# VISUALIZATION AND MACHINE LEARNING FOR INTERACTIVE CYBER THREATS ANALYSIS IN CRITICAL INFRASTRUCTURES

R. AZZABI
CEA Tech Occitanie
31670 Labège, France
Email: radhouene.azzabi@cea.fr

C. GOUY-PAILLER
Institut LIST, CEA, Université Paris-Saclay
F-91120, Palaiseau, France

F. VALLEY
CEA Fontenay-aux-Roses
92265 Fontenay-aux-Roses Cedex, France

H. DUBOIS
CEA Tech Occitanie
31670 Labège, France

**Abstract**
Critical infrastructures are now under constant threat from cyber adversaries looking to exploit vulnerable systems and networks in order to achieve theirs objectives (denial of service, sabotage, or financial losses). As illustrated in the IAEA Nuclear Security Series No. 17 (Computer Security at Nuclear Facilities [1], fig. 7), the sophistication of attacks against computer networks is continuously growing disproportionately to the growth of defense technologies. Implementing computer security partly relies on strict levels of logging and monitoring of each entity of the system. As recalled in the IAEA reference manual, human errors and previously unknown threats have to be taken into account, to allow investigators and operators to take appropriate actions to mitigate risks. Therefore the supervision of nuclear industrial processes and related information systems is a mandatory component of nuclear facilities security. The resulting logs and activity monitoring signals are gathered in Security Operations Center (SOC). The role of SOCs is to complement security tools such as Intrusion Detection Systems or Antivirus, by using machine learning, rule-based or manual investigations approaches to detect suspicious behaviors that deviate from usual and specified activities (anomalies). These additional detections usually generate a large number of alerts, to be processed automatically or manually by an operator, who is in charge of investigating the severity of these alerts. This paper presents a software tool designed to help the operators in this task. It relies on a combination of visualization and machine learning techniques. Specifically we advocate that operators' investigations can be accelerated using advanced contextualization. Many tasks of the operator boil down to quickly deciding whether a user's activity can be considered as legitimate or further investigated. By contextualizing specific activities, which means characterizing the historical behavior of a user, understanding the historical usage profile of a domain name, or clustering users or domain names according to the historical patterns, operators are able to quickly decide whether an activity is considered as a threat or a legitimate activity. We show that such contextualization tasks often reduce to clustering problems in high dimensional spaces. Due to the streaming nature and the volume of the acquired data, a careful combination of dimensionality reduction and clustering techniques is necessary to deal with these data. The proposed software is demonstrated in a particular use case of investigation following an alert.

## 1. INTRODUCTION

Critical infrastructures are now under constant threat from cyber adversaries looking to exploit vulnerable systems and networks in order to achieve theirs objectives. As mentioned in [1], the risk in the computer security context can be evaluated as a combination of likelihood of an event and the severity of its consequences. Computer risks arise from the exploitation of potential vulnerabilities. While the risk assessment and management is out of the scope of this paper (see [2] for details), a key message of the risk assessment literature is that new vulnerabilities can always emerge and have to be treated with appropriate analysis tools by operators. Vulnerabilities have always been present in computers. Attackers' techniques have evolved from simple approaches involving huge knowledge about the target (password guessing) to advanced techniques targeting individuals using phishing attacks or distributed attacks tools. Attackers profiles cover a wide range of cases, both from the internal side, and the external side. As extensively detailed in [1], internal attackers can be covert agents, disgruntled employee, whose motivations range from theft of business information, economic gains or revenge. External attacker profile are diverse, ranging from recreational hackers, militant, terrorists or Nation States. Their motivations are also widely distributed, from economic gains to intelligence collection. As a consequence threat identification and characterization is a continuous and perpetual effort, human errors and previously unknown vulnerabilities have to be taken into account.

In the cybersecurity landscape, a number of companies have emerged in recent years. They address various aspects of computer security, *e.g.* data security and encryption, software security, web and mobile streams protection, computer and equipment network security, or industrial systems security. They usually rely on a combination of proprietary software technologies and expertise to address a wide range of customers. From the software side, major tools are intrusion detection systems, antivirus, and behavioral anomaly detection based on observed nominal computer behavior models. While on-device technologies can detect and mitigate a number of potential vulnerabilities in real time, computer network operations and maintenance still require large scale supervision in a time-deferred fashion. This approach is necessary to complement on-device technologies in case of undetected threats and vulnerabilities, or internal threats. Therefore numerous monitoring data is usually gathered in computer and equipment networks. The resulting logs and activity monitoring signals are gathered in Security Operations Center (SOC). The role of SOCs is to supplement security tools (Intrusion Detection Systems, Antivirus) by using machine learning, rule-based or manual investigations approaches to detect suspicious behaviors, which deviate from usual activities. These additional detections usually generate a large number of alerts, to be processed automatically or manually by an operator, who is in charge of investigating the severity of these alerts.

