Monitoring Network Telescopes and Inferring Anomalous Traffic Through the Prediction of Probing Rates

Mehdi Zakroum, Jerome Francois, Isabelle Chrisment, Mounir Ghogho

Monitoring Network Telescopes and Inferring Anomalous Traffic Through the Prediction of Probing Rates

Mehdi Zakroum, *Member, IEEE*, Jérôme François, *Member, IEEE*,
Isabelle Chrisment, and Mounir Ghogho, *Fellow, IEEE*

***Abstract*—Network reconnaissance is the first step preceding a cyber-attack. Hence, monitoring the probing activities is imperative to help security practitioners enhancing their awareness about Internet's large-scale events or peculiar events targeting their network. In this paper, we present a framework for an improved and efficient monitoring of the probing activities targeting network telescopes. Particularly, we model the probing rates which are a good indicator for measuring the cyber-security risk targeting network services. The approach consists of first inferring groups of network ports sharing similar probing characteristics through a new affinity metric capturing both temporal and semantic similarities between ports. Then, sequences of probing rates targeting similar ports are used as inputs to stacked Long Short-Term Memory (LSTM) neural networks to predict probing rates 1 hour and 1 day in advance. Finally, we describe two monitoring indicators that use the prediction models to infer anomalous probing traffic and to raise early threat warnings. We show that LSTM networks can accurately predict probing rates, outperforming the non-stationary autoregressive model, and we demonstrate that the monitoring indicators are efficient in assessing the cyber-security risk related to vulnerability disclosure.**

***Index Terms*—network monitoring and measurements, network telescope, threat monitoring, security management, security situational awareness, artificial intelligence, machine learning, deep learning, unsupervised learning.**

I. Introduction

Nowadays, cyber-security is a major concern. According to [1], 68% of business leaders reported that cyber-security risks have increased. In 2020, the global total cost of data breaches is estimated to $3.86 million and it takes an average of 280 days to detect and contain a data breach [2]. Also, Cisco reported that the total number of DDoS attacks worldwide will reach 15.4 million attacks by 2023 [3]. To fine-tune their attacks, cyber-attackers conduct network reconnaissance through probing campaigns in their endeavor of discovering vulnerable services and security gaps through which they can infiltrate. Thus, monitoring the probing activities could help to detect an ongoing attack or to gain insight about imminent cyber threats. Network telescopes [4], also known as darknets or black holes, can serve this purpose. They consist of ranges of unassigned IP addresses that have never been declared to be hosting services. Hence, the traffic passively captured by a network telescope is suspect by nature and requires further analysis. This traffic is usually part of large-scale events happening on the Internet such as massive scan campaigns or back-scatter packets caught as a side-effect of spoofed denial-of-service attacks [5] [6] [7]. The darknet traffic may also include probing activities targeting specifically the organization owning the range of IP addresses used in the darknet. Therefore, monitoring such traffic is valuable to assess the attractiveness of network services from the attacker perspective [8].

In this research work, we present a complete framework to monitor the probing activities collected by network telescopes and to infer abnormal traffic targeting network services. This is done through the modeling of the probing rates - *i.e.* the number of received packets by the darknet targeting a network service during a period of time. First, this can be used to forecast the probing activity and pro-actively configure or reconfigure security functions, for example by rate-limiting given ports or deploying service-specific middleboxes. Second, wrong predictions can be a sign of a drastic change in the probing activity due to a new emerging threat, which in that case requires manual investigations by security experts. Most of the work in the literature focuses the attention on analyzing the relationship between the probing activities recorded by network telescopes and specific cyber incidents such as worms and vulnerabilities [9] [10] [11], or specific devices like IoTs [12] [13]. In contrast, our generic framework monitors the cyber-security risk related to network services by considering the historical observations of the darknet.

