	Yes
Stream type	Native	Micro batches	Native	Native
Batch support	No	Yes	No	Yes
Advanced features	No	Stream joins event time processing	Stream joins	Stream joins event time processing

Table 31: Comparative of most important features of each technology for stream processor

Policy-based detection

The only way to deal with today's security threats is to address them holistically throughout the entire attack cycle: before, during and after the attack. Some threat detection methods rely on running files in a sandbox for detection and analysis, the use of virtual emulation layers to confuse malware, and the use of reputation-based whitelisting to distinguish acceptable applications from malicious ones. More recently, attack chain simulation and analysis detection have also been included. All these dimensions can be managed by individuals from high-level abstractions to high-level abstractions from policies. In this way, by collecting telemetry data from the process level, it is possible to detect events in real time. Analysis can be carried out at different levels. Analysis involves more than just event enumeration and correlation, so telemetry data is connected to provide a broader view of what is happening in the entire managed environment from the Data Manager module.

AI-based detection

The inference models will have a preprocessing pipeline to ingest the data through the streaming processing system and form the most appropriate input. In this process, if necessary, based on the predictive model, the data preprocessing will be formed from a preparation and validation internal data pipeline which will allow data transformation, feature engineering, embedding, etc., according to the requirements of each model provided in an internal model zoo. New models could be integrated into the framework after validation in order to ensure that every model integrated into the detection framework copes with the required characteristics in order to guarantee explainability, correct definition of input data pipeline from the Data Processor outcomes, model registry and parameters storage in the Data Storage Module, metaparameters and performance metrics in order

to monitor concept drift, etc. Multiple agents will be in charge of providing all classified events and present them in reports that will be suitable to support decision making process.

Prediction

The outputs from of policy-based and AI-based detections will be provided in a time series format. The results will be presented in a time frame that will aggregate all the results and present them in clear representations that can be composited into multiple views from selective filters.

This presentation tool will have different visualizations oriented to facilitate the exploratory analysis of the detected events and the predictive analysis of the presented time series, being able to decompose them into their different components based on the Cleveland STL decomposition:

- **Trend**: It is the increasing/decreasing behavior of the series over time, this relationship does not have to be linear.
- **Seasonal**: These are components that occur when there is seasonality, i.e., parameters that occur with a certain frequency.
- **Cyclical**: These are fluctuations that do not always occur at a certain frequency; it is the least obvious component to present in the decomposition.

Within the exploratory analysis tools, there will be views oriented to common data transformations, for example, application of different moving averages of variable windows, median filter, logarithmic, frequency domain transformations (Fast Fourier Transform, Wavelets, etc). It will also provide descriptive statistics and interfaces for the approach of the most common hypotheses to be performed in the study of time series, e.g., ANOVA or Dickey-Fuller test for the study of stationarity.

In addition, companies and organizations heavily rely on this data for critical tasks such as predictions and decision making based on them for countermeasure proposals. While all these tasks generally focus on actual data, it is also useful to apply them to alternative scenarios to prepare for developments that differ from expectations or to assess the robustness of current strategies. Tools such as TensorFlow's What If [211] will be integrated for the presentation of various scenarios.

The prediction module will be based on data science frameworks such as those mentioned above in section 3.6. From them, a suite of classical and advanced models will be presented for the generation and training of predictive models that can be registered in the data storage module for the generation of experiments and their comparison with other registered models.

Prediction models will be available within the following categories:

- Econometric models:

  Exponential Smoothing, Integrated Autoregressive Moving Average Models (ARMA, ARIMA, SARIMA...), Dynamic Linear Models.

- Deep Learning Models:

  Multilayer Perceptron, vanilla Recursive Neural Networks, Convolutional Neural Networks, Long-Shot Term Memory, Gated Recurrent Units and hybrid architectures.

From this data fusion, export and registration tools will be made available to the user for the generation of new datasets based on data and artefact versioning mechanisms (DVC, Feast, etc.). Model lifecycle will be managed within the platform and the outcomes will be reported to the Data Manager module, in a suggested scheme similar to that displayed in Figure 48.

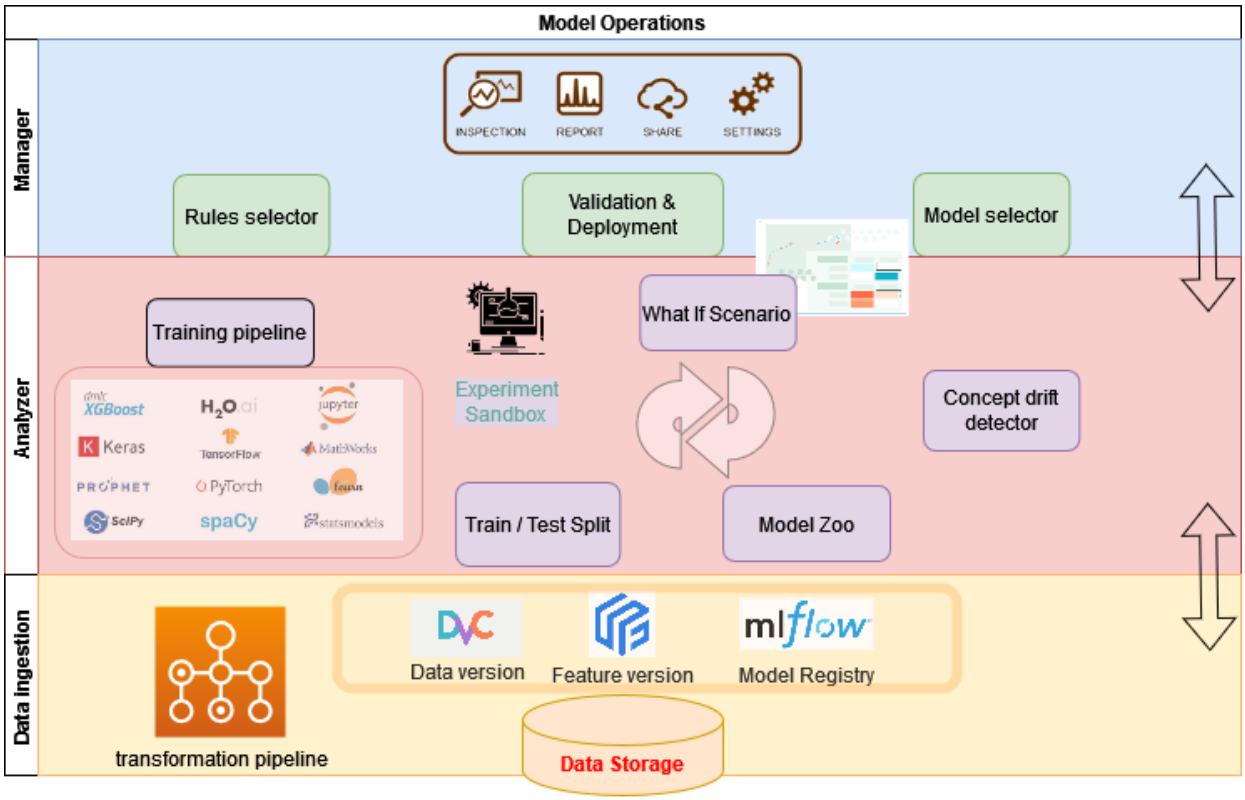

Figure 48: Data analyzer model lifecycle.

5.1.5 Data Manager

With the explosion of web and cloud technologies, databases have evolved from traditional relational databases to more advanced types of databases such as NoSQL, columnar, key-value, hierarchical, distributed databases, etc. Each type can handle structured, semi-structured, and even unstructured data. On top of that, databases are continuously handling mission-critical and sensitive data. When this is coupled with compliance requirements and the distributed nature of most data sets, managing databases has become highly complex. As a result, organizations require robust, secure, and user-friendly tools to work and maintain these databases.

At its most basic level, Data Management (DM) describes the process of collecting, storing, and using data efficiently, securely, and cost-effectively. The primary objective here is to connect and pipe in data from different sources and make critical business decisions. Effective Data Management is a combination of best practices, concepts, processes, procedures, and an extensive collection of tools that help enterprises control and manage their data resources effectively. In other words, it's a multiplatform heterogeneous process that involves various tools and objectives to achieve centralized data coherence. With these premises, it can be noted that the "Data Manager" module aims to provide the framework with data management, information management, and generated intelligence management. Moreover, all these capabilities will be implemented by the different subtasks of the module: Inspection, Report, Share and Settings.

Figure 49: Data Manager Process in Prediction Framework

Inspection

The main objective of the Situation Awareness prediction framework is to provide the system user with the appropriate information to increase the accuracy and efficiency of the decision making process. The way in which the system will present this information will be mainly through the automatic reports described in the next point of this document. However, there is another way to obtain specific information manually: in the event that the user wants to access certain specific information (whether or not it is contained in the automatic reports), he can make use of a customized request system to the Data Storage module through the Data Manager component, which will present the data obtained in response to his request from the databases. This process will be transparent to the user, who will make the requests and visualize the results in the interface that the Data Manager component makes available for this purpose. This interface will be in charge of the necessary communication with the other components of the system, the processing and presentation of the requested data. In this way, the user is freed from all these tasks, giving him the possibility of obtaining the requested information in a simple way, regardless of the type of database consulted, its structure or the methods and language with which each of the requests is performed.

To perform customized information requests, the user can use different tools that will allow she/him to perform intelligent searches and filter the information based on different tags or characteristics associated with the data in the Data Processor component. By using these tools, it will be possible to perform searches, such as, for example, threats with CVSS greater than 6 and/or matching a certain IoC, which could be a specific IP address or URL. This functionality provided by the Data Manager component increases both the level of detail with which the data is presented and the user's ability to effectively explore the information collected by the Situation Awareness prediction framework.

Report

Management reports are a source of business intelligence that helps companies make more accurate, data-driven decisions. However, these reports are as useful as the work that goes into preparing and presenting them. These reports aim to inform different aspects of the business, to help, the final recipients of these reports, to make better informed decisions. They collect data from various pre-defined sources and present it, allowing decision makers to take the right path to a given incident or situation, thereby increasing operational efficiency.

In order to serve a joint cyber situation prediction approach, we are going to analyze the different characteristics that a report generated from the “Data Manager” must have for the report of events of a SOC before the managers of the company.

First of all, we will have to see the different ways of presenting the reports that we can obtain through the "Data Manager". These will be determined by the type, size, and nature of the data:

- Written Reports.
    - Control Reports.
    - Summary Control Reports.
    - Tables.
    - Ratios.
    - Etc.
- Visual Reports.
    - Graphs.
    - Pie Charts.
    - Stats.
    - Bar Diagrams.

A monthly or quarterly report is a great way to summarize a SOC's performance and uncover ideas for executive leadership*. But what kind of information should these reports give us?*

There are often significant gaps between what the SOC knows and what it informs decision makers. According to EY, only 15% of organizations say their information security reports fully meet their expectations, and only 17% report areas for improvement.

The SOC relies heavily on executives' acceptance of these reports. It is vital that the SOC communicates effectively so that the most important (and most accessible) information can be passed through the chain of command and significant changes can be made. In addition to information relevant to your organization's specific concerns, an effective SOC report will contain the following sections:

Key Findings

The report's most critical findings and action items should be summarized in non-technical language that executives and Board members can understand. Key findings should also include at-a-glance insights into the organization's security performance with clear metrics, such as security ratings. This information should be provided at the beginning of the report, where it is most likely to be viewed and read carefully.

Monitoring Summary

This summary should provide an overview of what was monitored for the report, including the number and locations of servers, workstations, monitored devices, and so on. It should not be neglected to request information about what was not monitored – it is important to identify gaps in the SOC's field of vision, so that strategies can be implemented to close those gaps.

Incident Summary

Here the administrators of the "Data Manager" must provide the total number of incidents detected and resolved, as well as more specific data, such as:

- Breakdown of incidents by type, objective and severity
- Mean time of detection (MTTD)
- Average time to resolution (MTTR)
- Specific actions taken for each incident, such as log collection, quarantine, installation of security patches and password reset or other changes to the authentication system.

Threat Summary

This field should describe the most serious threats facing the company in the last month or quarter, specify whether or not the organization anticipated them, and detail how they were addressed by the SOC. Information about emerging malware trends and recommended actions to prepare for these threats will also be helpful. The threat brief is also where cybersecurity concerns need to be put into context. The SOC administrator needs to present information about common cyber-attacks, using real incidents as examples. As part of the threat summary, the following questions:

- What incidents have occurred recently in our industry?
- What type of threat(s) will pose the greatest risk to our organization in the coming month/quarter?
- How does our organization compare to its peers and competitors when it comes to mitigating risk?

Recommendations

The SOC can take advantage of the opportunity provided by the generation of these reports to advocate and request any additional resources that are necessary to improve performance. Providing concrete recommendations (and if possible, estimated costs), will make the job easier as decisions are made. These recommendations do not have to be entirely within the scope of the SOC. By generating these reports, we can consider how other departments can work together to promote a culture of cybersecurity awareness within the organization.

These recommendations do not have to be entirely within the scope of the SOC. By generating these reports, we can consider how other departments can work together to promote a culture of

cybersecurity awareness within the organization. For example, a large percentage of malware enters organizations through phishing emails, a problem that requires employee training to correct. An SOC could recommend stricter enforcement of cybersecurity policies across all cybersecurity departments or workshops organized by learning and development to address this issue.

Additional Considerations

As each section of the report is composed, you must consider the audience and purpose so that senior-level managers, executives, and Board members can turn data into action. To achieve maximum impact, reports must take into account both clarity and context.

- **Clarity;** most executives and Board members will have limited technical understanding, so clarity is key. An effective report will use language that non-technical people can understand and will make use of synthesized metrics such as safety ratings to deliver complicated information in an easily digestible format. Whenever possible, non-critical information should be in the appendix, so as not to obstruct the body of the report with an excess of data that would make it more difficult to understand.
- **Context;** it is not effective to simply present data in a vacuum. To effectively communicate findings, context and analysis are required. The report should compare cybersecurity KPIs with historical performance, the performance of their peers and competitors, and progress toward set goals. You should also try to provide meaningful analysis: what does it mean to the company that these incidents occurred or could occur in the future? Which incidents pose the greatest risk to revenue, customer trust, and legal costs? When a report successfully illustrates the tangible impact of actual and potential attacks, security leadership can present a more compelling case for allocating more resources to the SOC.

Share

With all other modules, the prediction framework is able to work on its own. However, one of the most important things when it works with information or intelligence is being able to share it to provide other systems or departments with complementary information, to better perform their duties. One of the tasks of the “Data Manager” module is precisely to share information and threat intelligence with other modules and tools. The “Share” task will be in charge of taking the information and sharing it.

Threat intelligence is information, descriptors, and indicators about a specific threat, and it is necessary to share this type of information in a structured and secure way. For this reason, it is very important have a standard structure to represent all the threat information, and a standard way to send this intelligence. “Share” task implements few ways to make it possible, in order to be able to adapt to different environments. Hereby, the framework can work with STIX and SCAP as a representation standard, and TAXII and MISP as transport standards.

STIX

STIX (Structured Threat Information eXpression) is a standard language to describe all about a threat information, it was created by MITRE and now is maintained by OASIS. This standard was thought to represent security incidents, threat indicators and contextual information like threat actors, tactics, techniques and procedures (TTP), targets, etc. See Section 3.1 for more information about STIX.

SCAP

SCAP (Security Content Automation protocol) is a suit of specifications form NIST, to describe all information needed about vulnerabilities, threats, errors, etc. This suit includes all of these specifications:

- CVE: Common Vulnerabilities Exposure is a list of codes to identify al well-known vulnerabilities.

- CPE: Common Platform Enumeration is an enumeration code to identify the platforms and systems.
- XCCDF: eXtensible Configuration Checklist Description Format is a XML format to prepare security documents.
- OVAL: Open Vulnerability and Assessment Language is a language to represent security information and assessment information from the systems.
- CVSS: Common Vulnerability Scoring System is a scoring system to represent the system vulnerabilities.
- OCIL: Open Checklist Interactive Language is a language to represent all data information stored.
- AI: Asset Identification is a format to represent assets information.
- ARF: Asset Reporting Format is a format to represent relations between assets.
- CCSS: Common Configuration Scoring System is a scoring system to represent the value of the systems configuration.
- TMSAD: Trust Model for Security Automation Data is a specification to use digital signatures.

TAXII

TAXII (Trusted Automated eXchange of Intelligence Information) is a protocol that defines the way to share threat information. See Section 3.2 Cyber Threat Intelligence Platform (CTIP) for more information.