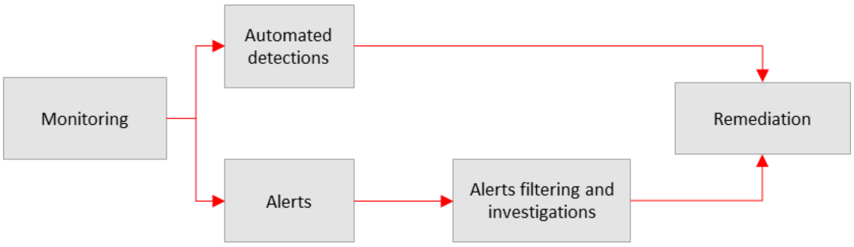


Figure 1: usual workflow of Security Operations Centers, based on monitoring data from the computer and equipment network. Based on data gathered from the computers or servers, alerts are raised, to be further investigated by a human expert. The goal of the operator is to quickly decide whether the alert necessitate an action or is a false alarm.

While huge expectations have been made on machine learning and artificial intelligence technologies to automatically detect, classify and mitigate threats, we observe that the creativity of attackers still surpasses supervised approaches, in which statistical learning is used to train a model to detect anomalies based on historical data. Additionally, these technologies bring their lot of false positive alerts. A wide range of existing monitored characteristics, based on packet flows, TOR connections, logins attempts, already exist in monitoring systems, resulting in numerous alerts to be treated by operators due to the intrinsic uncertainty of the decision. We thus advocate that machine learning and artificial intelligence techniques are rather of interest in operators' assistance in the task of alerts investigations. This paper presents a software tool designed to help the operators in this task. It relies on a combination of visualization and machine learning techniques. Specifically we advocate that operators' investigations can be accelerated using advanced contextualization. Many tasks of the operator boil down to quickly deciding whether a user's activity can be considered as legitimate or further investigated. By contextualizing specific activities, which means characterizing the historical behavior of a user, understanding the historical usage profile of a domain name, or clustering users or domain names according to the historical patterns, operators are able to quickly decide whether an activity should be considered as a threat or a legitimate activity. We show that such contextualization tasks often reduce to clustering problems in high dimensional spaces. Due to the streaming nature and the volume of the acquired data, a careful combination of dimensionality reduction and clustering techniques is necessary to deal with these data. The proposed software is demonstrated in a particular use case of investigation following a threat targeting a user.

The remaining of this paper is organized as follows: in section 2, we quickly present a state of the art of artificial intelligence and visualization in the context of cybersecurity. The proposed software tool of the paper is then presented in section 3. In section 4, a practical use case in cybersecurity is introduced. Section 5 is devoted to experimental results and qualitative assessments of the proposed framework.

## 2. STATE OF THE ART

Recent successes of artificial intelligence rely on statistical techniques, commonly denoted machine learning techniques. Among these techniques, we usually split the approaches into three main categories. First the supervised techniques [3] rely on big sets of labelled data to train a statistical model able to associate sensors inputs (logs, images, signals) with associated labels, as defined by the model designer. As an example a supervised technique is used in some advanced software security tools to classify exe files into malicious or not [4]. Another example is to train a neural network to detect algorithmically generated malicious domain names [5]. While supervised techniques have had tremendous successes in the recent years, their performances largely depend on the availability of a labelled dataset and the exhaustiveness of this dataset. Generalization ability of supervised algorithms (deep neural networks or general machine learning algorithms) is indeed still limited to already encountered situations. The variety of situations in the cybersecurity context, especially in the large scale monitoring context, and the scarcity of labelled datasets prevent supervised approaches from being widely used in computer network monitoring and anomaly detection. Second in situations where a machine learning algorithm can interact with the environment to acquire feedback and adapt its action policy, reinforcement learning [6] can be used. Reinforcement learning learns quite slowly by doing, thus it entails being able to let the agent interacting with the computer environment or build a complex virtual environment, mimicking observed signals available in real computer networks. This training method is thus costly and difficult to set up in the cybersecurity context. But this approach seems promising and substantial efforts are currently being made to bring reinforcement learning into the cybersecurity literature [7]. Third the unsupervised approach [8] is a machine learning technique aiming at discovering hidden relationships in datasets. Instead of categorizing, predicting or deciding, unsupervised approaches aim at discovering structures in data, or gathering similar objects. In the operator task of quickly deciding whether an alert has to be considered as a threat or a false positive, unsupervised learning techniques, *aka* representation learning techniques, and visualization make a perfect match to enhance the operator with contextualized information about users' activity or domain names.