Hence, the goal of this research work is to answer the following questions:

- Q1. How to model the probing traffic at the service level?
- Q2. Are the probing rates predictable?
- Q3. How to leverage the probing rates predictions to infer anomalous traffic targeting specific network services?
- Q4. To which extent there is a relationship between probing activities and vulnerability disclosures?

Modeling the probing rates is in fact a challenging problem because of the diversity of the traffic captured by a network telescope, as for example misconfiguration packets, backscatter packets related to DoS attacks, worms and vulnerability probes. These different kinds of events are difficult to predict, and this is manifested in the probing rates' time series which

Mehdi Zakroum, Jérôme François (jerome.francois@loria.fr) and Isabelle Chrisment (isabelle.chrisment@loria.fr) are with Université de Lorraine and Inria Nancy-Grand Est, France.

Mehdi Zakroum (mehdi.zakroum@uir.ac.ma) and Mounir Ghogho (mounir.ghogho@uir.ac.ma) are with TICLab, International University of Rabat, Morocco.

exhibit non-stationarity in terms of the trend and the variance.

Our contributions and findings are as follows:

1) We design a novel affinity metric to capture temporal and semantic relationships between TCP/UDP ports based on the historical records of the probing traffic. Details are in Section V.
2) We define a method to model and predict the probing rates targeting the different network services. Our method uses stacked LSTM neural networks and relies on port affinities to infer the feature space of these predictive models through port clustering and port ranking. This answers question Q1 and details are in Section VI.
3) The performance of the prediction models are evaluated on the most targeted services using more than 300 days of probing traffic (more than 1.5 TB of data). We show that the probing rates are predictable with an average coefficient of determination R^2 ranging between 0.70 and 0.83, surpassing for most of the network services the performance of autoregressive-based models [14]. Also, we show that the information carried by the semantic similarity between ports contributes in defining an improved feature space which positively impacts the performance of the prediction model. This answers question Q2 and details about experimental evaluations are in Section VIII-B.
4) Finally, to address questions Q3 and Q4, the predictive model is leveraged to design two monitoring indicators that track anomalous traffic at the service level. We find that time series of the monitoring indicators and the vulnerabilities show significant relationships for different network services. To perform this evaluation, we designed scoring metrics relying on Dynamic Time Warping (DTW). More information is provided in Sections VII and VIII-C.

To the best of our knowledge, this research work is the first to describe a complete framework with new metrics for the real-time monitoring of large-scale probing activities.

The remainder of this paper is structured as follows. In the next section, we review the related work. In Section III, we describe the used data set. Section IV presents an overview of the proposed approach. In Section V, we describe our method for inferring the affinities between ports. Section VI presents the architecture and the hyper-parameters of the predictive models. In Section VII, we describe how the prediction models could be used to infer anomalies in the probing traffic. The evaluation is reported in Section VIII. Finally, we conclude by the lessons learned, the limitations of this work and future improvements.

II. Related Work

In the area of situational awareness in cyber-security, many studies leverage network telescope traffic to monitor large-scale cyber-security incidents and to model, infer and forecast cyber-threats. In [15], on a recent traffic captured by 3 typical darknets, the authors confirmed a known fact that the size of the darknet as well as the distribution of its IP blocks play a salient role in reflecting and monitoring cyber-security events. However, darknets are used to monitor only large-scale events targeting the entire IP address space or a sufficiently wide subset of it; they barely provide insight about localized scans and attack vectors targeting specific networks. To overcome the latter issue, authors in [8] used as a network telescope a Content Distribution Network (CDN) distributed over more than 13000 networks to monitor localized scanning activities. Even though our framework leverage typical network telescopes to monitor the darknet traffic, it could complementarily be used to monitor the traffic comprising localized cyber-attacks as long as the isolation of the probing activities from legitimate traffic is possible, as shown in [8].