MISP

MISP (Malware Information Sharing Platform) is a threat intelligence platform created to share, store and correlate Indicators of Compromise (IoCs). MISP is the most widely used tool for sharing IOCs (Indicators of Compromise), developed by Computer Incident Response Center Luxembourg (CIRCL), the Belgian defense team and NATO (NIRC). MISP's objective is to encourage the structured exchange of information between companies. The MISP platform not only has the necessary elements to facilitate the exchange of information, but it also has mechanisms to consume information from other sources, as well as connectors with SIEM and IRP platforms.

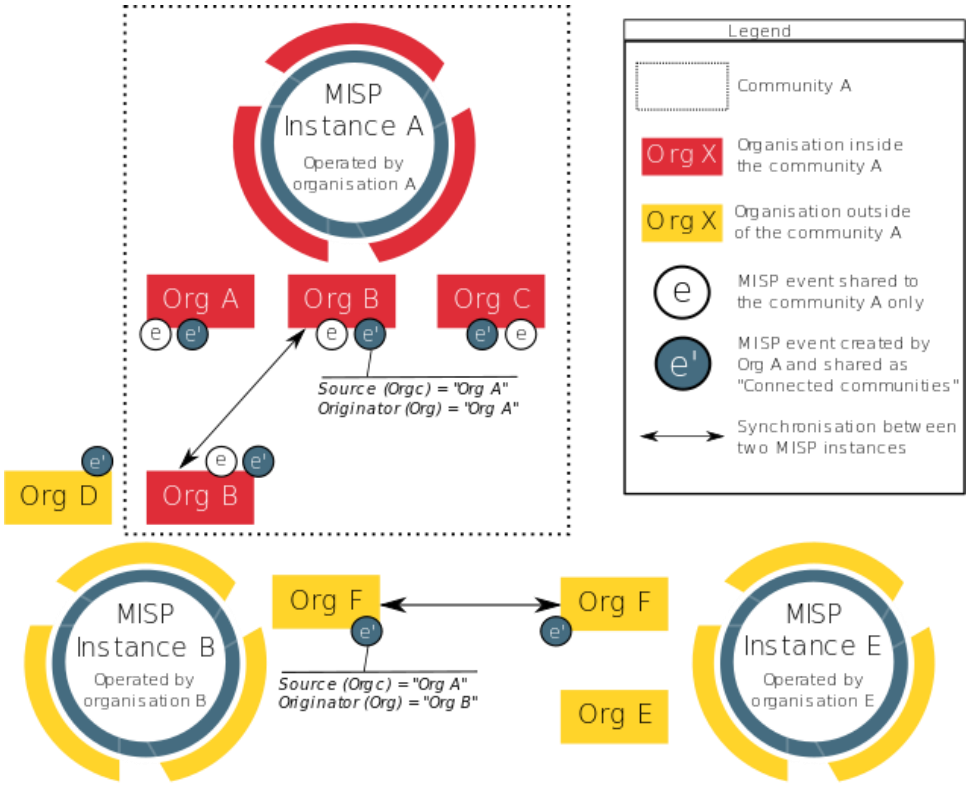

Figure 50: MISP Suitable data model [276]

These tools propose an exchange system based on the sender of the message, since when the information is ready, it can be sent to the receiver, once or in a periodic way. But it is necessary to have an on-demand query system, where the “client” asks for the specific required information. In line with this, the framework has an integration API to all of systems or tools that need to get or update information about threats. This integration API allows users to request threat information, and for users with more privileges, it also allows to modify or even delete this information. However, threat intelligence can also be received from other frameworks to complete and enrich the information. This action is necessary for the exchange of information to be reciprocal. In practice, an API management platform has several components:

- Developer Tool: This is the interface that developers use to define APIs, test and debug them, generate documentation, control the use of APIs through access and use policies, and implement APIs in production, staging, and quality control environments.
- Gateway: To handle data exchange, an API management platform employs a server to authorize requests, pass them to back-end services, and return responses to the requestor. The gateway is in charge of maximizing API efficiency and minimizing downtime.
- Reporting and Analysis: An API management tool should provide a dashboard to monitor and report on API usage and load, including statistics on overall hits, transactions completed, number of data objects returned, amount of compute time and other internal resources consumed, and volume of data transferred. The platform should also generate alerts if it encounters error conditions, such as too high load or hardware or network failure.

Some relevant Open-source API Management Platforms are:

- API Umbrella[212]
- Gravitee.io[213]
- APIman.io[214]
- WSO2 API Manager[215]
- Kong Enterprise[216]
- Tyk.io[217]

- Fusio[218]
- Apigility[219]

All the tools and systems mentioned in "Data Source" are susceptible to include a reciprocal exchange of information, in order to feed and enrich the information on security events that they produce. In this regard, it is possible to use the same tools, protocols and standards, mentioned above, for receive this type of information. STIX or SCAP to represent all the information, and TAXII or MISP to ensure a correct exchange.

On the other hand, SPARTA framework has the capability to access by using an integration API to get or update information. It must be able to access the framework with different roles, a standard user is able to request for information using this API, and users with more privileges are also able to manage, deleting or editing, the threat information.

The manner in which alerts are reported is established by the **NIS** directive, which is a common cybersecurity standard and promotes cooperation between EU countries in the fight to prevent cyberattacks. An important factor of the NIS regulation is to establish a system of cooperation and coordination between CSIRTs (Computer Security Incident Response Team).

NIS regulation establishes that after the first notification of an incident, it must be provided to the CSIRT of reference, at least, the following information:

a) Confirmation of the correctness of the data assigned to the incident, in particular verifying, if this information exists, the validity of:

1.º Incident classification.

2.º Incident dangerousness.

3.º Incident impact.

b) CSIRT action plan to manage the technical resolution of the incident.

c) Any information to determine the potential transboundary impact of the incident.

Whenever possible, the National Platform for Notification and Monitoring of Cyberincidents shall be used for these communications.[277]

The following publications should be taken into account for the sharing and communication on Cybersecurity Threats:

- NIST SP 800-150 ‘Guide to Cyber Threat Information Sharing.[278]
- ENISA ‘Good Practice Guide on Information Sharing’.[279]
- SANS ‘Guidelines for an Information Sharing Policy’.[280]
- MISP ‘ISO/IEC 27010:2015'.[281]

Settings

All data collected, processed, and transformed into information and intelligence, provide the user with a great knowledge of their environment. However, this knowledge has to be organized in order to be helpful in the management and anticipation of threats.

“Data Manager” is able to offer a way to visualize all the information and generate reports with this information. Also, offers the possibility of sharing the information and the reports with other security tools, databases, or people who needs to know this information.

Nevertheless, all this information would be useless if it could not be managed, therefore, the “Settings” process allows users to manage some features of the framework, as the events that can be shown on the “Inspection” process or what type of report can be chosen. Even though “Settings” not only influences the display of the information. “Settings” process allows managing the rest of the modules, “Data Process”, “Data Processor”, “Data Storage”, and “Data Analyzer”.

- **"Data Source": "**Settings" allows selecting what type of input data (SIEM, mail, OSINT Feeds…) will be able to receive. In this way, it can be filter what kind of information will be relevant for analysis.
- **"Data Processor"**: This is the most autonomous module since it can work without previous definitions of the "Settings" process. It only marks the sections of the schema that are not relevant, and do not need to appear.
- **"Data Storage"**: In this module, the "Settings" process allows to choose which information will be stored and which will not be relevant for the databases.
- **"Data Analyzer":** The analyzer is the module with the algorithms and intelligence needed for threat prediction. For this reason, the analysis could be customizable, in order to offer best results. It should be able to customize the characteristics of the analysis, with what information it should be enriched, and what type of algorithms will be better for the prediction.

5.2 Knowledge Representation and Data Models

For a strong and collective defence in the digital domain, we need to produce, consume, analyse, and share cyber threat intelligence. With an increasing amount of information available, we need automation to be effective. Gartner defines threat intelligence as 'evidence-based knowledge, including context, mechanisms, indicators, implications, and practical advice, about an existing or emerging threat or hazard to assets that can be used to inform decisions about the subject's response to that threat or danger.'

Threat intelligence is served in various formats and channels, and with a varying degree of structure. There was a need for a data model that would allow automation and analysis of available threat intelligence. Combining all available data in one place, allowing you to combine and analyze different data sources, will increase an analyst's analytical capacity and eliminate repetitive tasks. Importing and exporting Cyber Threat Intelligence (CTI) from a system is trivial given that the data is stored in a consistent and structured manner, covering all relevant data. The way we model our data is therefore the basis of everything else.

A key requirement for automation and analytics is data quality. Data quality is both content and format. Threat intelligence depends on collaboration between a variety of organizations and communities. Any tools or systems used by contributors must be openly available to the community without restrictions. There are several attempts to structure cyber threat intelligence (CTI). The motivations for the different approaches seem to differ and these influence the results. Barnum suggested the Scrutinized Expression of Threat Information (STIX) [220]. This was created with the motivation to share CTI, preferably as something more than data. STIX was intended as a data exchange format and not as a suggestion on how to store data. Although it was created with the aim of becoming the standard for representing CTI, STIX critics argue that STIX's flexibility makes it less useful for automation, as there are different possibilities to express the same data and information, in addition to a good amount of data included in custom fields or as comments, it is difficult to automate consumption and further analysis.

ATT&CK[165] is a framework and knowledge base for describing adversary behavior though the enumeration of adversary groups, tactics, technique, and tools and the relationships between them. The knowledge base is maintained by MITRE and published online. ATT&CK uses data modelling with defined relationships to structure its knowledge base.

The OpenCTI platform[172] was published at the end of 2019 and is a platform that aims to consume, analyze and share cyber threat intelligence. The OpenCTI platform includes STIX observables and STIX relationships in its data model.

Grank is used to allow graphical query of data and includes rule-based reasoning to infer new relationships. OpenCTI is limited to the scope of STIX and therefore limits the possibilities of consumption and analysis within the platform. An ontology in the field of computer science is a formal description of concepts and how they relate to each other, often referred to as classes and properties.

In turn, ontologies provide computational meaning to data by establishing relationships with logic in ontology and thus allow us to use reasoning methods (such as induction or deduction) in our data in our knowledge base.

While there are many implementations of knowledge bases and ontologies, the World Wide Web Consortium (W3C) chose a three-pronged model for the facts and called it the Resource Description Framework (RDF). RDF also allows us to implement the RDFS2 schema language and OWL3, the ontology language that is based on RDFS.

An ontology, in the field of computer science, is a formal description of concepts and how they relate to each other, often referred to as classes and properties. In turn, ontologies provide computational meaning to data by constructing semantic and logical relationships in ontology that allows us to use reasoning methods (e.g., induction or deduction) on our data in our knowledge base. Although there are many implementations of knowledge bases and ontologies, the World Wide Web Consortium (W3C) chose a triplet model for facts and calls it the Resource Description Framework (RDF). RDF also allows us to implement the RDFS schema

Below is an example list of openly available sources that are commonly used:

Source	Relevant object types
ATT&CK	Tactics, Techniques, Tools, Threat Actor
VirusTotal	IPv4, IPv6, FQDN, URI, Content, Hash, Tool, ToolType, Query, Path, Scheme, Basename
Shadowserver ASN	IPv4, IPv4Network, ASN, Organization, Country
Passive DNS	IPv4, IPv6, FQDN
MISP Galaxies	Tool, Threat Actor, Sector
STIX vocabularies	Sector
Open-Source Intelligence extracted with NLP	All

Table 32: The sources influencing the data model.[286]

5.2.1 Identified key requirements for the data model

Among the various key requirements discussed, three of them were most outstandingly recurrent: flexibility, extensibility/adaptability, and formality. They are described as follows:

Flexibility. Flexibility is an important characteristic, if not the most important one, as the goal of the data model is first to gather multi-domain information and intelligence. Second, a minimal system that can be extended on top is required for multi-domain functionality, such as sharing of cyber-physical information (I.e., influence or destabilization campaigns). The model must be able to define incidents themselves, but as well to have the possibility to add info that is necessary for analytical processes to be applied later on.

Extensibility and adaptability. Another key requirement of the data model is its extensibility. The extension process of MISP is basically a construction mechanism that works with JSON objects and descriptions of the new objects. The only restriction is that these objects are a composite of different data objects like files, for example. A file is made up of certain types of hashes, filenames, and those individual data points that go into an object description, however, have to be in the default building block list of MISP. These types can easily be added in MISP and MISP has currently around 200

data types. The idea is here to use an existing model that is extensible, and to create additional value through the extension of this model. The evaluation and identification of the extensibility specifications of existing models will thus allow eliminating several models.

Formality. Regarding the ontology-based model, the idea is to take a domain, define attributes, identify what kind of attributes are covered, and then create a relationship with the data model. These relations could be used in order to build a data model that would be a transformation model between these domains. This transformation model would use XML or JSON and as further step will be implemented through a MISP instance, to prove the model.

A foundational challenge here is that by mapping, the data is translated into a common language first. The translation represents a risk of losing vital information for automated operations that can be performed if all the data from the different domains are expressed with one format only. The ontology would create additional value in the sense that it would facilitate the construction of translators between taxonomies, and allow people with no strong technical background, to use the data model. Here is the complete data model represented as a graph:

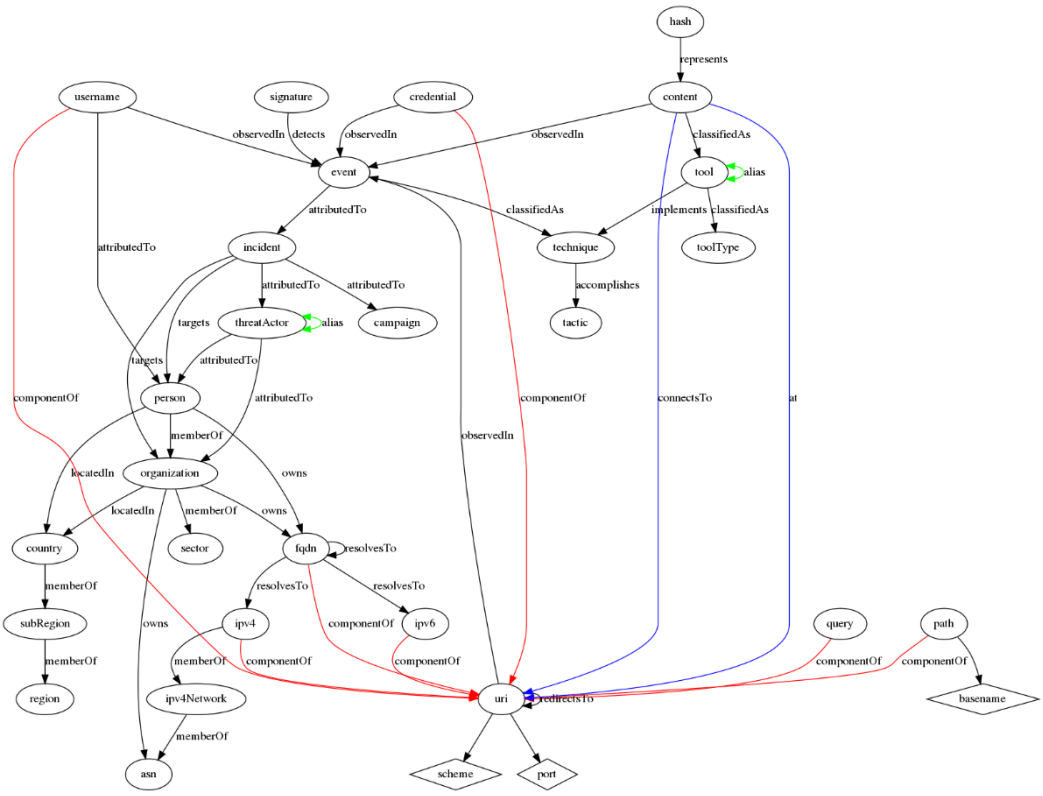

Figure 51:The complete data model represented as a graph[286]

5.3 Knowledge Inference and Reasoning

Perception – the topic of the previous section – is only the first step towards threat prediction. The second and crucial step consists in extracting knowledge – or information – from data. For example, data sources may produce many Gigabytes of data from which the threat prediction information is simply the existence of 3 hosts X, Y, and Z that are part of a botnet as they are contacting a command & control server. Extracting knowledge from data is the topic of this section.

Two important facts can be stated about this problem:

- At the scale of the systems we consider, extracting information from data – typically big data – requires machine learning or data mining methods.
- Given the fact that threats evolve, managing concept drift is essential.

Despite the importance of these two facts, background on these two topics was already discussed in Sections 2.2 and 2.3. Therefore, instead of handling the topic in abstract, we present two approaches we designed, implemented, and evaluated in TSKARK: circumventing scale (DVC20) and detecting change (FDC20).