In many cases, raw data results can be presented in several ways: textual, tabular and graphical. The way of presenting data depends on it properties: volume, distribution of values, quantity of entities which this data represents and type. To familiarize with a little block of information we can read it in textual format. To read several rows with properties of objects we can use tables, but if number of classes increased it would be hard to grasp it. Nevertheless, even with a small size of dataset for human it is hard to understand how one value from another differ, what are the relations between values, what weight one value have in total amount or in another value. The key to avoid these problem is to use graphical representation. A visual approach significantly facilitates the task and simplify the understanding of the results. One interesting aspect of visual representations is that they cause the analyst to explore and discover results, answer a questions and pose new ones. A human has the capability to look at a visual representation of data and see patterns. When we look at an image, some elements are detected immediately by the human visual system. No conscious attention is required to notice them. These elements are decorated with so-called pre-attentive visual properties. Visual properties are all the different ways of encoding data, such as shape, color, orientation, and so forth. The human visual system has its own rules. We can easily see patterns presented in certain ways, but if they are presented incorrectly, they become invisible.

Visualization of big data is a complex problem according to its diversity and heterogeneity. To achieve this task we need to combine several data manipulation tricks with advanced visualization and interaction techniques to provide a synthetic and simplified display for the user to interact and explore the data in a complex and multidimensional context. Several research works propose [9] a novel method for a synthetic multivariate network exploration and analysis approach based on an interactive selection of interest and a juxtaposed detail and height overview. In [10] the authors propose a pixel-oriented technique for visualizing large spatial datasets by representing the maximum data object as possible on the same screen at the same time by mapping each data value to a pixel of screen and arranging them adequately. Another work are focused on extended visualisation techniques. In [11][12], it is shown that the visualisation of a very large representation is simplified by using a complementary projection surface, interactive wall screens, immersive rooms, or immersive headsets. The use of these devices has made it possible to represent a large amount of data in a wide field of view and with a High Definition resolution.

Most research works propose novels techniques and approaches for complex data visualisation, but all of them follow the same process flow as mentioned by Ben Shneiderman [12] which is overview first, zoom and filter, then details on-demand. These techniques can be applied to the most security raw data analysis, records from networking workflow, system logs, and so on. We can separate security data into two broad categories: time-series and static data. Time-series data are data, which describe events in time. Static data is information that has no dependency on time associated with it. Also, any kind of information about the machines in environment or information about users can be considered static information. Nowadays visual exploratory analysis is an active research domain. Pin Ren [14] implements a visual metaphor "Flying Term" to visualize the dynamic nature of DNS application. The basic idea of this approach is to visualize target object (query string, IP address, and query type) inside a rectangular area, where coordinates of objects mapped using frequency data – normalized accumulated frequency and frequency distribution of the visualized subject over time. Another related work [15] introduced an approach of combining different graphs types to one chart (to reduce complexity of data and increase performance) and implements interactive part to allow user to investigate data on their query.

Because of the vast amount of log data provided by most security tools, such as firewalls, intrusion detection systems, DNS logs, user activities logging systems, and so on, security analysts need to choose the proper visualization tools and approaches in order to visualize the data story. The usage of artificial intelligence can simplify the analysis process complexity by adding mining for the analyzed data, pre-filtering and clustering steps for more accurate, assisted and rapid investigation. Visual analytics, is the proper approach for the security intelligence. It helps analysts to better understand what they are looking at, and helping data scientists understand what their algorithms have done.