In [16], the authors studied the impact of vulnerabilities on the volume of scans. They designed machine learning models to predict the impact of vulnerabilities, and they show that, by leveraging a set of features characterizing a vulnerability, they can accurately predict whether or not it will imply an increase in the volume of scans after its disclosure. However, vulnerabilities' descriptions and features are usually disclosed after the release of security patches and their impact is not always manifested by an increase in the probing rates. Instead of considering vulnerabilities as a starting point, our approach is rather to leverage the internal dynamics of the probing activities to infer anomalous probing traffic that could be related to external events, including vulnerabilities.

As for detecting anomalous traffic, in [6], the authors used darknet data to infer orchestrated probing campaigns, and thus to raise early warnings about imminent threats. Their approach relies on the prediction of missing values in probing flow time series extracted from the darknet traffic. They show that the latter predictive model is able to infer packets belonging to orchestrated probing events. In [17], to tackle the same issue, the authors rely on reducing the dimensionality of the big darknet data by using methods based on Fourier transforms and Kalman filtering applied to probing time series. We also exploit probing time series to reduce the dimensionality of data, however, our approach uses unsupervised learning techniques based on a new affinity metric measuring the temporal and the semantic similarities between ports. Other approaches of inferring anomalous traffic include the analysis of the probing behaviors of network probes by modeling their scanning activities using graphs [18], the clustering of latent representations of the destination ports present in a network flow [19], or detecting infected devices serving as point of cyber-attacks (e.g. botnets) [20]. In this work, instead of identifying anomalous traffic by analyzing the behavior of the source of scans, we rather track the probing rates at the service level at a larger scale. This allows us to measure the cyber-security risk of emerging cyber-threats by monitoring the evolution of the large-scale probing activities in time at the service level.

To find the relationship between ports, the authors in [21] designed a semantic similarity metric based on the behavioral patterns of network probers expressed in the form of a graph. However, an essential aspect in inferring the similarity between ports is the measure of the temporal correlations between network services at a higher granularity. In this paper, the similarity between ports is inferred by leveraging both the

TABLE I: Descriptive Statistics of the Traffic Captured by our Darknet

Designation	Value
Total number of packets	≈ 80 billion
TCP & UDP distribution	TCP: 94% / UDP: 5% / 1% Other
Avg. daily packet rate	≈ 60 million packets per day
Avg. daily unique source IP addr.	≈ 620000 IP addr. per day
Avg. daily darknet IP addr. hits	8192 IP addr. per day

temporal and the semantic aspects of the probing activities targeting ports, resulting in the definition of an improved feature space of the predictive models.

In relation with predictive models leveraging darknet traffic, the authors in [14] used the vector autoregressive model to predict the probing rates at the port level. Their approach for training the model consists in adapting the learning process to overcome the issue of the non-stationarity of the autoregressive model's parameters over time. To reduce the dimensionality of the model's feature space, the authors used recursive feature elimination to select most correlated ports. Even though the model produces good predictions, the autoregressive order, the feature vector and the model parameters are all learned online at each time step, which requires significant computational resources. In contrast, we propose to use LSTM recurrent neural networks, trained only once, with features initially learned offline (using a pre-clustering), which considerably reduces the computation complexity.

III. Network Telescope Traffic Data

A network telescope is a network sensor that consists of ranges of IP addresses not replying to any incoming traffic. These IP addresses are never declared to be hosting network services. Thus, any recorded traffic is considered suspicious and requires further analysis.

Our data set consists of network traffic collected by two /20 network telescopes (8192 IP addresses) hosted in France and Japan. The captured traffic includes headers of the incoming TCP and UDP packets, with no payload. For each packet, the recorded information are the timestamp, the protocol, the source and the destination IP addresses, the source and the destination ports, and the flags in the case of TCP packets. The probing activities were recorded for a period of 3.5 years starting from January 2017. Table I summarizes descriptive statistics of the data set.

Fig. 1 shows the time series of probing rates from January 2017 to September 2020. As shown, the captured traffic is constantly increasing over time and its variance tends to increase as well.