5.3.1 The general approach

Our approach is based on unsupervised or semi-supervised machine learning. The objective is to detect or predict threats. Most approaches to the problem are either knowledge-based (search for known patterns) or anomaly-based (detect deviations from baseline behavior). However, both have limitations:

- Knowledge-based – knowledge about threats tends to be incomplete, as new threats and threat variants are constantly appearing.
- Anomaly-based – requires clean training data, i.e., data of normal operation without malicious activity, to train the detector, which is hard to obtain in systems in production, i.e., in practical systems and networks.

Our approach instead uses clustering or outlier detection to identify certain entities – typically hosts – that have an anomalous behavior. The idea is to detect anomalies that have to be further diagnosed as threats or another anomaly. This approach has the benefits of not requiring previous knowledge about threats (e.g., in the form of signatures/rules) or clean training data. We consider as data source packet flows, i.e., information about sequences of packets with similar characteristics (e.g., same destination IP address and port). However, other sources might be used.

Next, we explain the two problems we studied: circumventing scale with DynIDS and detecting change with C2BID.

5.3.2 Circumventing scale with DynIDS.

There are several works that use an approach like the one we just explained. Typically, they have features associated to ports, e.g., numbers of bytes sent to a certain port, e.g., 22 (SSH). This allows detecting, e.g., brute force attempts against SSH. All the papers we found considered only a few ports, some arguing that it would be easy to add more ports. However, such a statement ignores a problem known as the curse of dimensionality [VF05]: with many features, typically more than 1000, clustering and similar techniques no longer works as expected as relevant features are masked by others as geometry behaves non intuitively in high dimensions.

With DynIDS we solve this problem: allowing to have many more features for many more ports, without falling into the curse of dimensionality [DVC20]. The idea is to use a method to define features at runtime, “dynamically”, according to data analyzed in each time window (e.g., of 5 or 10 minutes). That is, our approach defines for every period the features that are used in the clustering process, by analyzing the network data corresponding to that period.

We evaluated DynIDS experimentally with a dataset publicly available (CIC-IDS-2018) and real traffic data obtained at a large military infrastructure. The source code is freely available for download.[2] DynIDS achieved an overall F-Score of 0.97 for the public dataset, which is a very good performance; it outperformed related approaches from the literature and alternative approaches. With the real-world dataset DynIDS detected not only the emulated attacks with high recall, but also unexpected anomalies that required further investigation.

The evaluation has also shown that DynIDS is able to predict threats. For instance, it allowed detecting port scans and botnets, which are preliminary steps towards further attacks.

5.3.3 Detecting change with C2BID.

DynIDS and other works that use clustering divide entities, e.g., hosts in groups of entities with similar behavior. Often large groups can be disregarded as containing normal hosts, e.g., workstations, but

[2] https://github.com/a3ceProject/DynIDS

how to identify hosts that contain threats? This is a limitation of that approach. Some works assume clusters with a single entity are suspicious, or clusters with less than a certain threshold number are suspicious.

With C2BID we proposed a new threat detection approach [FDC20]. This approach continues to be based on clustering, but clustering is just the first step. The main idea is to detect intrusions by monitoring host behavior changes. For that purpose, C2BID defines and extracts features from network data – network flow data –, aggregates hosts with similar behavior using clustering, then analyses how hosts move between clusters along a period of time and detects outliers. Detection is only performed at the end of the period of analysis, e.g., a period of one day, not in real time.

C2BID uses sequential series clustering in sequences of time windows to understand how hosts change of clusters. Also relevant is the creation of new clusters. Studying the temporal behavior of clusters, it is possible to identify anomalous behaviors, doing outlier detection that is another form of unsupervised learning. This leads to higher precision than marking only one host cluster. C2BID improves previous works by correlating multiple time windows to detect attacks at different rates and dealing with fixed window limitations.

We evaluated C2BID experimentally with an artificial dataset and a real-world dataset from a military administrative network. Our evaluation has shown that C2BID was able to detect not only the labelled attacks but also found unlabelled (unreported) attacks in both datasets, highlighting the advantages of its unsupervised approach. Moreover, C2BID obtained higher values for F-score and reduced the false positive rate. C2BID also allows predicting threats as it detected, among other anomalies, infiltrations and port scans.[282][283][284][285]

5.3.4 Reinforcement Learning and adaptive tactics.

Reinforcement learning (RL) is an area of machine learning inspired by behaviorist psychology and human physiology with an intrinsic relation with Control Theory, in particular adaptive control [237] and optimal control [238]. It aims to maximize performance. The underneath principle of reinforcement learning consists of a paradigm where an actor or agent perceives a reward or penalty signal after performing an action and tries to maximize the perceived reward in time. The problem of optimizing performance to adopt the best strategy has been described from the point of view of classical planning. RL takes the background from the markovian framework to present a solution to this problem. R.S. Sutton and A.G. Barto are considered the “fathers” of RL. They defined the basis of RL [239] and the architecture for the actor-critic structure.

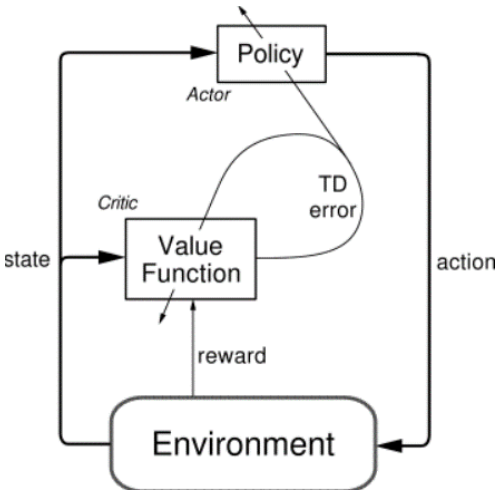

Figure 52: Actor-Critic model as illustrated by Sutton and Barto [239]

Under SPARTA's interests, the advantages of applying reinforcement learning over other fields of machine learning (supervised and unsupervised learning) is the abstraction of the model from the intrinsic dynamics of the model. In this way, instead of assuming a probability distribution to the

model, the learning is in some aspects abstracted from the intrinsic dynamics of the model. This adaptive nature has been traditionally exploited to deal with concept drift [240] [241], but special consideration must be taken in the meta-parameters selection, for example in order to avoid entrenchment syndrome if the learning rate is not set correctly [242]. It defines policies and value functions that reacts to a partially observable environment and reacts to the changes of it in based of actions performed. One of the major advantages of RL is that it provides tools to cope with complexity and dynamics of cyber-attacks, which require protecting mechanisms to be responsive, adaptive, and scalable [243]. As seen in previous section RL has reposted great results on combination with traditional approaches [244] [245], but RL has been extensively applied to cybersecurity in other domains [246]. Therefore, SPARTA will explore and pay special attention to A3C, Double DQN (with multiple complementary strategies), Q-Learning and other RL implementations.

On the other hand, one of the main disadvantages on the use of RL algorithms is the challenges that generates in terms of explainability, mainly due to the large amount of decisions taken by the algorithms in time sequences at nearly real time and the nature of the training data, typically unlabelled. 1.- human collaboration, 2.- visualization, 3.- policy summarization, 4.- query-based explanations and 5.- Verifications are the techniques that will be applied in order to cope with it as suggested in some reviews

5.4 Projection and anticipation

The main scope of this framework is to predict cyberevents, which means, to detect cyberevents with enough anticipation to prepare counter measurements or, at least, to avoid further damage.

In order to do that, it is necessary to choose the metrics that will be predicted; these metrics can be selected from the list of available variables, or a metric created specifically for this kind of analysis, such as a variable which describes the potential damage of a successful attack. The choice or design of these particular metrics should be further analyzed after processing some values of the information collected by the data sources. The final selected metric to be predicted should be selected considering, at least, the following dimensions:

- Dangerousness: as not all events have the same possibilities to be successful attacks or relevant events, they should be analyzed in different ways as these probabilities could significantly affect to the performance; for example, a cyberattack that exploits a recently known Zero-day vulnerability could be detected, but that vulnerability has recently been patched in the infrastructure, thus a highly dangerous event has now lost all its relevance.
- Infrastructure reach: plenty of vulnerabilities and attacks use an element of a certain infrastructure to gain access to different services or equipment. Hence, their analysis will be different as an attack that threats only one personal device cannot be treated as an attack which threats half of the devices connected to the infrastructure.
- Criticality: the value of the elements present in the infrastructure is heterogeneous. For example, some could be disconnected from the network in order to avoid attacks or propagation, while others are so critical for the infrastructure that their performance can’t be interrupted. This disparity is also relevant for the metric.

The available variables should be evaluated according to these dimensions, either qualitative or quantitatively, in order to select the most relevant one to be analyzed. If any discrepancy emerges, a new variable can be created by merging others; this merging is not trivial, as it should be a mathematical combination carefully designed.

The viability of the analysis should be evaluated after selecting the variables to be analyzed and to be predicted, as not all of the desired ones could be suitable. After this evaluation, the model could

be discarded or redesigned, for example, creating a new objective variable to be predicted as it was introduced above.

5.5 Trustworthy AI, Audit and Self-Protection

In the recent years, computational power, large dataset disposability and dedicated hardware have contributed to the fast evolution of AI towards more complex models. Black box models such as Deep Neural Networks have grown in favor of more simple ones. While the former could outperform accuracy of classical methods under previously unaffordable domains, the later are easier to interpret and explain under business logic. Nota bene the ethical implications of providing an AI with the ability to make inferences that affect decision-making and hence responsibility for all derived actions. In other words, deciding based on an AI without understanding why it has given a certain output is, by nature, unethical.

5.5.1 Explainable AI

Explainable artificial intelligence (XAI) is an emerging field of AI. XAI is becoming increasingly relevant, as evidenced by the inclusion of a dedicated symposium on this topic at NIPS 2017 [248]. Since then, contributions to the field have risen emerging new tools and methodologies to bring transparency to AI. In parallel, regulation activities have been taken. For example, to address this problem and regulate potential malpractice arising from the use of AI. In April 2017, Darpa launched an Explainable Artificial Intelligence Program aimed at providing evaluation feedback on the use of XAI to improve user understanding, user trust, and user task performance [249]. In July 2017 Chinese government launched the *"New Generation Artificial Intelligence Development Plan"* in order to establish artificial intelligence laws and regulations, ethical norms and policy systems, and form artificial intelligence security assessment and management capabilities [250]. In June 2018 the EU set up an expert group to provide advice on its artificial intelligence strategy [251]. In the context of SPARTA, XAI comprises as a set of tools, processes and methodologies that aim to enable an actor to understand and trust the results and outcomes generated by the algorithms integrated into the cybersecurity threat prediction framework described on this deliverable. Where actors are any human being or agent involved in the decision-making process as well as any stakeholder.

Under this framework, model evaluation, transparency and impartiality are facilitated by bias mitigation strategies (among other techniques) to have a direct impact on decision making, with the ultimate goal of making these decisions auditable and trustworthy.

XAI is placed in a context that depends on the task, the capabilities, and the expectations of the user [252]. Therefore, in each case the criteria of the system under the specific domain must be defined with reference to explainability and interpretability.

Explainability:

Explainability is the property that provides algorithms the ability to explain or predict the reasoning behind an outcome denoting any action or procedure taken looking for clarifying or detailing its internal functions [253].

Interpretability:

"Interpretability describes to which extend an explanation is understandable for humans" [254]. Explanations supporting the output of a model are crucial, i.e. to provide proper countermeasures for any detected threat.

Below, design criteria are set out that define the above-mentioned criteria under the threat detection framework and taking into consideration the requirements and limitations set forth in previous sections, section 4.3 and 4.4.

5.5.2 Design principles

This sub-section will summarize some of the techniques and tools that the threat prediction framework will leverage. It should be noted that XAI is not an AI capable of explaining itself, but a series of decisions and techniques applied from the design of the development of an AI solution. As proposed by Markus Langer et al., these design decisions must be considered and classified under satisfaction criteria. SPARTAs' design principles consider stakeholders' desiderata as quality criteria from a users' centered design perspective, as recommended by multiple frameworks [255][256]

Figure 53: Quality criteria for explanations in SPARTA.

Reproducibility

Reproducibility in machine learning means being able to replicate the orchestration of the lifecycle processes of the models created in the SPARTA. Special emphasis will be taken on being able to replicate scenarios in which the chain of an attack can be replicated under similar conditions in order to be able to assess reliability. In this sense, SPARTA will promote the use of MlOps-oriented frameworks and orchestration tools such as Kubeflow [257], MLFlow [258], Apache Airflow [259], etc., and will manage data and features from versioning tools such as DVC [260] or Feast [261]. The proposed architecture will allow to replicate the reported results against the original work.

Transparency and Bias

Bias from and estimator in classically defined as the difference between the expectation over the data and the true underlying value from the distribution [262] . In the context of fair and trustworthy AI refers to the discrimination produced when some classes or results are more heavily weighted than others or the nature of the underlying distribution is poorly represented by the training sample.

There exist many different sources of introducing bias in an algorithm from dataset used in the training phase to unforeseen cases in model validation or extreme regularization. It is a great thread in AI practice. An infamous example of this kind of bias could be the application of Framinghams' risk score of any coronary heart disease event, fatal or non-fatal based on features derived from the US population applied to the European populations. The studies carried out over European population evidenced that the reference score obtained from the US population overestimated absolute risk in populations with lower coronary heart disease rates [263].

Many different efforts have been done looking for identifying and classifying the bias in AI.

The mainly identify sources are classified as [264]:

1. Skewed data: This bias comes from the source of acquisition.

2. Tainted data: The main cause of this bias comes from data modelling, categorization and labelling if the domain is not properly represented.
3. Limited features: It is a common practice to reduce dimensionality of input features of a model, but it has the counterpart of inducing some bias in the model, sometimes intentionally, for example, in the case of feature engineering.
4. Sample size disparities: The ideal dataset must provide balanced sets of the most sensitive features.
5. Proxy features: There may be correlated features with sensitive ones that can induce bias even when the sensitive features are not present in the dataset, an example of this bias could be the use of race categories previously commented.

Interpretability helps ensure impartiality in decision-making to correct from bias in the training dataset. Several research groups are currently focused on dealing with bias and fight against bad practices and biased datasets to avoid gender and racial discrimination. IBMs' AI Fairness 360 toolkit provides metrics to detect and removed bias in datasets and models that could be hidden of overseen unintentionally. Microsofts' Fairlearn is another open-source toolkit which provides interactive visualizations and bias mitigation algorithms to explore datasets prior to their use. Another example could be the "Teach and Test" methodology framework from Accenture which aims at helping decision making to overcome bias or other risks mainly focused on financial environments that could assist in the provision of countermeasures and associated costs.

Model Uncertainty

When applying artificial intelligence algorithms to threat prediction it is important to provide decision makers with values that they can use as a reference for better-informed decision making. A reliable system must accompany its predictions with a measure of uncertainty based on the premise that there is no such thing as a perfect system. Therefore, looking at uncertainty provides robustness to a system by allowing it to assess, for example, whether the system in question is basing its predictions on characteristics that can be considered artefacts. Begoli et al.[265] highlighted the importance of uncertainty quantification in guiding decisions based on Deep Learning algorithms. Epistemic uncertainty is usually determined using Bayesian neural networks; in theory, this uncertainty can be modelled [267]. However, a more practical and computationally simple approach is to approximate this Bayesian inference is typically performed by using dropout layers while testing the models, known as Test Timed Dropout. The use of uncertainty maps is another practice to provide better explainability of the models.

5.5.3 Design decisions on Trustworthy AI, Audit and Self-Protection

SPARTA will ensure compliance to the GDPR, with particular emphasis on *right to explanation*, and the forthcoming ePrivacy regulations, embracing a Privacy by Design approach in the whole development and piloting process and follow the guidelines for trustworthy AI developed by the European Commission's High-Level Group on Artificial Intelligence.

Self-protection issues

The SPARTA framework will present the ability to accurately infer on self-protection issues by allowing inference at scales of granularity on self-protection issues. How accurate is each particular function, procedure or calculation that can lead to infer a threat or risk from a set of observed indicators. Both binary and multi-class metrics are typically adopted to assess lower granularity aspects, which can be supplemented with similarity distances for higher-grain qualitative data. Self-protection situations triggered by discordant behaviors can be measured as "Anomaly Recognition" problems, which was preliminarily detailed. Both time series and clustered/conglustered related observations can be adopted to measure dynamic threats, which can also lead to manage sequential situations as "Multi-Step Attacks", which will be validated against them as preliminarily explained. [268][269][270][271][272][273][274][275]

Chapter 6 Framework Validation and Evaluation

This chapters proposes ways to validate and evaluate the instantiations of the proposed framework. First, a methodology for evaluation is chosen. Second, suitable datasets for evaluation are presented. Finally, the proposed solution will be evaluated according to the chosen methodology.