## 3. PROPOSED FRAMEWORK

This work proposes a software tool, combining unsupervised [15] machine learning approaches and visualization, designed to provide alerts with adequate supplementary information. The rationale is that a majority of alerts are generated because of a lack of synthetic and comparative knowledge about the involved entities or the generated events. A prominent example is the detection of some secured https-based communication generated between a server from the monitored network to an external server. While this activity can appear as suspicious for non-specialized users, the communication historyenables a disambiguation of this kind of alerts. Therefore, machine-learning approaches have been developed to model and summarize behaviors and to detect similar behaviors (computer, group of computers, external requests…). A key point of the tool is to offer investigators with interactive 3D-based visualization, enabling simple and efficient data exploration with multi-level filtering and identification operations.

### 3.1. Software architecture

The main goals of our tool is to provide an interactive interface, for security analyst that allow data exploration, incidents visual investigation and anomalies detection. The tool relies on a client-server architecture.

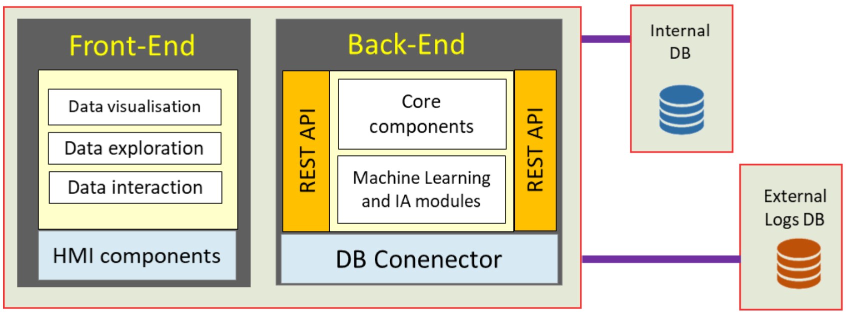


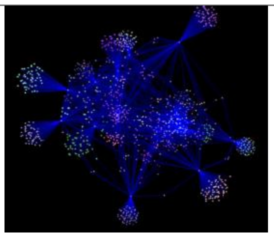
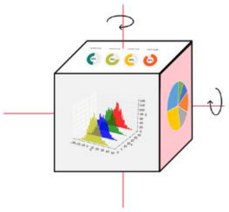
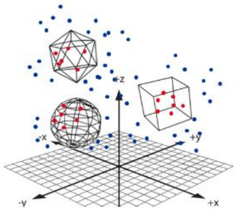
Figure 2: software architecture

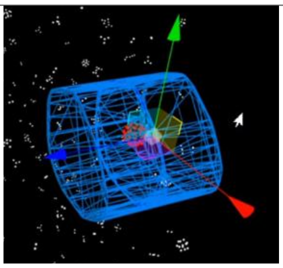
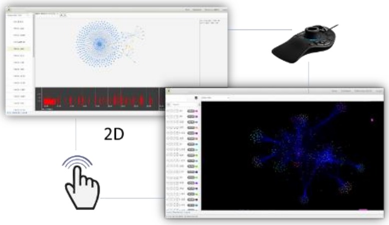
The front-end side is designed to be modular, components are integrated as plugins. The different interfaces are coded on JavaScript technologies (ReactJS). The server side is composed by four major blocks. The core block, the intelligence block, the database connector block and finally the REST API block. The tool, has its own database (MongoDB) on which we save configuration and IA results and another external DB on which we fetch security data logs (eg. Splunk, Elasticsearch …).

Different interactions flows are allowed. The front-end interface requests data results and configuration from the server-side via REST API. The server can notify the Front-end via a push notification system. When a new analysis is started, the server fetches raw data from the external database and call IA algorithms. Execution results can be saved to the local database or sent to the client-side in the case of real-time execution algorithm.