Fig. 2 shows the cumulative percentage of the received traffic by number of ports. We observe that 95% of the traffic has targeted 49172 ports and 50% of the traffic has targeted only 1074 ports, whereas the remaining ports have received negligible amount of probes. To reduce the computation complexity, we consider in this study the 1074 most targeted ports. In the remaining, $\mathcal{D} = \{p_1, p_2, \ldots, p_{1074}\}$ denotes the set of the most targeted ports. It is worth mentioning that to evaluate the predictive models, the probing traffic is split into two subsets: 75% for the training set, and 25% for the testing set which is approximately the last 300 days of traffic.

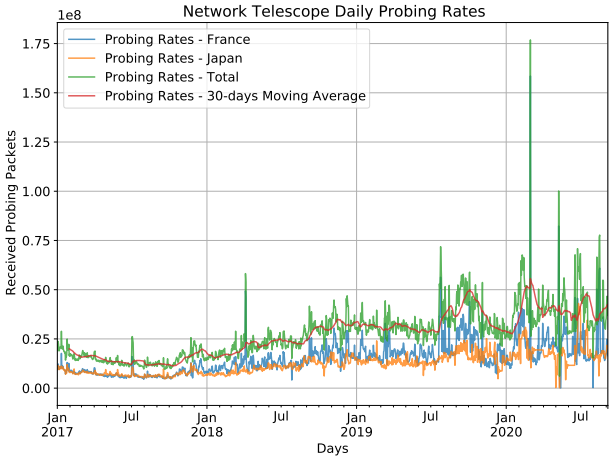


Fig. 1: Network telescope traffic aggregated in 1 day bins recorded from Jan. 2017 until Sept. 2020. The blue and the orange time series represent the traffic recorded by the network telescopes deployed in France and Japan, respectively. The green line is the sum of both and the red line is the moving average of the latter sum using a time window of 30 days.

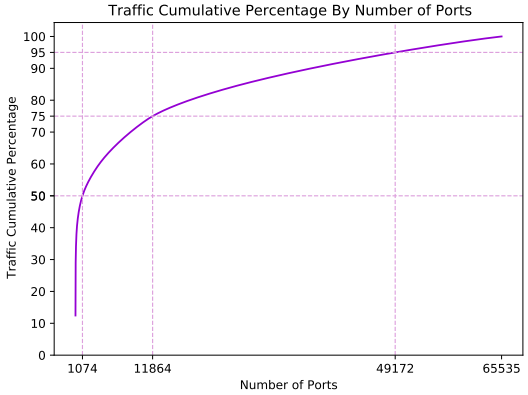


Fig. 2: Traffic cumulative percentage by number of ports

IV. Approach Overview

Fig. 3 summarizes the approach to model and predict the probing rates. In the following, we briefly describe each component of the framework, and more details are provided in the sections given in the Fig. 3's annotations.

The input of the framework is the darknet traffic detailed in Section III. From the latter, we infer the ports sharing similar probing characteristics. This allows us to refine the input of the predictive model by considering only ports having semantically and temporally correlated probing activities. This is done through the design of a new affinity metric that captures two salient aspects of relationships between ports:

- The temporal probing similarities expressed as the cross-correlations between the probing rates' times series of the ports. This allows us to infer ensembles of ports targeted by synchronized probing activities.
- The semantic similarities extrapolated from the probing activities of network probers that scan ports sequentially. This reveals if some ports host the same type of service (e.g. remote access services) or services often co-located together (e.g. ssh, http and https) [21]. The effectiveness