6.1 Evaluation methodology

The instantiation of the proposed solution may be validated and evaluated using different approaches. In an ideal world one would have suitable datasets at hand that covers each and every case of the framework. This way, automated tests can be run and established performance metrics like a confusion matrix, F-score, or else can be computed by comparing the output of the framework with the expected one. If there is no such data available, the solution may be tested using real world data. In that case, however, the outcome cannot be verified automatically but requires human expertise in order to assure the correctness of the output. That yields limited insight and may be labour intense. Finally, if no data nor software is available one might validate the framework by theoretical argumentation. To this end the defined requirements will be used as evaluation criterion to discuss to what extent the proposed framework fulfils them.

Apart from the technical evaluation, the framework also needs to be tested with respect to user research for instance under the aspect of usability. These kinds of studies may be conducted as field experiments or laboratory experiments. The first, will let the users experience the framework in their used-to environment in an unguided fashions. The second allows a more controlled measurement but might cause different behaviour of the users as they might feel observed.

Verification by experimentation with ground truth data

This kind of verification requires datasets that must be complete, significant, and labelled. If these datasets exist, one can automatize the experiments in a reproducible manner. For a further analysis of datasets in question and their availability see the following Section.

For the experimental setup the framework can be fed with data from multiple datasets. The produced output, i.e., threat prediction, may be compared to the ground truth. This way, one can compute the ratio of True Positives (correctly predicted), True Negatives (correctly not predicted), False Positives (incorrectly predicted), and False Negative (incorrectly not predicted). From these, multiple performance metrics may be derived.

Often leveraged metrics comprise Precision (the ratio of correct predictions among all made predictions), Recall (the ration of correctly made predictions among all possible correct predictions), and the F1-score which is the harmonic mean of said metrics.

In order to calculate these metrics known ground truth is required. As already stated in Section 3.4, there are not labelled datasets available for all types of data sources. A manual creation of relevant datasets would be very time consuming and labour intense. Thus, this kind of verification is unlikely to happen.

Verification by experimentation with real world data

Without ground truth data a verification of the framework “in the wild” would be possible. To this end, the framework is setup as it would be used in production. As many data sources as possible are connected to the framework to allow it to make predictions. These predictions, however, cannot be used in the process of decision making. The framework’s output needs to be collected and verified by hand in order to assess the framework’s performance.

An expert must check if the prediction became true in the future in for calculating the framework’s True Positives and False Positives. The opposite metric of Negatives would require a mapping of events to the past in order to check if the framework did or did not predict the threat.

Overall, this kind of verification is suitable if there is no labelled data available. The framework runs in parallel to other methods of threat prediction and is evaluated retrospectively. Results, however, are not as reproducible as with automated ground truth datasets. Moreover, this approach requires manual verification which may be time consuming.

Verification by theoretical argumentation

If neither datasets nor the framework as software is available, one may assess the suitability of the framework by theoretical argumentation. To this end, the beforehand formulated requirements are used as evaluation criterion. If the framework satisfies most or all of them its functionality can be validated. This method, however, is unable to assess the quality of predictions made by the framework.

As there is no dataset and no software available for the proposed solution, it will be validated by theoretical argumentation in the following sections.

6.2 Datasets

The datasets are a precious resource. Even more so if they are publicly available. To this end, we started to look for the relevant datasets in the early stage of the TSHARK programme bearing in mind the comprehensive nature of TSHARK approach to cybersecurity. To this end, we either looked for the specific datasets in the respective domains or a dataset that covers multiple domains.

As a result of our search for the datasets, we prepared a summary table, Table 31, depicting the most relevant datasets, their descriptions, availability, and also explanation if the dataset was used to evaluate a result of a particular subcase.

Subcase	Name	Availability	Link	Description	Utilization
DDoS Backscatter Traffic Detection	DDoS Backscatter Dataset	Public	data.mendeley.com/datasets/37zz4pvjzp/1	DDoS backscatter traffic is generated by the victim of DDoS when it is under an attack. The backscatter traffic is a set of packets the victim generates as a response to the query packets with the spoofed source IP addresses received from the attacker. The uploaded dataset serves to train a classifier which is able to recognize DDoS backscatter traffic from the legitimate traffic. The csv files contain feature vectors (one feature vector per line) computed per victim DDoS IP addresses. The label feature is called DDOS (DDOS column, True in case the IP address is under DDoS, False otherwise). The IP addresses in the dataset are anonymized by a prefix preserving algorithm.	Yes.
DDoS Backscatter Traffic Detection	DDoS Evaluation Dataset (CIC-DDoS2019)	Public	www.unb.ca/cic/datasets/ddos-2019.html	CICDDoS2019 contains benign and the most up-to-date common DDoS attacks, which resembles the true real-world data (PCAPs). It also includes the results of the network traffic analysis using CICFlowMeter-V3 with labelled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files).	No, the dataset does not contain backscatter DDoS traffic.
Virtual Control Room	Chomium web pages	Internal	N/A	Leonardo's DSS web page (dashboard) Theat Intelligence System web page CCTV camera web page; PLC Siemens web page	Yes.

Subcase	Name	Availability	Link	Description	Utilization
Detection of cyber and physical attacks on critical infra…	JCCI railway dataset	Public, based on subscription under SPARTA JCCI	https://jcci-node.cyber.inov.pt/	INOV has created a Rail infrastructure simulator that creates network communication messages exchanged between the rail line systems, central control server and public displays at the stations. The installation adapted BP-IDS to monitor train activities on the Portuguese rail stations between Cascais and Cais do Sodré. The simulator uses a set of pre captured packets based in this scenario and modifies the packet payload with the respective simulation information. The packets have the ip address of the elements in the network and simulate a capture on several points of the scenario network. The packet frame is unmodified and is the same for all the same packets in a simulation execution. The simulator injects these packets into an interface, respecting the simulation order. This Rail infrastructure simulator creates network communication messages exchanged between the command-and-control server and station servers. From that, it is possible to track the train position and the schedule for train arrivals and departures.	Yes, the dataset was used to validate BP-IDS as described and evaluated on 5.3.1 This dataset was used on the BP-IDS publication https://ieeexplore.ieee.org/abstract/document/9307723 Dataset used in part for the validation of the tool Incident Prediction Engine.

Subcase	Name	Availability	Link	Description	Utilization
Detection of cyber and physical attacks on critical infra…	EPIC - power grid dataset	Public	itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_epic	The dataset contains records of network operations performed in a power grid test-bed composed of five classes of assets. Generation control assets, composed of: one PLC and two IED devices directly connected to power generators. Transmission control assets, composed of one PLC and three IED directly connected to auto-transformers. Smart home control assets, composed of one PLC and four IED devices directly connected to loadbanks. Microgrid control assets, composed of one PLC and three IED devices directly connected to photovoltaic cells. Supervisory assets, composed of a PLC, one Raspberry Pi, SCADA system that issues operations to control units and registers the results on a historian database. The overall system described in the dataset is composed of IT devices (11 network switches, IoT devices and 1 SCADA workstation, 1 SCADA historian and 1 AMI), control units (5 PLCs and 11 IEDs), and physical components (such as power supply unit, circuit breakers), that communicate using ICS protocols (MMS, GOOSE, DCE-RPC, MODBUS) and IT protocols (TCP, UDP, ICMP, SNMP, etc.). The packet captures found on the dataset portray the facility running for about 30 minutes under different configurations (eight different scenarios). In scenarios 1-6, the system is running with different physical characteristics. Scenario 5, the power grid is connected to three external generators. In scenarios 7 and 8, the EPIC power grid is supplying energy to other iTrust lab systems. In scenario 7, EPIC is powering a water treatment system (SWaT). In scenario 8, EPIC is powering a water distribution infrastructure (WADI).	Yes, the dataset was used to validate BIA as described and evaluated on 5.3.2 This dataset was used on the BIA publication at SAC2022: CIIA: Critical Infrastructure Impact Assessment https://www.sigapp.org/sac/sac2022/file2022/TOC-Jan-23-2022.pdf

Subcase	Name	Availability	Link	Description	Utilization
Detection of cyber and physical attacks on critical inf…	Baggage handling system data	Private, third party, NDA	N/A	The simulation platform offers airport digital twins. The digital twins installed on the platform include an airport network infrastructure with VLAN-connected virtual machines. The platform mimicked a BHS manufactured by Alstef6. This simulation platform included a virtual AODB that provided the BHS sortation unit with fictitious identifiers of bags and flights assigned to BHS locations. The simulation also included virtual physical equipment (EDS, ATR, and conveyors) managed by PLCs connected to the BHS sortation unit in the simulation platform. The simulation used Emulate3D7 to mimic the physical equipment. Emulate3D is a high-fidelity emulator used by BHS providers to test their systems against contractual operating conditions before airport installment.	Yes, the dataset was used to validate FingerCI as described and evaluated on 5.3.2 This dataset was used on the FingerCI: publication at SAC2022: FingerCI: Generating Specifications for Critical Infrastructures https://www.sigapp.org/sac/sac2022/file2022/TOC-Jan-23-2022.pdf

Subcase	Name	Availability	Link	Description	Utilization
Detection of cyber and physical attacks on critical infra….	Secure Water Treatment (SWaT)	Public	https://itrust.sutd.edu.sg/testbeds/secure-water-treatment-swat/	Secure Water Treatment (SWaT) is a water treatment testbed for research in the area of cyber security. SWaT consists of a modern six-stage process. The process begins by taking in raw water, adding necessary chemicals to it, filtering it via an Ultrafiltration (UF) system, de-chlorinating it using UV lamps, and then feeding it to a Reverse Osmosis (RO) system. A backwash process cleans the membranes in UF using the water produced by RO. The cyber portion of SWaT consists of a layered communications network, Programmable Logic Controllers (PLCs), Human Machine Interfaces (HMIs), Supervisory Control and Data Acquisition (SCADA) workstation, and a Historian. Data from sensors is available to the SCADA system and recorded by the Historian for subsequent analysis.	No, the dataset lacked useful information about the nominal values necessary for the full experimentation with the Incident Prediction Engine tool.

Subcase	Name	Availability	Link	Description	Utilization
Detection of cyber and physical attacks on critical infra…	LANL corporate network dataset	Public	https://csr.lanl.gov/data/	This dataset represents 58 consecutive days of pseudonymized event data collected from five sources: authentication events, hosts process start/ stop, DNS, netflow, and red team events. We did not use the whole dataset but only the redteam and netflow data. The netflow events have 1.1 GB when compressed and correspond to 129,977,412 events for 12,027 computers. The red team events provide us with attacker IP addresses, only 4, that we use to identify malicious events in the other dataset, i.e., to obtain ground truth for the evaluation. The dataset comes in text files. Each line of the netflow event files contains a timestamp (an epoch time starting at 0), connection duration, source computer, source port, destination computer, destination port, protocol, packet count, and byte count. The well-known ports (e.g., 80 and 443) are not pseudonymized, only the IP addresses.	Used to evaluate the Outgene threat detection system.
Detection of cyber and physical attacks …	Military network dataset	Private	N/A	This dataset was obtained by the IST team from the Security Information and Event Management (SIEM) system in production in that network, which collects Netflow events from internal routers. Collecting these flows can give us insights of eventual misbehavior of internal entities, undetected by deployed security systems. The dataset corresponds to a full month, with approxi- mately 5,500 computers and 160 GB of size. We emulated 4 stealth/slow attacks (e.g., probing) at different paces. We also emulated a noisy attack (high volume of data exfiltrated to an unexpected destination).	Used to evaluate the Outgene, DynIDS and C2BID threat detection systems.

Subcase	Name	Availability	Link	Description	Utilization
Detection of cyber and physical attacks …	CIC-IDS-2018	Public	www.unb.ca/cic/datasets/ids-2018.html	This dataset was developed to provide data to analyse, test and evaluate NIDSs. To generate the dataset, its authors developed a systematic approach in order to produce a diverse and comprehensive benchmark dataset. They created user profiles with abstract representations of activity seen on typical networks. The benign behavior of each machine was generated using a tool to generate B-Profiles, i.e., realistic benign behaviors of a network. The tool uses machine learning and statistical analysis techniques to generate network events as if users in a typical network produced them. The network topology represents a typical medium company, with six subnets, deployed on the AWS computing platform. This dataset includes seven different attack scenarios: Brute- force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside. The attacks were performed from one or more machines, using Kali Linux, in a specific network (within public IPs range) created only to attacker machines. Some of the tools used were Patator for brute force, Ares botnet, Selenium and Heartleech for web testing, Hulk, GoldenEye, Slowloris, Slowhttptest for DoS, and Low Orbit Ion Canon (LOIC) for DDoS.	Used to evaluate the DynIDS and C2BID threat detection systems.

Subcase	Name	Availability	Link	Description	Utilization
Research: 5G network and elections	Cell tower usage statistics	Internal	N/A	Mobile network activity data. Aggregated periodic data containing 3G/4G/5G technology activity counts as well as total transferred data rates are those indicators describing the activity of each base station.	Yes, data was used to draw conclusions of election activity monitoring based on mobile network activities
Anticipation of the cyber-physical attack on Transport CII	Evaluation of Cyclostationarity-Based Network IDSs	Public	https://nesg.ugr.es/nesg-ugr16/march.php#INI	The dataset presented here is built with real traffic and up-to-date attacks. These data come from several netflow v9 collectors strategically located in the network of a spanish ISP. It is composed of two differentiated sets of data that are previously split in weeks: A CALIBRATION set of data gathered from March to June of 2016 (4 months) containing real background traffic data A TEST set of data gathered from July to August of 2016 containing real background and synthetically generated traffic data that corresponds with several and well know types of attacks. The main advantage of this dataset over previous ones is its usefulness for evaluating IDSs that consider long-term evolution and traffic periodicity. Models that consider differences in daytime/night or labour weekdays/weekends can also be trained and evaluated with it.	Yes.

Table 33: Datasets studied for evaluation of framework instantiations

The table shows the datasets that were used to assess individual results. The list captures a diverse set of publicly available datasets, internal datasets and third-party datasets that were provided based on membership in the community or close cooperation with the company in the respective domain. In some cases, such as in the case of DDoS Backscatter Dataset, the TSHARK partner anonymized its originally internal dataset and made it publicly available or in the case of the case of JCCI railway dataset it is catalogized within SPARTA Joint Competence Centre Infrastructure system. Nevertheless**, there is no dataset available to enable the evaluation of the TSHARK Prediction framework instantiation as a whole**.

6.3 Results

As decided in the previous Section, the validation and evaluation of the proposed framework will be based on theoretical argumentation. To this end, the requirements as formulated in Section 4.3 have been used as evaluation criteria. The comparison of the requirements against the framework are displayed in the table below. Out of 13 requirements the proposed framework fulfils 11. Only requirement R06 and R10 are partially fulfilled. For R06 mainly the way of operating the framework is decisive. However, it is already planned to perform hardening, self-protection and recurrent audits. In case of R10, it might be possible to train the models on different time horizons. It has been considered but not yet fully argued.

#	Short Name	Ful-filled	Reason
R01	Extensibility, Expandability and Multi-level scalability by design	Yes	Modular software architecture that allows to easily add new sources, processors and analysers.
R02	Non-Stationarity	Yes	Concept drift detector in the analyser module
R03	Uncertainty	Yes	Providing confidence of each predictor for the outcome
R04	Open solutions and COTS reliant	Yes	Market analysis and use of existing technologies for singular tasks inside the framework
R05	Privacy by design	Yes	compliance to the GDPR
R06	Security by design	Partially	Depends on operation/hosting and audits
R07	Ethics and trustworthy AI by design	Yes	Providing probabilistic reasoning for each AI-based predictor
R08	Human-centred by design	Yes	Human–computer interface in the data manager module that allows inspection and report of information
R09	Full-spectrum cyber security threats	Yes	Multiple and heterogeneous data sources

#	Short Name	Ful-filled	Reason
R10	Threat projection at different time horizons	Partially	Models may be trained on different time horizons
R11	Non-Determinism	Yes	Zoo of multiple and different models
R12	Reduce the human intervention	Yes	Full automatization of data collection, processing, storing, and analysis
R13	Notification and Information sharing	Yes	Automated reporting and sharing capabilities of the data manager module

Table 34: Results of Framework Validation and Evaluation

Overall, the participant and consulted stakeholders agreed that the framework seems suitable for the task of threat prediction based on full-spectrum threat intelligence from multiple and heterogeneous data sources. Everything for the technical function is set up and considered.