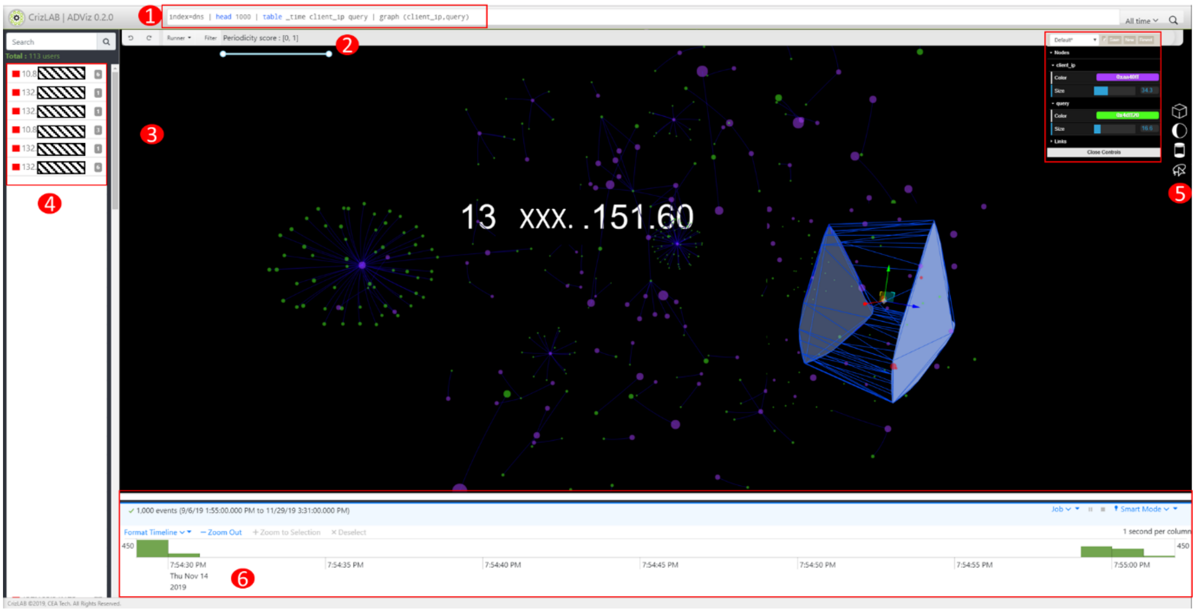
### 3.2. Visualization building blocks

The visualization methodology relies on visual analysis: for helping security analysts exploring security data, detecting anomalies and better understand machines learning algorithms results. It is based on (a) an intuitive design for large-scale and heterogeneous data visualization, (b) efficient data exploration on a mixed 2D/3D space (c) interactive filtering and graphical data selection (d) interactive annotation and on-demand machine learning algorithms customization. Bellow, we present, the visualization building blocks used.

| | | |
|---|---|---|
| **2D Representation** | Different 2D representations are included. User can select and customize the desired graphical representation type ( time-series, scatter-plot, bar-chart …) | |
| **3D Representation** | We introduce a 3D visualization word space, in order to visualize complexed representations. eg : Connected graph is computed based on analysis raw data results and points cloud positions are computed via a 3D force layout algorithm |  |
| **Extended visualization Area** | We propose a new visualization approach to extend the visualization areas by drawing additional information such as images, graphics or texts data.<br>Eg: for each selected point of interest, we can resume many information and show it, simultaneous, on different facets of 3D cube. |  |
| **3D points cloud selection techniques:** | We suggest in our tool a selection module to interact and select multiple points in the point cloud. We integrate three predefined types of volume selection, eg: box geometry, cylinder geometry and sphere geometry. |  |

| | | |
|---|---|---|
| | In addition to the different predefined volume selection geometries, we added a new way to select points cloud on 3D space by creating a free-form shape. We start by drawing a contour, converting the 3D world into a 2D. Then, user create the freeform volume where depth's is defined by mouse wheel. |  |
| **Multi-interaction level** | We integrate an interaction module combining different kinds of interaction, such as mouse and touch-hand interaction. We added, also, a joystick interaction possibility for better 3D navigation. |  |

The visualization building blocks, described below, are included in our tool as a separated modules. In the picture below, we present the final interface of our tool. It's composed by (1) a search bar to query the database and fetch results, (2) represent a control bar, for running online-offline functionalities, saving and loading results and also to filtering results. (3) Represent the 3D workspace, on which we plot results. (4) a right-side bar to show items list and anomalies flags if they exist. (5) a left-side menu, to select one off the four proposed selection type, predefined volume box, sphere, cylinder or the freeform volume selection. (6) a timeline to visualize timestamped data.



```
index=dns | head 1000 | table _time client_ip query | graph (client_ip,query)
```

✓ 1,000 events (9/6/19 1:55:00.000 PM to 11/29/19 3:31:00.000 PM)

Periodicity score : [0, 1]