However, as mentioned in Section 4.2 and Section 4.4, there are some crucial assumptions and limitations. Most noticeable is the lack of suitable datasets. Without datasets that resemble the actual data sources no models can be trained, and predictions cannot be verified easily.

Chapter 7 Requirements traceability

The proposed framework has been designed with the intention of satisfy the requirements defined in the chapter 3.3 of the document:

- Req 01 - Extensibility, Expandability and Multi-level scalability by design: as the proposed framework is built by different components with their own functionalities. The vertical architecture hierarchization of the components allows to modify, update or create new elements easily.
- Req 02 - Non-Stationarity: thanks to the internal and external data sources, the available information is constantly updated, and the prediction models are consequently updated and constantly evaluated, which leads to models adapted to the current environment.
- Req 03 - Uncertainty: the implementation of different prediction and learning model paradigms (e.g. supervised, federated, frugal) adapts the framework to a probabilistic approach.
- Req 04 - Open solutions and COTS reliant: the different elements which compose the framework are based on pre-built frameworks or solutions, so its use is trustable.
- Req 05 - Privacy by design: the data collection and storage managed in this framework are aligned to the European privacy legislation.
- Req 06 - Security by design: the deployment and communications between the different elements should be internally audited and secured.
- Req 07 - Ethics and trustworthy AI by design: the data collection and storage managed in this framework are aligned to the European ethics and trustworthy AI legislation.
- Req 08 - Human-centred by design: the implementation of a presentation layer which offers user-friendly interfaces will improve the understanding of the predictions.
- Req 09 - Full-spectrum cyber security threats: by using plenty of internal and external data sources embraces the full spectrum of known cyber security threats.
- Req 10 - Threat projection at different time horizons: as the framework uses different learning and prediction models, each one of them will be more feasible under certain circumstances, like the different time horizons.
- Req 11 - Non-Determinism: like in the previous requirement, different models lead to different performances, hence, under certain circumstances any of those predictions (despite its likelihood) can lead to plan countermeasures.
- Req 12 - Reduce the human intervention: as the proposed models include semi-supervised and unsupervised models, the human intervention can be avoided as much as desired.
- Req 13 - Notification and Information sharing: multiple solutions are included in this framework to communicate and share information, such as protocols, APIs and alert managers.

Chapter 8 Wide-system design/implementation decisions

All the researches and discussions thorough the project follow the requirements previously cited. As not all of the components are affected by the same requirements and limitations, next each section will be analyzed:

- When talking about the Situational Awareness, which in other words means the current situation that involves the infrastructure and how can we know about it, some technologies to gather information are evaluated, and it is strictly related to the requirements 2 (Non-Stationarity) and 9 (Full-spectrum cyber security threats) as this systems provide information about a wide variety of current threats.
- Next, some prediction frameworks are compared; in order to do so, a SWOT analysis were performed to assess the proposed frameworks/technologies/tools. Once again, no one was discarded as its use could be relevant under some circumstances, following the non-determinism established in the Req-11. According to the SWOT analysis, it looks like some technologies like TensorFlow or PyTorch could be more relevant as they directly satisfy most of the Requirements (those which apply to this kind of technologies Req 1-4 and 10-13). Furthermore, some data could need to be processed before it can be used in the analysis, and the frameworks based on technologies like Python could ease this processing.
- Later, these results will be shown to the user using an interface, which correspond to the Presentation Layer (this layer is also includes other functions, as it can be used to interact with the database, for example) and is strictly related to the Req-08 as it focuses on the human interaction. Also, these results and data could be shared and communicated, and it should always compliance the European regulations (Req 5-7).
- Finally, when talking about the performances and model selection, all of them should be periodically (at least) evaluated as their performance could be relevant in some cases, according to the requirements 3 and 11 which establish that the results should be based on probabilities that could be subjectively selected under some circumstances.

Chapter 9 Lessons Learned

Along the researches developed for this project, some aspects were undoubtedly more relevant than others. Next, the most important lessons learned during this researched will be highlighted:

- First of all, in this kind of projects the main core resides in the data, both qualitative and quantitatively, as trying to predict event using limited or poor information will lead to models and predictions that cannot be trusted. In order to do so, multiple internal (e.g. SIEM) and external (e.g. databases) sources should be included if possible, but the data should be processed before ingesting it into the prediction or learning models, as the use of multiple sources could lead to data redundancy and model deterioration. This should be one of the main focus in every prediction model.
- As it has been mentioned, the data needs to be processed; this is not an easy or trivial task, as it is multidimensional and will directly affect the models performance. This procedure will translate the different sources into a common language, standardizing the messages under a common syntax, to be later analyzed. This includes three tasks (process, enrich and label the data) and every single one of them must be carefully studied, as they include different sub-tasks as it has been specified in the project.
- Other relevant aspect references the data storage. This task includes both technical and legal requirements. Briefly, the data storage should satisfy requirements about storage capacity, write and read speed (which includes hardware and software specifications), data access, data management and multiple legislations compliance could apply, such as GDPT, PCI/DSS, HIPAA or SOX.
- When analyzing the data, multiple options are available; this includes modelling scopes, different frameworks and technologies. This research included multiple information about different possibilities, which should be independently evaluated in the actual deployment before discarding any of them. Nonetheless, multiple models could be implemented in the final solution as the general performance could be evaluated according to the combination of different results.
- Finally, the visualization and sharing of the information and results also needs to be made according to the different legislations, as its compliance also affects the information sharing.

Chapter 10 Conclusions and future research work

Many organizations produce, collect, and share information related to potential and known cyberattacks. According to the National Institute of Standards and Technology (NIST), cyber threat information is any information related to threats that might help organizations in protecting themselves against cyberattacks or in detecting the activities of adversaries, while Cyber Threat Intelligence (CTI), is what threat information becomes after its processing and analysis. Another definition given by Gartner considers CTI as evidence-based knowledge that includes context, mechanisms, indicators, implications, and actionable advice about an existing or emerging menace or hazard to IT or information assets.

This deliverable documents our approach with which we will address the challenge to establish, within the framework of the SPARTA project, adopting an eminently practical approach, a descriptive of a series of solutions, systems and tools that cover the different phases defined, within the thread intelligence cycle, which will help us manage the information collected and available, enrich this information and its subsequent treatment and making available for processing and analysis, and thereby generate Cyber Threat Intelligence, for the fight against cyberattacks on our organization.

Likewise, this information, already collected and enriched, can be shared efficiently, using different tools and environments between different entities. Once the state of the art and the evaluation of the different proposed tools have been carried out, we relate in the document the different components that would make up the architecture of the proposed solution, complying with the requirements and needs of SPARTA.

To finish our research and the proposal of the different tools, solutions, and systems to be implemented have led us to find several areas and possible options for future development of solutions of great applicability in these CTI environments, which will allow us to adapt naturally to the possible changes to come in the field that concerns us.

As potential opportunities or future improvements for this work, the following aspects could be explored:

- Integration with the SPARTA Data Sharing infrastructure, as it is a keystone for T-SHARK.
- Integration with EU-HYBNET, enabling a Pan-European collaborative security network.
- Countermeasure proposal: after the data analysis and threat identification, the system could automatically propose appropriate countermeasures to mitigate or avoid the risks.
- Interface centralization: as this project implies the use and cooperation of different tools and solutions, the unification of the interface that is shown to the users could highly improve the UX.
- Continuous model evaluation: along the evolution of the threat environment, the computational environment is also continuously evolving. The optimal approach will likely change as the data, the hardware, and the threats change, so it should be periodically re-evaluated.
- Liaising: as well as many sources have been proposed to be used for this project, the coordination and cooperation between related projects should be kept in mind for its liaising.

Chapter 11 List of Abbreviations

Abbreviation	Translation
ACID	Atomicity, Consistency, Isolation, Durability
AIS	Automated Indicator Sharing
API	Application Programming Interface
APTs	Advance Persistent Threats
AWS	Amazon Web Service
C2BID	Cluster Change-Based Intrusion Detection
CCTV	Closed-circuit television
CEF	Common Event Formal
CERT	Computer Emergency Response Team
CIDR	Classless Inter-Domain Routing
CIRCL	Computer Incident Response Center Luxembourg
CISA	Cybersecurity and Infrastructure Security Agency
CLI	Command-Line Interface
CoAs	Courses of Action
COTS	Commercial off-the-shelf
CPS	Cyber Physical Systems
CRLF	Carriage Return and Line Feed
CSA	Cyber situation awareness
CSIRTs	Computer Security Incident Response Team
CSV	Comma-separated values
CTI	Cyber threat intelligence
CTIP	Cyber Threat Intelligence Platform
CUDA	Compute Unified Device Architecture
CVE	Common Vulnerabilities and Exposures

Abbreviation	Translation
CVSS	Common Vulnerability Scoring System
DC3	Defense Cyber Crime Center
DDoS	Distributed Denial of Service
DEF CON	Defense Condition
DIME	Diplomatic, Information, Military and Economic
DM	Data Management
DNS	Domain Name System
DVC	Digital Volume Correlation
DynIDS	Dynamically Intrusion Detection System
EDoS	Economical Denial of Sustainability
EDR	Endpoint detection and response
ELM	Extreme learning machine
ENISA	European Union Agency for Network and Information Security
EPS	Earnings Per Share
FDC	Floppy Disk Controller
FedAvg	Federate Average
FedConD	Federated Learning for Sensor Data with Concept Drift
FedMA	Federate Matched Average
FedSGD	Federated Stochastic Gradient Descent
FIFO	First-In, First-Out
FIRST	Forum of Incident Response and Security Teams
FL	Federate Learning
FLUX	Florida Linux User Xchange
GCT	Granger Causality Test
GDRP	General Data Protection Regulation
GPL	General Public Licence

Abbreviation	Translation
HDFS	Hadoop Distributed File System
HFL	Horizontal FL
HIDS	Host-based Intrusion Detection
HIPAA	Health Insurance Portability and Accountability Act
HIPS	Host-based Intrusion Prevention System
ICT	Information Communication Technology
IDS	Intrusion detection systems
IoCs	Indicators of Compromise
IODEF	Incident Object Description Exchange Format
KNN	K-Nearest-Neighbor
LIDS	Log Intrusion Detection Systems
LIFO	Last-In, First-Out
LSTM	Long Short-Term Memory
MISP	Malware Information Sharing Platform
ML	Machine learning
MLP	Multilayer Perceptron
MORDA	Mission Oriented Risk and Design Analysis of Critical Information
MQTT	Message Queuing Telemetry Transport
MTL	Multi-Task Learning
MTTD	Mean time of detection
MTTR	Average time to resolution
NCIRC	NATO Computer Incident Response Capability
NIDS	Network Intrusion Detection Systems
NSQ	Network & Systems Quality
OODA	Observe Orient Decide Act
OS	Operating System

Abbreviation	Translation
OSINT	Open-Source Intelligence
OSSEC	Open Source HIDS Security
PCI/DSS	Payment Card Industry Data Security Standard
PH	Page-Hinkley
PLCs	Programmable Logic Controllers
RAT	Remote Access Trojan
RDF	Resource Description Framework
RMON	Remote Network Monitoring
RO	Reverse Osmosis
ROC	Receiver Operating Characteristic
RSS	Rich Site Summary
SA	Situational Awareness
SaaS	software-as-a-service
SCADA	Supervisory Control and Data Acquisition
SCAP	Security Content Automation protocol
SCP	Cyber security and privacy
SIEM	Security Information and Event Management
SIM	Security Incident Management
SIRP	Security Incident Response Platform
SMEs	Small and medium-sized enterprises
SOAR	security orchestration, automation, and response
SOC	Security Operations Center
SOCMINT	Social Media Intelligence
SOX	Sarbanes Oxley
SQL	Structured query language
STIX	Structured Threat Information eXpression

Abbreviation	Translation
SVM	Support Vector Machine
SWaT	Secure Water Treatmen
SWOT	Strength weakness opportunity and threat
TAXII	Trusted Automated Exchange of Intelligence Information
TIP	Threat Intelligence Platform
TLP	Traffic Light Protocol
TOR	The Onion Routing
TTP	Techniques Tactics and Procedures
TTPs	tactics, techniques, and procedures
UF	Ultrafiltration
UI	User Interface
USAF	United State Air Force
USB	Universal Serial Bus
VFL	Vertical FL
XML	Extensible Markup Language
XSS	Cross-site Scripting

Chapter 12 References

[1] Martínez, Á. L., Vidal, J. M., & González, V. A. V. (2021). Understanding and Assessment of Mission-Centric Key Cyber Terrains for joint Military Operations. *arXiv preprint arXiv:2111.07005*.

[2] M. Endsley, Situation awareness global assessment technique (SAGAT), in: Proceedings of the IEEE 1988 National Aerospace and Electronics Conference, Vol. 3, IEEE, Dayton, OH, USA, 1988, pp. 789–795. doi: 10.1109/NAECON.1988.195097.

[3] G. Bedny, D. Meister, Theory of Activity and Situation Awareness, International Journal of Cognitive Er-gonomics 3 (1) (1999) 63–72, publisher: Routledge eprint: https://doi.org/10.1207/s15327566ijce0301 5. doi:10.1207/s15327566ijce0301_5. URL https://doi.org/10.1207/s15327566ijce0301_5

[4] K. Smith, P. A. Hancock, Situation Awareness Is Adaptive, Externally Directed Consciousness, Human Factors 37 (1) (1995) 137–148, publisher: SAGE Publications Inc. doi:10.1518/001872095779049444. URL https://doi.org/10.1518/001872095779049444

[5] M. P. T. Jr, Shaping and adapting: Unlocking the power of Colonel John Boyd's OODA Loop (Apr. 2015).

[6] S. Jajodia, S. Noel, P. Kalapa, M. Albanese, J. Williams, Cauldron: Mission-centric cyber situational aware-ness with defense in depth, Proceedings - IEEE Military Communications Conference MILCOM (2011) 1339–1344doi:10.1109/MILCOM.2011.6127490.

[7] J. Maestre Vidal, A. Orozco, L. Villalba, Adaptive artificial immune networks for mitigating DoS flooding attacks, Swarm and Evolutionary Computation 38 (2018) 94–108.

[8] K. Demertzis, N. Tziritas, P. Kikiras, S. Llopis Sanchez, L. Iliadis, The Next Generation Cognitive Security Operations Center: Network Flow Forensics Using Cybersecurity Intelligence, Big Data and Cognitive Computing 2 (35).

[9] J. Maestre Vidal, M. Sotelo Monge, Obfuscation of Malicious Behaviors for Thwarting Masquerade Detection Systems Based on Locality Features, Sensors 20(7) (2084).

[10] M. A. Sotelo Monge, J. Maestre Vidal, G. Martínez Pérez, Detection of economic denial of sustainability (edos) threats in self-organizing networks, Computer Communications 145 (2019) 284–308.

[11] K. Demertzis, N. Tziritas, P. Kikiras, S. Llopis Sanchez, L. Iliadis, The Next Generation Cognitive Security Operations Center: Adaptive Analytic Lambda Architecture for Efficient Defense against Adversarial Attacks, Big Data and Cognitive Computing 3 (6)

[12] J. Maestre Vidal, M. Sotelo Monge, Framework for Anticipatory Self-Protective 5G Environments, in: Proceedings of the 14th nternational Conference on Availability, Reliability and Security (ARES).

[13] J. Maestre Vidal, M. Sotelo Monge, A novel Self-Organizing Network solution towards Crypto-ransomware Mitigation, in: Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES).

[14] McGuinness, B., Foy, L., A Subjective Measure of SA: The Crew Awareness Rating Scale (CARS), in: 1st, Human performance, situation awareness and automation conference; user-centered design for the new millennium, SA Technologies, Savannah, GA, 2000, p. 6.

[15] V. Lenders, A. Tanner, A. Blarer, Gaining an Edge in Cyberspace with Advanced Situational Awareness, IEEE Security Privacy 13 (2) (2015) 65–74, conference Name: IEEE Security Privacy. doi:10.1109/MSP.2015.30.

[16] D. Buckshaw, G. Parnell, W. Unkenholz, D. Parks, J. Wallner, O. Saydjari, Mission Oriented Risk and Design Analysis of Critical Information Systems, Military Operations Research 10 (2005) 19–38. doi:10.5711/morj. (10.02.19)

[17] D. S. E. Noel, W. J. Heinbockel, An Overview of MITRE Cyber Situational Awareness Solutions, in: NATO Cyber Defence Situational Awareness Solutions Conference, MITRE, Bucharest, Romania, 2015, p. 17. URL https://www.mitre.org/publications/technical-papers/an-overview-of-mitre-cyber-situational-awaren

[18] J. Hingant G´omez, M. Zambrano, F. P´erez, I. Perez-Llopis, M. Esteve, HYBINT: A Hybrid Intelligence System for Critical Infrastructures Protection, Security and Communication Networks 2018 (2018) 1–13. doi:10.1155/2018/5625860.

[19] Franke, U., & Brynielsson, J. (2014). Cyber situational awareness–a systematic review of the literature. Computers & security, 46, 18-31.

[20] Wagner, T. D., Mahbub, K., Palomar, E., & Abdallah, A. E. (2019). Cyber threat intelligence sharing: Survey and research directions. Computers & Security, 87, 101589.

[21] ENISA. (November 2014). Standards and tools for exchange and processing of actionable information.

[22] Vasilomanolakis, E., Karuppayah, S., Mühlhäuser, M., & Fischer, M. (2015). Taxonomy and survey of collaborative intrusion detection. ACM Computing Surveys (CSUR), 47(4), 1-33.

[23] Bartos, V., Zadnik, M., Habib, S. M., & Vasilomanolakis, E. (2019). Network entity characterization and attack prediction. Future Generation Computer Systems, 97, 674-686.

[24] Sun, N., Zhang, J., Rimba, P., Gao, S., Zhang, L. Y., & Xiang, Y. (2018). Data-driven cybersecurity incident prediction: A survey. IEEE communications surveys & tutorials, 21(2), 1744-1772.

[25] Husák, M., Komárková, J., Bou-Harb, E., & Čeleda, P. (2018). Survey of attack projection, prediction, and forecasting in cyber security. IEEE Communications Surveys & Tutorials, 21(1), 640-660.

[26] Husák, M., Bartoš, V., Sokol, P., & Gajdoš, A. (2021). Predictive methods in cyber defense: Current experience and research challenges. Future Generation Computer Systems, 115, 517-530.

[27] Zhang, K., Luo, S., Xin, Y., Zhu, H., & Chen, Y. (2020). Online mining intrusion patterns from ids alerts. Applied Sciences, 10(8), 2983.

[28] Katipally, R., Yang, L., & Liu, A. (2011, October). Attacker behavior analysis in multi-stage attack detection system. In Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research (pp. 1-1).

[29] Bar, A., Shapira, B., Rokach, L., & Unger, M. (2016, June). Identifying attack propagation patterns in honeypots using Markov chains modeling and complex networks analysis. In 2016 IEEE international conference on software science, technology and engineering (SWSTE) (pp. 28-36). IEEE.

[30] Ghafir, I., Kyriakopoulos, K. G., Lambotharan, S., Aparicio-Navarro, F. J., AsSadhan, B., BinSalleeh, H., & Diab, D. M. (2019). Hidden Markov models and alert correlations for the prediction of advanced persistent threats. IEEE Access, 7, 99508-99520.

[31] Perry, I., Li, L., Sweet, C., Su, S. H., Cheng, F. Y., Yang, S. J., & Okutan, A. (2018, December). Differentiating and predicting cyberattack behaviors using lstm. In 2018 IEEE Conference on Dependable and Secure Computing (DSC) (pp. 1-8). IEEE.

[32] Werner, G., Yang, S., & McConky, K. (2017, April). Time series forecasting of cyber attack intensity. In Proceedings of the 12th Annual Conference on cyber and information security research (pp. 1-3).

[33] Sokol, P., & Gajdoš, A. (2017, September). Prediction of attacks against honeynet based on time series modeling. In Proceedings of the Computational Methods in Systems and Software (pp. 360-371). Springer, Cham.

[34] Wang, H., Zhao, D., & Li, X. (2020). Research on network security situation assessment and forecasting technology. Journal of Web Engineering, 1239-1266.

[35] Abdlhamed, M., Kifayat, K., Shi, Q., & Hurst, W. (2017). Intrusion prediction systems. In Information fusion for cyber-security analytics (pp. 155-174). Springer, Cham.

[36] Ramaki, A. A., Khosravi-Farmad, M., & Bafghi, A. G. (2015, September). Real time alert correlation and prediction using Bayesian networks. In 2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC) (pp. 98-103). IEEE.

[37] Jiang, G., & Cybenko, G. (2004, June). Temporal and spatial distributed event correlation for network security. In Proceedings of the 2004 American Control Conference (Vol. 2, pp. 996-1001). IEEE.

[38] Anbarestani, R., Akbari, B., & Fathi, F. (2012, May). An iterative alert correlation method for extracting network intrusion scenarios. In 20th Iranian Conference on Electrical Engineering (ICEE2012) (pp. 684-689). IEEE.

[39] Wu, Z., Xiao, D., Xiao, M., & Peng, X. (2008, August). Using multilevel correlation in a unified platform of network security management: Design and implementation. In 2008 International Symposium on Electronic Commerce and Security (pp. 402-406). IEEE.

[40] Benferhat, S., Kenaza, T., & Mokhtari, A. (2008, July). A naive bayes approach for detecting coordinated attacks. In 2008 32nd Annual IEEE International Computer Software and Applications Conference (pp. 704-709). IEEE.

[41] Mathew, S., Shah, C., & Upadhyaya, S. (2005, March). An alert fusion framework for situation awareness of coordinated multistage attacks. In Third IEEE International Workshop on Information Assurance (IWIA'05) (pp. 95-104). IEEE.

[42] Li, Z., Zhang, A., Lei, J., & Wang, L. (2007, October). Real-time correlation of network security alerts. In IEEE International Conference on e-Business Engineering (ICEBE'07) (pp. 73-80). IEEE.

[43] Dain, O., & Cunningham, R. K. (2002). Fusing a heterogeneous alert stream into scenarios. In Applications of Data Mining in Computer Security (pp. 103-122). Springer, Boston, MA.

[44] Yu, D., & Frincke, D. (2004, June). A novel framework for alert correlation and understanding. In International Conference on Applied Cryptography and Network Security (pp. 452-466). Springer, Berlin, Heidelberg.

[45] Zhu, B., & Ghorbani, A. A. (2006). Alert correlation for extracting attack strategies. Int. J. Netw. Secur., 3(3), 244-258.

[46] Qin, X., & Lee, W. (2003, September). Statistical causality analysis of infosec alert data. In International Workshop on Recent Advances in Intrusion Detection (pp. 73-93). Springer, Berlin, Heidelberg.

[47] Moura, G. C., Sadre, R., & Pras, A. (2011, October). Internet bad neighborhoods: the spam case. In 2011 7th International Conference on Network and Service Management (pp. 1-8). IEEE.

[48] Moura, G. C., Sadre, R., Sperotto, A., & Pras, A. (2012, April). Internet bad neighborhoods aggregation. In 2012 IEEE Network Operations and Management Symposium (pp. 343-350). IEEE.

[49] Van Wynsberghe, A. L., & Moura, G. M. (2013). The embedded values and the example of Internet Security. Responsible Research and Innovation in ICT, Oxford, Technical Report, 1101.

[50] Moura, G. C., Sadre, R., & Pras, A. (2014). Bad neighborhoods on the internet. IEEE communications magazine, 52(7), 132-139.

[51] Bartos, V., Zadnik, M., Habib, S. M., & Vasilomanolakis, E. (2019). Network entity characterization and attack prediction. Future Generation Computer Systems, 97, 674-686.

[52] Zhang, J., Porras, P. A., & Ullrich, J. (2008, July). Highly predictive blacklisting. In USENIX security symposium (pp. 107-122).

[53] Soldo, F., Le, A., & Markopoulou, A. (2011). Blacklisting recommendation system: Using spatio-temporal patterns to predict future attacks. IEEE Journal on Selected Areas in Communications, 29(7), 1423-1437.

[54] Thomas, K., Amira, R., Ben-Yoash, A., Folger, O., Hardon, A., Berger, A., & Bailey, M. (2016, September). The abuse sharing economy: Understanding the limits of threat exchanges. In International Symposium on Research in Attacks, Intrusions, and Defenses (pp. 143-164). Springer, Cham.

[55] Wahid, A. (2013). Estimating the internet malicious host population while preserving privacy (Doctoral dissertation).

[56] V. Bartoš, M. Žádník, An analysis of correlations of intrusion alerts in an NREN, in: 19th International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD), IEEE, 2014, pp. 305–309.

[57] Dulaunoy, A., Wagener, G., Iklody, A., Mokaddem, S., & Wagner, C. (2018). An indicator scoring method for misp platforms. In The Networking Conference TNC (Vol. 18).

[58] Hernández, A., Sanchez, V., Sánchez, G., Pérez, H., Olivares, J., Toscano, K., ... & Martinez, V. (2016, March). Security attack prediction based on user sentiment analysis of Twitter data. In 2016 IEEE international conference on industrial technology (ICIT) (pp. 610-617). IEEE.

[59] Shu, K., Sliva, A., Sampson, J., & Liu, H. (2018, July). Understanding cyber attack behaviors with sentiment information on social media. In International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation (pp. 377-388). Springer, Cham.

[60] Shao, P., Lu, J., Wong, R. K., & Yang, W. (2016, November). A transparent learning approach for attack prediction based on user behavior analysis. In International Conference on Information and Communications Security (pp. 159-172). Springer, Cham.

[61] Gheyas, I. A., & Abdallah, A. E. (2016). Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis. Big Data Analytics, 1(1), 1-29.

[62] K. M. Salman, Ken Ferens, S. Siddiqui, "A Cognitive and Concurrent Cyber Kill Chain Model". Computer and Network Security Essentials, 2017. https://doi.org/10.1007/978-3-319-58424-9_34

[63] P. Holgado, V.A. Villagrá, L. Vázquez «Real-Time Multistep Attack Prediction Based on Hidden Markov Models". IEEE Transactions on Dependable and Secure Computing Vol. 17,pp. 134-147, 2017

[64] V. Paxon, S. Floyd, "Wide area traffic: the failure of Poisson modeling". IEEE/ACM Transactions on Networking, Vol. 3(3), pp. 226-244, 1995

[65] J. Jiang, S. Papavassiliou, "Enhancing network traffic prediction and anomaly detection via statistical network traffic separation and combination strategies". Computer Communications, Vol. 29 (10), pp. 1627-1638, 2006

[66] P. Singh, G. Dhiman, "A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches". Journal of Computational Science, DOI: 10.1016/j.jocs.2018.05.008, 2018.

[67] V. Pejovic, M. Musolesi, "Anticipatory Mobile Computing: A Survey of the State of the Art and Research Challenges". ACM Computing Surveys, 47(3) No. 47, 2015

[68] J. Maestre Vidal, M.A. Sotelo Monge, L.J.G. Villalba, "A novel pattern recognition system for detecting Android malware by analyzing suspicious boot sequences". Knowledge-Based Systems, Vol. 150, pp. 198-217, 2018

[69] E. Matania, L. Yoffe, "Some Things the Giant Could Learn from the Small: Unlearned Cyber Lessons for the US from Israel", Cyber Defence Review, pp. 101-109 2022.

[70] C.G. Akcora, Y. Li, Y.R. Gel, M. Kantarcioglu, "BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain". Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2020.

[71] S. Altalhi, A. Gutub, "A survey on predictions of cyber-attacks utilizing real-time twitter tracing recognition". Journal of Ambient Intelligence and Humanized Computing, Vol. 12 pp. 10209–10221, 2021

[72] M. Girdhar, J. Hong, H. Lee; T. Song, "Hidden Markov Models based Anomaly Correlations for the Cyber-Physical Security of EV Charging Stations".IEEE Transactions on Smart Grid, 2021. Doi: 10.1109/TSG.2021.3122106

[73] N. Sun, J. Zhang, P. Rimba, S. Gao, L.Y. Zhang, Y. Xiang, "Data-Driven Cybersecurity Incident Prediction: A Survey". IEEE Communications Surveys & Tutorials, Vol. 21(2),pp. 1744-1772, 2018.

[74] M. Husák, J. Komárková, E. Bou-Harb, P. Čeleda, "Survey of Attack Projection, Prediction, and Forecasting in Cyber Security". IEEE Communications Surveys & Tutorials, Vol. 21, pp. 640-660, 2018

[75] S. Altalhi, A. Gutub, "A survey on predictions of cyber-attacks utilizing real-time twitter tracing recognition". Journal of Ambient Intelligence and Humanized Computing, Vol. 12 pp. 10209–10221, 2021

[76] J. Maestre Vidal, M.A. Sotelo Monge, "Framework for anticipatory self-protective 5G environments". Proceedings of the 14th International Conference on Availability, Reliability and Security, Kent, UK, 2019

[77] D.J. Hand, "Classifier Technology and the Illusion of Progress", Statistical Science, Vol. 21, No. 1, pp. 1-14, February 2006.

[78] R. Elwell, R. Polikar, "Incremental Learning of Concept Drift in Nonstationary Environments", IEEE Transactions on Neural Networks, Vol. 22, No. 10, October 2011.

[79] J. Maestre Vidal, M.A. Sotelo Monge, S.M. Martínez Monterrubio, "Anomaly-Based Intrusion Detection: Adapting to Present and Forthcoming Communication Environments". Handbook of Research on Machine and Deep Learning Applications for Cyber Security, 195-217,2020. Doi: 10.4018/978-1-5225-9611-0.ch01

[80] B. Krawczyk, A. Cano, "Online ensemble learning with abstaining classifiers for drifting and noisy data streams". Applied Soft Computing, Vol. 38, 2018, pp. 677-692, 2018

[81] D. Brzezinski, J. Stefanowski, "Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm". IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, pp. 81-94, 2013.

[82] B. Krawczyk, A. Cano, "Online ensemble learning with abstaining classifiers for drifting and noisy data streams". Applied Soft Computing, Vol. 38, 2018, pp. 677-692, 2018

[83] J. Maestre Vidal, M.A. Sotelo Monge, S.M. Martínez Monterrubio,"EsPADA: enhanced payload analyzer for malware detection robust against adversarial threats". Future Generation Computer Systems, Vol. 104, pp. 159-173, 2020

[84] J. Maestre Vidal, A.L.S. Orozco, L.J.G. Villalba, "Online masquerade detection resistant to mimicry". Expert Systems with Applications, Vol. 61, pp. 162-180, 2016.

[85] B. Yener, T. Gai, "Cybersecurity in the Era of Data Science: Examining New Adversarial Models". IEEE Security & Privacy, Vol. 17 (6), pp. 46-53, 2019.

[86] Han, J., Kamber, M., & Pei, J. (July de 2011). Data Mining: Concepts and Techniques. Elsevier.

[87] A. Pereira, C. Thomas, "Challenges of Machine Learning Applied to Safety-Critical Cyber-Physical Systems". Machine Learning and Knowledge Extraction. 2020; 2(4):579-602. https://doi.org/10.3390/make2040031

[88] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, “Robust network traffic classification,” IEEE/ACM Trans. Netw., vol. 23, no. 4, pp. 1257–1270, Aug. 2015

[89] C. Sabottke, O. Suciu, and T. Dumitra¸s, “Vulnerability disclosure in the age of social media: Exploiting Twitter for predicting real-world exploits,” in Proc. USENIX Security Symp., 2015, pp. 1041–1056

[90] J. Maestre Vidal, M.A. Sotelo Monge, L.J.G. Villalba, "A novel pattern recognition system for detecting Android malware by analyzing suspicious boot sequences". Knowledge-Based Systems, Vol. 150, pp. 198-217, 2018

[91] A.A. Mihailović, J.C. Smolović, I. Radevic, N. Rasovic, N. Martinovic, "COVID-19 and Beyond: Employee Perceptions of the Efficiency of Teleworking and Its Cybersecurity Implications". Sustainability, 13(12), No. 6750, 2021

[92] J. Maestre Vidal, M.A. Sotelo Monge, S.M. Martinez Monterrubio, "Anomaly-Based Intrusion Detection: Adapting to Present and Forthcoming Communication Environments". Handbook of Research on Machine and Deep Learning Applications for Cyber Security, 2019. DOI: 10.4018/978-1-5225-9611-0.ch010

[93] Braei, M., & Wagner, S. (2020). Anomaly detection in univariate time-series: A survey on the state-of-the-art. arXiv preprint arXiv:2004.00433.

[94] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain and A. Nawaz, "HML-IDS: A Hybrid-Multilevel Anomaly Prediction Approach for Intrusion Detection in SCADA Systems". IEEE Access, vol. 7, pp. 89507-89521, 2019, doi: 10.1109/ACCESS.2019.2925838.

[95] H.S. Cheung et al. "A Machine Learning Approach for Predicting Bank Customer Behavior in the Banking Industry." Machine Learning and Cognitive Science Applications in Cyber Security, edited by Muhammad Salman Khan, IGI Global, 2019, pp. 57-83. http://doi:10.4018/978-1-5225-8100-0.ch002

[96] G. Andresini et al. , "INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection". Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security, pp. 111-122, 2021. Doi:10.1145/3474369.3486864

[97] G. Andresini et al. , "INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection". Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security, pp. 111-122, 2021. Doi:10.1145/3474369.3486864

[98] M.A. Sotelo Monge, J. Maestre Vidal, G. Martínez Pérez, "Detection of economic denial of sustainability (EDoS) threats in self-organizing networks". Computer Communications, Vol. 145, pp. 284-308, 2019.