Total : 113 users

Figure 3: ADViz : Anomalies detection & visualization

It should be noted that our tool, in this version, has been connected to Splunk as an external log database. So, some of visualization modules are included using splunk-sdk, e.g. : the timeViewer (6) and the searchBar (1). The searchBar has been modified in order to render search results as connected 3D graph

### 3.3. Algorithmic building blocks

Before going into the details of the three categories, one defines that an algorithm is called memoryless if and only if its output does not depend on the internal preexisting state of the algorithm. It means that a memoryless algorithm is operationally used as a functional black box, and that the call to the algorithm at different time, with the same input, will result in the same output. This is opposed to stateful algorithms, whose output will also depend on the internal state of the algorithms. Lightweight algorithms will denote processing techniques, which require less than 1 minute to run. Due to the required computation times of the majority of algorithms, there are three main possibilities of interaction between the user and the algorithms.

- Lightweight memoryless algorithms: in this case a direct communication between the user and the algorithm is established. Users are able to call the algorithm on specified data. An example of such a model of interaction is the call to a classification algorithm, which give a score to a domain name to detect algorithmically generated domain names. Basically, the user is selecting a domain name, calling the scoring algorithm and get the result in a fraction of second.
- Heavyweight algorithms: in a majority of cases, algorithms take longer than 1 minute to run on a specified input. In that case it is not reasonable to impose the user a waiting time while the algorithm is computing the output. Therefore we designed a CRON-like approach, which consists in pre-computing results. Thus the user only access the results of the computation from a database rather than waiting for the output to compute.
- Stage-separable algorithms: in many cases encountered in cybersecurity, the final result is computed in more than one stage. For example, dimensionality reduction techniques are often used in conjunction with clustering to get user or domain name grouping. While the first stage of the algorithm (dimensionality reduction) can be very expensive, its role is that further computations, such as distance computations or clustering become lightweight in the reduced space.

Various machine-learning algorithms are available in the tools, to enable efficient interactive exploration. Functionally, the goal of these algorithms are of three kinds:

1. Behavior modeling and summarization: the aggregate history of a unique computer is usually difficult to interpret. Approaches have been developed to perform disaggregation of activities into various classes, depending on the necessary level of implication by a human (e.g. periodic log activities have to be separated from other ones such as in [16]). Using pre-processed activities resulting from the disaggregation, events are then modelled through graph-based approaches [17]. Such graphs are designed to naturally represent successive events linked together. The tool then offer graph-based algorithms developed in this context to summarize and characterize behaviors, which are occurrences of random walks on events graphs.

2. Dimensionality reduction: the analysis of various activities generate high dimensional spaces. A prominent example is the analysis of domain names requests by computers of the internal network. On an open network, the number of unique requests (number of requested unique domain names) can be in the order of magnitude of a few millions, if observed within a few weeks. It is thus often desirable to perform dimensionality reduction on such spaces, to allow for subsequent manipulations within relatively small durations. An extensive study has been performed to make this stage generic. Based on the size of the resulting matrix, we integrated three distinct algorithms to perform dimensionality reduction. The first option is to use features hashing [18]. This technique uses a hashing function, allowing collisions between features, to reduce the dimensionality of the feature space. This technique is very efficient and allows users to control the dimension of the output features space. While this technique is very efficient for highly sparse data, one drawback is that the hashing trick does not preserve the structure of the original space. When the feature space dimension is very high, and the structure has to be preserved, we can use structured random projection [19]. This technique has been shown to preserve distances, as classical random projection, while the computations can be accelerated using the special structure of the projection matrix. Third when the feature space dimension is moderate, we use advanced techniques to find the reduced space [20]. This approach is more precise than the random approach for preserving structure, but is more computationally expansive.

3. Similarity search, top-k requests and clustering: once users and events have been correctly described and summarized, it is crucial to allow the operator to perform various comparison actions. It can be of interest to perform similarity search, which consists in finding entities, successive events, or group of computers similar to an object already identified. This can be of primary importance when specific behavior resulting from a targeted attack has been identified and the investigator wants to ensure that no computer has been also attacked or infected. Variants of such scenario include situations when the investigator wants to identify the first k users most similar to an identified one, or when he request users or events to be clustered in number of groups. In the proposed tool, we implemented k-means and hierarchical clustering, using Jaccard and Euclidean distances [3]. While this part

is difficult to evaluate, future requirements in terms of data volume and result precision could be improved with graph-based clustering approach techniques such as [21].