[99] A. Sheikh, "Evading Intrusion Detection Systems, Firewalls, and Honeypots". Certified Ethical Hacker (CEH) Preparation Guide, pp. 153-164, 2021.

[100] J. Maestre Vidal, J.D. Mejía Castro, A.L.S. Orozco, L.J.G. Villalba,"Evolutions of evasion techniques aigainst network intrusion detection systems". Proc. ICIT 2013 The 6tm International conference on Information Technology, 2013.

[101] H. Fan, F. Zhang, R. Wang, X. Huang and Z. Li, "Semi-Supervised Time Series Classification by Temporal Relation Prediction," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 3545-3549, doi: 10.1109/ICASSP39728.2021.9413883.

[102] H. Cam, "Model-Guided Infection Prediction and Active Defense Using Context-Specific Cybersecurity Observations." Proc. IEEE Military Communications Conference (MILCOM). 2019.

[103] Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. Machine Learning, 109(2), 373-440.

[104] L. Bilge, Y. Han, M. Dell'Amico, "RiskTeller: Predicting the Risk of Cyber Incidents". In Proc. ACM SIGSAC Conference on Computer and Communications Security, pp. 1299-1311, 2017.

[105] A.E. Ibor, O.B. Okunoye, F.A. Oladeji, K.A. Abdulsalam, "Novel Hybrid Model for Intrusion Prediction on Cyber Physical Systems' Communication Networks based on Bio-inspired Deep Neural Network Structure". Journal of Information Security and Applications, Vol.65, March 2022. Doi: 10.1016/j.jisa.2021.103107

[106] Y. Wei, K.P. Chow, S.M. Yiu, "Insider threat prediction based on unsupervised anomaly detection scheme for proactive forensic investigation". Forensic Science International: Digital Investigation, Vol. 38, October 2021. doi: 10.1016/j.fsidi.2021.301126

[107] X.Zheng, P. Li, X. Hu, K. Yu, "Semi-supervised classification on data streams with recurring concept drift and concept evolution". Knowledge-Based Systems, Vol. 2015, 2021. Doi: 10.1016/j.knosys.2021.106749

[108] X. Bi, C. Zhang, X. Zhao, D. Li, Y. Sun, Y. Ma, "CODES: Efficient Incremental Semi-Supervised Classification Over Drifting and Evolving Social Streams". IEEE Access, Vol.8, pp. 14024-14036, 2020

[109] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[110] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). Mastering the game of go without human knowledge. nature, 550(7676), 354-359.

[111] Martinez, J. F., & Ipek, E. (2009). Dynamic multicore resource management: A machine learning approach. IEEE micro, 29(5), 8-17.

[112] Nguyen, Thanh Thi, and Vijay Janapa Reddi. "Deep reinforcement learning for cyber security." arXiv preprint arXiv:1906.05799 (2019).

[113] Caminero, Guillermo, Manuel Lopez-Martin, and Belen Carro. "Adversarial environment reinforcement learning algorithm for intrusion detection." Computer Networks 159 (2019): 96-109.

[114] Webber, Frederick Charles, and Gilbert Peterson. "Enhancing Multi-Objective Reinforcement Learning with Concept Drift." The Thirtieth International Flairs Conference. 2017.

[115] Goldberg, Dani, and Maja J. Matarić. "Maximizing reward in a non-stationary mobile robot environment." Autonomous Agents and Multi-Agent Systems 6.3 (2003): 287-316.

[116] Han, Yi, et al. "Reinforcement learning for autonomous defence in software-defined networking." International Conference on Decision and Game Theory for Security. Springer, Cham, 2018.

[117] Gammerman, A., Vovk, V., & Vapnik, V. (2013). Learning by transduction. arXiv preprint arXiv:1301.7375.

[118] Caruana, Rich. "Multitask learning." Machine learning 28.1 (1997): 41-75.

[119] Zhang, Yu, and Qiang Yang. "A survey on multi-task learning." arXiv preprint arXiv:1707.08114 (2017).

[120] Thrun, Sebastian, and Joseph O'Sullivan. "Discovering structure in multiple learning tasks: The TC algorithm." ICML. Vol. 96. 1996.

[121] Bickel, Steffen, et al. "Multi-task learning for HIV therapy screening." Proceedings of the 25th international conference on Machine learning. 2008.

[122] Alothman, Basil, Helge Janicke, and Suleiman Y. Yerima. "Class balanced similarity-based instance transfer learning for botnet family classification." International Conference on Discovery Science. Springer, Cham, 2018.

[123] Chadza, Timothy, Konstantinos G. Kyriakopoulos, and Sangarapillai Lambotharan. "Learning to learn sequential network attacks using hidden Markov models." IEEE Access 8 (2020): 134480-134497.

[124] Q. Li et al., 'A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection', IEEE Trans. Knowl. Data Eng., pp. 1–1, 2021, doi: 10.1109/TKDE.2021.3124599.

[125] M. Chen, H. V. Poor, W. Saad, and S. Cui, 'Wireless Communications for Collaborative Federated Learning', IEEE Communications Magazine, vol. 58, no. 12, pp. 48–54, Dec. 2020, doi: 10.1109/MCOM.001.2000397.

[126] L. Lyu, H. Yu, and Q. Yang, 'Threats to Federated Learning: A Survey', arXiv:2003.02133 [cs, stat], Mar. 2020, Accessed: Jan. 21, 2022. [Online]. Available: http://arxiv.org/abs/2003.02133.

[127] C. Dwork, "Differential privacy," in International Colloquium on Automata, Languages, and Programming, 2006, pp. 1–12.

[128] C. Zhao et al., "Secure Multi-Party Computation: Theory, practice and applications," Information Sciences, vol. 476, pp. 357–372, Feb. 2019, doi: 10.1016/j.ins.2018.10.024.

[129] C. Gentry, A fully homomorphic encryption scheme. Stanford University, 2009.

[130] Q. Li et al., "A survey on federated learning systems: vision, hype and reality for data privacy and protection," IEEE Transactions on Knowledge and Data Engineering, 2021.

[131] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. 2016. Communication-efficient learning of deep networks from decentralized data. arXiv:1602.05629.

[132] [1]H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated Learning with Matched Averaging," arXiv:2002.06440 [cs, stat], Feb. 2020, Accessed: Jan. 24, 2022. [Online]. Available: http://arxiv.org/abs/2002.06440

[133] Li, B., Wu, Y., Song, J., Lu, R., Li, T., & Zhao, L. (2020). DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems. IEEE Transactions on Industrial Informatics, 17(8), 5615-5624.

[134] Zhang, J., Pan, L., Han, Q. L., Chen, C., Wen, S., & Xiang, Y. (2021). Deep learning based attack detection for cyber-physical system cybersecurity: A survey. IEEE/CAA Journal of Automatica Sinica, 9(3), 377-391.

[135] [1]R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, 2015, pp. 1310–1321.

[136] Su, L., Xu, J., & Yang, P. (2021). Achieving statistical optimality of federated learning: Beyond stationary points. *arXiv preprint arXiv:2106.15216*.

[137] F. W. Bentrem, M. A. Corsello, and J. J. Palm, "Leveraging Sharing Communities to Achieve Federated Learning for Cybersecurity," arXiv:2104.11763 [cs], Apr. 2021, Accessed: Jan. 24, 2022. [Online]. Available: http://arxiv.org/abs/2104.11763

[138] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," IEEE Signal Process. Mag., vol. 37, no. 3, pp. 50–60, May 2020, doi: 10.1109/MSP.2020.2975749.

[139] G. Canonaco, A. Bergamasco, A. Mongelluzzo, and M. Roveri, "Adaptive Federated Learning in Presence of Concept Drift," in 2021 International Joint Conference on Neural Networks (IJCNN), Jul. 2021, pp. 1–7. doi: 10.1109/IJCNN52387.2021.9533710.

[140] Y. Chen, Z. Chai, Y. Cheng, and H. Rangwala, “Asynchronous Federated Learning for Sensor Data with Concept Drift,” arXiv:2109.00151 [cs], Aug. 2021, Accessed: Jan. 24, 2022. [Online]. Available: http://arxiv.org/abs/2109.00151

[141] G. Hinton, O. Vinyals, and J. Dean, ‘Distilling the knowledge in a neural network’, arXiv preprint arXiv:1503.02531, 2015.

[142] Q.-T. Le, ‘Multilayer Sparse Matrix Factorization’, ENS Paris Saclay, 2020.

[143] L. Zheng, E. Riccietti, and R. Gribonval, ‘Hierarchical Identifiability in Multi-layer Sparse Matrix Factorization’, arXiv preprint arXiv:2110.01230, 2021.

[144] S. Disabato and M. Roveri, ‘Tiny Machine Learning for Concept Drift’, arXiv:2107.14759 [cs], Jul. 2021, Accessed: Jan. 24, 2022. [Online]. Available: http://arxiv.org/abs/2107.14759

[145] H. Yu, T. Liu, J. Lu, and G. Zhang, ‘Automatic Learning to Detect Concept Drift’, arXiv:2105.01419 [cs], May 2021, Accessed: Jan. 24, 2022. [Online]. Available: http://arxiv.org/abs/2105.01419

[146] H. L. Hammer, A. Yazidi, and H. Rue, ‘A new quantile tracking algorithm using a generalized exponentially weighted average of observations’, Appl Intell, vol. 49, no. 4, pp. 1406–1420, Apr. 2019, doi: 10.1007/s10489-018-1335-7.

[147] C. O’Reilly, A. Gluhak, M.A. Imran, S. Rajasegarar, "Anomaly Detection in Wireless Sensor Networks in a Non-Stationary Environment", IEEE Communications Surveys & Tutorials, Vol. 16, No. 3, pp. 1413-1432, third quarter 2014.

[148] J. Maestre Vidal, M.A. Sotelo Monge, S.M. Martínez Monterrubio, "Anomaly-Based Intrusion Detection: Adapting to Present and Forthcoming Communication Environments". Handbook of Research on Machine and Deep Learning Applications for Cyber Security, 195-217,2020. Doi: 10.4018/978-1-5225-9611-0.ch01.

[149] C. Alippi, “Intelligence for Embedded Systems. Berlin”, Germany: Springer-Verlag, 2014.

[150] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019, doi: 10.1109/TKDE.2018.2876857.

[151] G. Ditzler, M. Roveri, C. Alippi, R. Polikar, "Learning in Nonstationary Environments: A Survey", IEEE Computational Intelligence Magazine, Vol. 10, Issue 4, pp. 12-25, November 2015.

[152] J. Liu, X. Li, W. Zhong, “Ambiguous decision trees for mining concept-drifting data streams”, Pattern Recognition Letters, Vol. 30, No. 15, pp. 1347-1355, November 2009.

[153] Cohen, G. Avrahami, M. Last, A. Kandel, “Info-fuzzy algorithms for mining dynamic data streams”, Applied Soft Computing, Vol. 8, No. 4, pp. 1283-1294, September 2008.

[154] Y. Ye, S. Squartini, F. Piazza, "Online sequential extreme learning machine in nonstationary environments”, Neurocomputing, vol. 116, pp. 94–101, September 2013.

[155] L. L. Minku, X. Yao, “DDD: A new ensemble approach for dealing with concept drift,” IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 4, pp. 619-633, April 2012.

[156] D. Brzezinski, J. Stephanowski, “Reacting to different types of concept drift: The accuracy updated ensemble algorithm,” IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, No. 1, pp. 81-94, January 2014.

[157] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizky, A. Bouchachia, "A survey on concept drift adaptation". ACM Computing Surveys, V. 46(4), N. 44, pp. 1-37, 2014.

[158] Dietterich, Thomas G., "Ensemble Methods in Machine Learning". Multiple Classifier Systems, pp. 1-15, 2000.

[159] Klinkenberg, Ralf & Joachims, Thorsten. (2000). Detecting Concept Drift with Support Vector Machines. Proceedings of ICML.

[160] Nishida, Kyosuke & Yamauchi, Koichiro. (2007). Detecting Concept Drift Using Statistical Testing. 264-269. 10.1007/978-3-540-75488-6_27.

[161] Page, E.S.. (1954). Continuous Inspection Schemes. Biometrika. 41. 100. 10.2307/2333009.

[162] Gama, João & Sebastião, Raquel & Rodrigues, Pedro. (2013). On evaluating stream learning algorithms. Machine Learning. 90. 317-346. 10.1007/s10994-012-5320-9.

[163] Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research, 1(1), 80.

[164] Kim, H., Kwon, H., & Kim, K. K. (2019). Modified cyber kill chain model for multimedia service environments. Multimedia Tools and Applications, 78(3), 3153-3170.

[165] MITRE ATT&CK. https://attack.mitre.org . Last seen 28/04/2022

[166] Pols, P., & van den Berg, J. (2017). The unified kill chain. CSA Thesis, Hague, 1-104.

[167] Three TAXII models. https://www.anomali.com/es/resources/what-are-stix-taxii . Last seen 13/01/2022.

[168] Watcher main view. https://securityonline.info/watcher-open-source-cybersecurity-threat-hunting-platform/ . Last seen 13/01/2022.

[169] Watcher Platform Architecture. https://securityonline.info/watcher-open-source-cybersecurity-threat-hunting-platform/ . Last seen 13/02/2022.

[170] Spiderfoot. https://www.spiderfoot.net/hx/ . Last seen 28/04/2022.

[171] Spiderfoot Scans View. https://www.spiderfoot.net/documentation/ . Last seen 13/01/2022.

[172] OpenCTI Main View. https://www.notion.so/Introduction-f99633ba66ba4ee3af1a4d832208dc99 . Last seen 13/01/2022.

[173] OpenCTI available connectors. https://www.notion.so/OpenCTI-Ecosystem-868329e9fb734fca89692b2ed6087e76 . Last seen 13/01/2022.

[174] OpenCTI Architecture. https://www.notion.so/Architecture-5ce8241eac7e4e24906249e9595314cd . Last Seen 13/01/2022.

[175] Csirtg-fm view. https://github.com/csirtgadgets/csirtg-fm-v1 . Last seen 13/01/2022.

[176] InfluxDB Diagram. https://www.influxdata.com/time-series-platform/telegraf/ . Last seen 13/01/2022.

[177] Kapacitor Integration with InfluxDB. https://www.influxdata.com/time-series-platform/kapacitor/ . Last seen 13/01/2022.

[178] Telegraf aggregator plugins. https://docs.influxdata.com/telegraf/v1.21/concepts/aggregator_processor_plugins/ . Last seen 14/01/2022.

[179] Chronograf – Infrastructure Monitoring. https://docs.influxdata.com/chronograf/v1.9/ . Last seen 13/01/2022.

[180] Chronograf – Alert Management. https://docs.influxdata.com/chronograf/v1.9/ . Last seen 13/01/2022.

[181] Chronograf – Database Management. https://docs.influxdata.com/chronograf/v1.9/ . Last seen 13/01/2022.

[182] Chronograf – Data Visualization. https://docs.influxdata.com/chronograf/v1.9/ . Last seen 13/01/2022.

[183] Splunk Validates Architecture. https://www.splunk.com/pdfs/technical-briefs/splunk-validated-architectures.pdf Last seen 06/05/2022.

[184] https://www.splunk.com/en_us/blog/tips-and-tricks/splunk-validated-architectures.html . Last seen 06/05/2022

[185] Ossec Architecture 1. Moukafih Nabil, Sabir Soukaina, Abdelmajid Lakbabi and Orhanou Ghizlane - SIEM selection criteria for an efficient contextual security May 2017 10.1109/ISNCC.2017.8072035 https://www.researchgate.net/figure/Integration-architecture-of-OSSEC-12_fig1_320649753

[186] AlienVault OSSIM https://www.xataka.com/empresas-y-economia/que-alienvault-que-hace-esta-empresa-ciberseguridad-at-t-haya-comprado Last seen 14/01/2022

[187] PatrOwl Technical Overview https://github.com/Patrowl/PatrowlDocs/raw/master/images/userguide/technical-overview.png Last seen 14/01/2022

[188] TheHive Project Workflow https://ciberseguridad.blog/un-trio-perfecto-con-thehive-cortex-y-misp/ Last seen 14/01/2022

[189] TheHive Alert Panel https://github.com/TheHive-Project/TheHive Last seen 14/01/2022

[190] TheHive Case Template Management https://docs.thehive-project.org/thehive/legacy/thehive3/admin/admin-guide/#2-case-template-management Last seen 14/01/2022

[191] TheHive 4 New Architecture https://blog.thehive-project.org/2020/02/05/under-the-mighty-hood-of-thehive-4/ Last seen 14/01/2022

[192] MISP views https://github.com/MISP/MISP Last seen 14/01/2022

[193] MISP Project https://www.misp-project.org/tools/ Last seen 04/05/2022

[194] Cortex view https://blog.thehive-project.org/2017/06/19/thehive-cortex-and-misp-how-they-all-fit-together/ Last seen 14/01/2022

[195] Alertflex high level design https://alertflex.github.io/doc/pages/architecture.html Last seen 14/01/2022

[196] Alertfelx low level design https://alertflex.github.io/doc/pages/architecture.html Last seen 14/01/2022

[197] Alertflex Management Console https://www.alertflex.org/ Last seen 14/01/2022

[198] Shuffle View https://github.com/frikky/Shuffle Last seen 14/01/2022

[199] Chainer https://docs.chainer.org/en/stable/ Last seen 03/05/2022

[200] M.F. Zhani, H. Elbiaze, "Analysis and Prediction of Real Network Traffic". Journal of Networks, Vol. 4, No. 9, pp. 855-865, 2009.

[201] R. Li, Z. Zhao, J. Zheng, C. Mei, Y. Cai, H. Zhang, "The Learning and Prediction of Application-Level Traffic Data in Cellular Networks". IEEE Transactions on Wireless Communications, Vol. 16 (6), pp. 3899-3912, 2017

[202] W. Liu, H. Zhang, Z. Ding, Q. Liu, C. Zhu, "A comprehensive active learning method for multiclass imbalanced data streams with concept drift". Knowledge-Based Systems, Vol. 215, No. 106778, 2021. Doi: 10.1016/j.knosys.2021.106778

[203] R.K. Polaganga, Q. Liang, "Self-Similarity and modeling of LTE/LTE-A data traffic". Measurement, Vol. 75, pp. 218-229, 2015

[204] T.S. Sethi, M. Kantardzic, "Handling adversarial concept drift in streaming data". Expert Systems with Applications, Vol. 97,pp. 18-40,2018.

[205] Zhang, Shuqin, Guangyao Bai, Hong Li, Peipei Liu, Minzhi Zhang, and Shujun Li. 2021. "Multi-Source Knowledge Reasoning for Data-Driven IoT Security" Sensors 21, no. 22: 7579. https://doi.org/10.3390/s21227579

[206] K. Apajalahti, E. Hyvönen, J. Niiranen, V. Räisänen, “Combining ontological modelling and probabilistic reasoning for network management”. Journal of Ambient Intelligence and Smart Environments, Vol. 9(1), pp. 63-76, January 2017.

[207] U. Das, V.C. Dev, V. Namboodiri, "On the effectiveness of data aggregation to manage network congestion in smart grid AMI". In Proc. of the 10th IEEE Conference on Power & Energy Society Innovative Smart Grid Technologies, Washington DC, US, 2018

[208] P. Porambage, M. Ylianttila, C. Schmitt, P. Kumar, A. Gurtiv, A.V. Vasilakos, "The Quest for Privacy in the Internet of Things". IEEE Cloud Computing, Vol 3(2), pp. 36-45, April 2016

[209] Polyglot Persistence https://en.wikipedia.org/wiki/Polyglot_persistence

[210] Attributes between different providers https://www.qlik.com/us/resource-library/cloud-data-lake-comparison-guide

[211] Tensorflow what if tool. https://www.tensorflow.org/tensorboard/what_if_tool . Last seen 28/04/2022

[212] API Umbrella. https://api-umbrella.readthedocs.io/en/latest/ . Last seen 28/04/2022.

[213] Gravitee. https://www.gravitee.io/ . Last seen 28/04/2022.

[214] APIman. https://www.apiman.io/latest/ . Last seen 28/04/2022.

[215] WSO2 API Manager. https://wso2.com/api-manager/ . Last seen 28/04/2022.

[216] Kong Enterprise. https://konghq.com/install#kong-community . Last seen 28/04/2022.

[217] Tyk. https://tyk.io/ . Last seen 28/04/2022.

[218] Fusio. https://www.fusio-project.org/ . Last seen 28/04/2022.

[219] Apigility. https://apigility.org/ . Last seen 28/04/2022.

[220] Barnum, S. (2012). Standardizing cyber threat intelligence information with the structured threat information expression (stix). Mitre Corporation, 11, 1-22.

[221] P. Katsumata, J. Hemenway, and W. Gavins, 'Cybersecurity risk management', in 2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE, Oct. 2010, pp. 890–895. doi: 10.1109/MILCOM.2010.5680181.

[222] C. Gordon and A. Arian, 'Threat and Decision Making', The Journal of Conflict Resolution, vol. 45, no. 2, pp. 196–215, 2001.

[223] E. Niewood, S. Lee, and G. Grant, 'Envisioning a New Command & Control (C2) Architecture for All-Domain Operations', NATIONAL SECURITY, p. 13, 2021.

[224] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, 'A survey on systems security metrics', ACM Computing Surveys (CSUR), vol. 49, no. 4, pp. 1–35, 2016.

[225] Y. Nikoloudakis et al., 'Towards a Machine Learning Based Situational Awareness Framework for Cybersecurity: An SDN Implementation', Sensors, vol. 21, no. 14, p. 4939, 2021.

[226] A. Oulasvirta, 'User interface design with combinatorial optimization', Computer, vol. 50, no. 1, pp. 40–47, 2017.

[227] M. Pantic, A. Nijholt, A. Pentland, and T. S. Huanag, 'Human-Centred Intelligent Human? Computer Interaction (HCI2): how far are we from attaining it?', International Journal of Autonomous and Adaptive Communications Systems, vol. 1, no. 2, pp. 168–187, 2008.

[228] S. Linda and K. K. Bharadwaj, 'A Decision Tree Based Context-Aware Recommender System', in International Conference on Intelligent Human Computer Interaction, 2018, pp. 293–305.

[229] J. Zhang et al., 'Why are adaptive methods good for attention models?', arXiv preprint arXiv:1912.03194, 2019.

[230] A. Fielder, E. Panaousis, P. Malacaria, C. Hankin, and F. Smeraldi, 'Decision support approaches for cyber security investment', Decision Support Systems, vol. 86, pp. 13–23, Jun. 2016, doi: 10.1016/j.dss.2016.02.012.

[231] M. Boddy, J. Gohde, T. Haigh, and S. Harp, 'Course of Action Generation for Cyber Security Using Classical Planning', p. 10.

[232] L. P. Rees, J. K. Deane, T. R. Rakes, and W. H. Baker, 'Decision support for Cybersecurity risk planning', Decision Support Systems, vol. 51, no. 3, pp. 493–505, Jun. 2011, doi: 10.1016/j.dss.2011.02.013.

[233] K. Chung, C. A. Kamhoua, K. A. Kwiat, Z. T. Kalbarczyk, and R. K. Iyer, 'Game theory with learning for cyber security monitoring', in 2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE), 2016, pp. 1–8.

[234] H. Hu, Y. Liu, C. Chen, H. Zhang, and Y. Liu, 'Optimal Decision Making Approach for Cyber Security Defense Using Evolutionary Game', IEEE Transactions on Network and Service Management, vol. 17, no. 3, pp. 1683–1700, Sep. 2020, doi: 10.1109/TNSM.2020.2995713.

[235] I. Atoum and A. Otoom, 'A Classification Scheme for Cybersecurity Models', IJSIA, vol. 11, no. 1, pp. 109–120, Jan. 2017, doi: 10.14257/ijsia.2017.11.1.10.

[236] K. Renaud and J. Ophoff, 'A cyber situational awareness model to predict the implementation of cyber security controls and precautions by SMEs', Organizational Cybersecurity Journal: Practice, Process and People, vol. 1, no. 1, pp. 24–46, Jan. 2021, doi: 10.1108/OCJ-03-2021-0004.

[237] F. L. Lewis and D. Vrabie, 'Reinforcement learning and adaptive dynamic programming for feedback control', Circuits Syst. Mag. IEEE, vol. 9, no. 3, pp. 32–50, 2009

[238] R. Kamalapurkar, P. Walters, J. Rosenfeld, and W. E. Dixon, Reinforcement learning for optimal feedback control: A Lyapunov-based approach. Springer, 2018.

[239] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, vol. 1. MIT press Cambridge, 1998.

[240] M. Salganicoff, ‘‘Density-adaptive learning and forgetting,’’ in Proc. 10th Int. Conf. Mach. Learn., 1993, pp. 276–283.

[241] B. C. da Silva, E. W. Basso, A. L. Bazzan, and P. M. Engel, ‘‘Dealing with non-stationary environments using context detection,’’ in Proc. 23rd Int. Conf. Mach. Learn., 2006, pp. 217–224.

[242] Webber, Frederick Charles, and Gilbert Peterson. "Enhancing Multi-Objective Reinforcement Learning with Concept Drift." The Thirtieth International Flairs Conference. 2017.

[243] T. T. Nguyen and V. J. Reddi, ‘Deep reinforcement learning for cyber security’, 2020.

[244] K. Chung, C. A. Kamhoua, K. A. Kwiat, Z. T. Kalbarczyk, and R. K. Iyer, ‘Game Theory with Learning for Cyber Security Monitoring’, in 2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE), Jan. 2016, pp. 1–8. doi: 10.1109/HASE.2016.48.

[245] M. Panfili, A. Giuseppi, A. Fiaschetti, H. B. Al-Jibreen, A. Pietrabissa, and F. Delli Priscoli, ‘A Game-Theoretical Approach to Cyber-Security of Critical Infrastructures Based on Multi-Agent Reinforcement Learning’, in 2018 26th Mediterranean Conference on Control and Automation (MED), Jun. 2018, pp. 460–465. doi: 10.1109/MED.2018.8442695.

[246] T. T. Nguyen and V. J. Reddi, ‘Deep Reinforcement Learning for Cyber Security’, IEEE Trans. Neural Netw. Learning Syst., pp. 1–17, 2021, doi: 10.1109/TNNLS.2021.3121870.

[247] Vue JS. https://vuejs.org/ . Last seen 29/04/2022

[248] A. G. Wilson, J. Yosinski, P. Simard, R. Caruana, and W. Herlands, ‘Proceedings of NIPS 2017 Symposium on Interpretable Machine Learning’, arXiv:1711.09889 [stat], Dec. 2017, Accessed: Jan. 11, 2022. [Online]. Available: http://arxiv.org/abs/1711.09889

[249] D. Gunning and D. Aha, ‘DARPA’s explainable artificial intelligence (XAI) program’, AI Magazine, vol. 40, no. 2, pp. 44–58, 2019.

[250] ‘国务院关于印发新一代人工智能发展规划的通知（国发〔2017〕35号）_政府信息公开专栏’. http://www.gov.cn/zhengce/content/2017-07/20/content_5211996.htm (accessed Jan. 13, 2022).

[251] ‘AI HLEG - steering group of the European AI Alliance | FUTURIUM | European Commission’. https://ec.europa.eu/futurium/en/european-ai-alliance/ai-hleg-steering-group-european-ai-alliance.html (accessed Jan. 13, 2022).

[252] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, “XAI—Explainable artificial intelligence,” Sci. Robot., vol. 4, no. 37, p. eaay7120, Dec. 2019, doi: 10.1126/scirobotics.aay7120.

[253] A. Barredo Arrieta et al., ‘Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI’, Information Fusion, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

[254] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, ‘A Survey of Methods for Explaining Black Box Models’, ACM Comput. Surv., vol. 51, no. 5, pp. 1–42, Sep. 2019, doi: 10.1145/3236009.

[255] M. Langer et al., ‘What do we want from Explainable Artificial Intelligence (XAI)? – A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research’, Artificial Intelligence, vol. 296, p. 103473, Jul. 2021, doi: 10.1016/j.artint.2021.103473.

[256] C. Meske, E. Bunde, J. Schneider, and M. Gersch, ‘Explainable Artificial Intelligence: Objectives, Stakeholders, and Future Research Opportunities’, Information Systems Management, vol. 39, no. 1, pp. 53–63, Jan. 2022, doi: 10.1080/10580530.2020.1849465.

[257] ‘Kubeflow’, GitHub. https://github.com/kubeflow (accessed Jan. 14, 2022).

[258] MLflow: A Machine Learning Lifecycle Platform. MLflow, 2022. Accessed: Jan. 14, 2022. [Online]. Available: https://github.com/mlflow/mlflow/mlflow

[259] Apache Airflow. The Apache Software Foundation, 2022. Accessed: Jan. 14, 2022. [Online]. Available: https://github.com/apache/airflow

[260] ‘iterative/dvc: Data Version Control | Git for Data & Models | ML Experiments Management’. https://github.com/iterative/dvc (accessed Jan. 14, 2022).

[261] feast-dev/feast. Feast, 2022. Accessed: Jan. 14, 2022. [Online]. Available: https://github.com/feast-dev/feast

[262] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1, No. 2). Cambridge: MIT press.

[263] R.M. Conroy, K. Pyörälä, A.P. Fitzgerald, S. Sans, A. Menotti, G. De Backer, D. De Bacquer, P. Ducimetière, P. Jousilahti, U. Keil, I. Njølstad, R.G. Oganov, T. Thomsen, H. Tunstall-Pedoe, A. Tverdal, H. Wedel, P. Whincup, L. Wilhelmsen, I.M. Graham, on behalf of the SCORE project group, Estimation of ten-year risk of fatal cardiovascular disease in Europe: the SCORE project, European Heart Journal, Volume 24, Issue 11, 1 June 2003, Pages 987–1003, https://doi.org/10.1016/S0195-668X(03)00114-3

[264] Barocas, S., & Selbst, A. D. (2016). Big data's disparate impact. Calif. L. Rev., 104, 671.

[265] Begoli, E., Bhattacharya, T., & Kusnezov, D. (2019). The need for uncertainty quantification in machine-assisted medical decision making. Nature Machine Intelligence, 1(1), 20-23.

[266] Natekar, P., Kori, A., & Krishnamurthi, G. (2020). Demystifying Brain Tumor Segmentation Networks: Interpretability and Uncertainty Analysis. Frontiers in Computational Neuroscience, 14, 6.

[267] Franke, U., & Brynielsson, J. (2014). Cyber situational awareness–a systematic review of the literature. Computers & security, 18-31.

[268] https://pypi.org/project/tf-explain/

[269] https://arxiv.org/abs/1706.03825

[270] https://arxiv.org/pdf/1703.01365.pdf

[271] https://captum.ai

[272] Ribeiro et al. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. https://arxiv.org/abs/1602.04938

[273] Ribeiro et al. 2018. Anchors: High-Precision Model-Agnostic Explanations. https://homes.cs.washington.edu/~marcotcr/aaai18.pdf

[274] https://github.com/marcotcr/lime

[275] https://pypi.org/project/shap/

[276] https://www.misp-project.org/compliance/ISO-IEC-27010/ [10/02/2022]

[277] https://www.boe.es/boe/dias/2021/01/28/pdfs/BOE-A-2021-1192.pdf [10/02/2022]

[278] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-150.pdf [04/05/2022]

[279] https://www.enisa.europa.eu/publications/good-practice-guide [04/05/2022]

[280] https://sansorg.egnyte.com/dl/FQSOYvuRrn [04/05/2022]

[281] https://www.misp-project.org/compliance/ISO-IEC-27010/ [04/05/2022]

[282] [KD99] J. Kruger and D. Dunning. "Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments." Journal of personality and social psychology 77.6 (1999): 1121.

[283] [DVC20] L. Dias, S. Valente and M. Correia. Go With the Flow: Clustering Dynamically-Defined NetFlow Features for Network Intrusion Detection with DynIDS. In Proceedings of the 19th IEEE International Symposium on Network Computing and Applications (NCA), Nov. 2020.

[284] [FDC20] T. Fernandes, L. Dias and M. Correia. C2BID: Cluster Change-Based Intrusion Detection. In Proceedings of Trustcom 2020, December 2020.

[285] [VF05] M. Verleysen and D. François, “The curse of dimensionality in data mining and time series prediction,” in International Work-Conference on Artificial Neural Networks, pp. 758–770, 2005.

[286] Bromander, S., Swimmer, M., Muller, L. P., Jøsang, A., Eian, M., Skjøtskift, G., & Borg, F. (2021). Investigating Sharing of Cyber Threat Intelligence and Proposing a New Data Model for Enabling Automation in Knowledge Representation and Exchange. *Digital Threats: Research and Practice (DTRAP)*, *3*(1), 1-22.