## 4. USE CASE STUDY

A crucial strength of the proposed approach relies on the diversity of the involved stakeholders. Specifically cybersecurity experts have been involved from the very beginning of the project. This has resulted in clear and realistic definitions of possible use case scenarios. We here describe one of these scenarios. Let one define the type of data, which could be remotely monitored in a Security Operations Center, managing security for a large network of computers:

- First DNS activity of the individual computers are recorded. This means that the time, source ip, and target domain name are recorded for every incoming requests.
- Second it is also a common practice in secured infrastructures to record and aggregate on-device (individual computers) Windows Security Logs Events [22], especially the ones recording process creation.

In the remaining of this paper, we will suppose that these two sources of data are recorded and aggregated in some database systems (typically splunk or elasticsearch). We will now describe an alert scenario and the subsequent operations, which have to be taken in the operations center. To simplify the description, let's imagine that an incident has been identified on one system of the network, *e.g.* execution of a malicious script. The goal of the investigation is to identify the particular DNS behavior of the offending machine to detect similar attacks on the network that could have been missed by antivirus or propagated. The manual procedure could follow five main steps:

1. DNS activity visualization of the identified targeted machine, during a short period of time around the vulnerability trigger;
2. Filtering: DNS data denoising to remove normal activities such as periodic system activity or activity-induced logs. This step would result in a significantly smaller amount of nodes to be visualized;
3. Node selection (suspicious domain names);
4. From the previous selection of domain names, perform a similarity search on the whole network to detect computers, which have had a similar DNS activity in an extended period of time (e.g. 7 days).
5. Correlate these DNS activities on the extended set of selected suspicious machines to analyze corresponding Active Directory activities to complete this specific malware detection.

This scenario has been a crucial guiding principle in the design and implementation of the proposed tool.

## 5. EXPERIMENTAL ASSESSMENT

Various actions have been launched to assess the ergonomic usability, visualization power, and efficiency of proposed software tool. First the ergonomic usability has been assessed following the successive steps of the previously defined scenario. Our cybersecurity expert has been tasked to accomplish data exploration inside its usual environment and using the proposed innovative software. While still under continuous improvements, the current version proved usable and easy to apprehend. Second a careful investigation of the visualization efficiency has been carried to test the limit of the 3D environment in terms of number of displayed objects. Successful experiments have been realized using as many as millions of distinct displayed objects. Third we conducted timing experiments to evaluate the limits of the machine learning-related computations. This last step have shown that some functionalities were unfeasible for real time investigations. This resulted in architectural adaptation of the software to allow for scheduled pre-computations, enable then real time interactions in the visualization environment.

The main goal of this tool is to ease and reduce investigation time on security incidents. Resolving non-trivial incidents implies digging into thousands of log events and the AI tools can easily scale down this number to a more reasonable one for a SOC operator. The 3D visualization is easy to read for the operator with large-scale data, although it requires some learning time, especially when it comes to node selections and navigation though a dense map. Without AI tools, 3D visualization is itself an efficient way to identify outlier, more powerful than a constrained 2D drawing. Dynamically filtering events based on what the AI considered a normal behavior (or at least already seen) is helpful, saving investigation time and tedious tasks. Flexibility is also greatly appreciated on these tools. Depending on the nature of the incident, the operator will have different needs of visualization. For example, investigating for a Command and Control (C&C) channel, the operator will carefully look for recurrent DNS requests and compare them with other endpoints. Whereas investigating on spear phishing, he will look for unusual DNS request filtered out by the behavior learned by AI.

## 6. CONCLUSION AND FUTURE WORK

This paper describes an innovative cybersecurity software tool designed to help expert investigators in their tasks of carefully reviewing network and machines activities in case of alerts and suspicious activities. Continuous interactions between cybersecurity, visualization and machine learning experts enabled very fruitful exchanges. While currently under evaluation, the software tool has already proved useful for reducing specific investigation tasks. In addition to powerful 3D visualization abilities, machine learning and AI techniques provide users with a way to enhance information provided on specific objects of the viewpoint, or highlight some information in the current scene. The design modularity of the architecture enables quick additions of new functionalities. Therefore the current evaluation step is the beginning of a continuous phase in which functionalities will be added on a regular basis to increase the usability of the software by operational cybersecurity experts.

## REFERENCES

[1] AIEA, “Computer Security at Nuclear Facilities,” 06-Sep-2016. [Online]. Available: https://www.iaea.org/publications/8691/computer-security-at-nuclear-facilities. [Accessed: 26-Nov-2019].

[2] International Electrotechnical Commission, “Information technology — Security techniques — Management of information and communications technology security — Part 1: Concepts and models for information and communications technology security management,” *ISO*, 2004. [Online]. Available: http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/90/39066.html. [Accessed: 27-Nov-2019].

[3] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Seconde. New-York, USA: Springer, 2009.

[4] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, “Microsoft Malware Classification Challenge,” *ArXiv180210135 Cs*, Feb. 2018.

[5] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, “Detecting Algorithmically Generated Malicious Domain Names,” in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, New York, NY, USA, 2010, pp. 48–61.

[6] J. Buckman, “Automation via Reinforcement Learning.” [Online]. Available: https://jacobbuckman.com/2019-09-23-automation-via-reinforcement-learning/. [Accessed: 26-Oct-2019].

[7] T. T. Nguyen and V. J. Reddi, “Deep Reinforcement Learning for Cyber Security,” *ArXiv190605799 Cs Stat*, Jun. 2019.

[8] R. O. Duda and P. E. Hart, *Pattern classification*, Second. Wiley-Interscience, 2000.

[9] Van den Elzen, Stef, and Jarke J. Van Wijk. "Multivariate network exploration and presentation: From detail to overview via selections and aggregations." IEEE Transactions on Visualization and Computer Graphics 20.12 (2014): 2310-2319.

[10] Keim, Daniel A. "Designing pixel-oriented visualization techniques: Theory and applications." IEEE Transactions on visualization and computer graphics 6.1 (2000): 59-78.

[11] Langner, Ricardo, Ulrike Kister, and Raimund Dachselt. "Multiple coordinated views at large displays for multiple users: Empirical findings on user behavior, movements, and distances." IEEE transactions on visualization and computer graphics 25.1 (2018): 608-618..

[12] Sicat, Ronell, et al. "Dxr: A toolkit for building immersive data visualizations." IEEE transactions on visualization and computer graphics 25.1 (2018): 715-725.

[13] Shneiderman, Ben. "The eyes have it: A task by data type taxonomy for information visualizations." Proceedings 1996 IEEE symposium on visual languages. IEEE, 1996.

[14] Ren, Pin, John Kristoff, and Bruce Gooch. "Visualizing DNS traffic." Proceedings of the 3rd international workshop on Visualization for computer security. ACM, 2006.

[15] M. Price-Williams, N. Heard, and M. Turcotte, “Detecting periodic subsequences in cyber security data,” *ArXiv170700640 Stat*, Jun. 2017.

[16] A. Paranjape, A. R. Benson, and J. Leskovec, “Motifs in Temporal Networks,” *Proc. Tenth ACM Int. Conf. Web Search Data Min. - WSDM 17*, pp. 601–610, 2017.

[17] K. Weinberger, A. Dasgupta, J. Attenberg, J. Langford, and A. Smola, “Feature Hashing for Large Scale Multitask Learning,” *ArXiv09022206 Cs*, Feb. 2010.

[18] K. Choromanski, F. Fagan, A. Morvan, C. Gouy-Pailler, J. Atif, and T. Sarlos, “TripleSpin - a generic compact paradigm for fast machine learning computations,” in *30th Annual Conference on Neural Information Processing Systems*, 2016.

[19] A. Morvan, A. Souloumiac, C. Gouy-Pailler, and J. Atif, “Streaming binary sketching based on subspace tracking and diagonal uniformization,” in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018, pp. 2421–2425.

[20] A. Morvan, K. Choromanski, C. Gouy-Pailler, and J. Atif, “Graph sketching-based Massive Data Clustering,” *SIAM Int. Conf. Data Min. SDM2018 Appear*, 2018.

[21] J. Talebi, A. Dehghantanha, and R. Mahmoud, “Introducing and Analysis of the Windows 8 Event Log for Forensic Purposes,” in *Computational Forensics*, Cham, 2015, pp. 145–162.

[22] N. Papernot, “A Marauder’s Map of Security and Privacy in Machine Learning: An Overview of Current and Future Research Directions for Making Machine Learning Secure and Private,” in *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, New York, NY, USA, 2018, pp. 1–1.