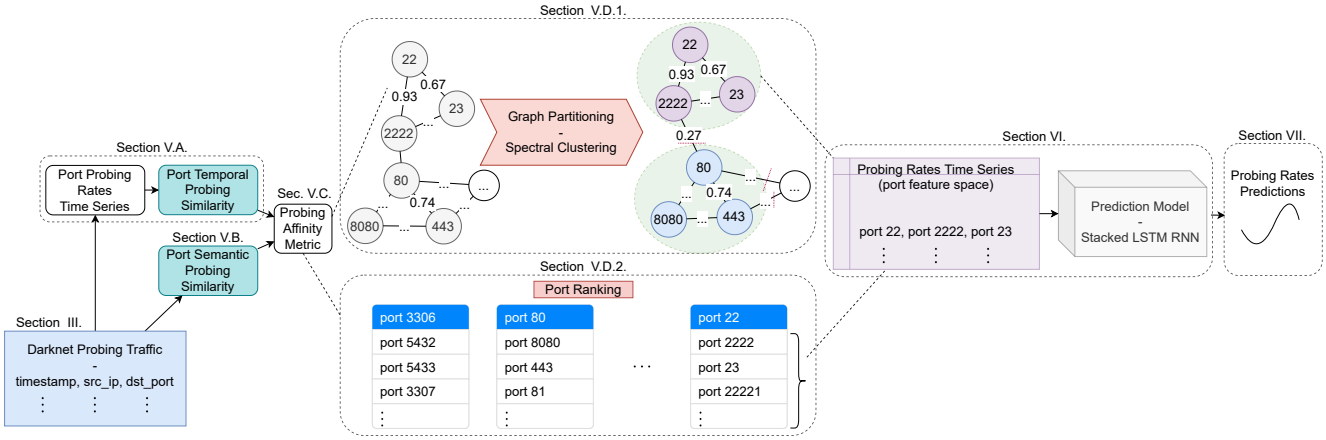


Fig. 3: Approach Overview of the Monitoring Framework.
The different components of the framework are annotated with the corresponding sections providing detailed explanations.

of the semantic similarity on the performance of the predictive models is evaluated in Section VIII.

Next, to reduce the dimensionality of the model's feature space and to alleviate the noise introduced by non-informative features, we leverage and evaluate two approaches:

1) port ranking in which ports are ranked based on their affinity score with the target port (*i.e.* the port for which the time series will be predicted) and the most correlated ones are selected,
2) port clustering in which spectral clustering is used to partition a large graph of ports weighted by the affinity score.

These two approaches are detailed in Section V-D.

Finally, stacked LSTM neural networks are trained to predict the probing rates. The model takes as input the probing activities history of the ports in affinity with the target port, and outputs expected probing rates of the target port.

In the remainder, we present in details the components of the darknet monitoring framework.

V. Network Port Affinities

In this section, our goal is to find clusters of ports having similar probing activities. Finding these groups of ports helps reducing the dimensionality of the prediction models' input space, thus alleviating the noise introduced by uncorrelated features. The rationale is that attackers usually perform an orchestrated scan targeting several ports rather than a single one. Thus, to predict the activity on a given port, it will be more efficient to focus on these related ports rather than all existing ports. Another application could be for instance to dynamically adapt security functions of a set of semantically similar ports after observing an abnormal traffic received by any of them.

In order to measure the similarity between ports, we design an affinity metric that captures two main aspects: the temporal correlations between port probing rates' time series and the semantic relationship between ports.

A. Temporal Probing Similarity

To find the temporal correlations between ports, we use port probing rates time series extracted from the traffic targeting the entire darknet IP address space. To reduce the computation complexity, we consider the probing rates' time series of the ports receiving 50% of the cumulative global traffic (cf. Fig. 2), denoted by $\mathbf{X} = \{X^{(p_i)}\}_{p_i \in \mathcal{D}}$. Each record of $\mathbf{X}$ is a vector of the number of packets received during the same time interval. This time interval is user-defined, and in our experiments, we consider time intervals of 1 hour and 1 day.

To measure the temporal probing similarity between two ports p_i and p_j, we compute the correlations between their probing rates' time series $X^{(p_i)}$ and $X^{(p_j)}$, after introducing lags $l \in [-L, L]$. Then, we consider the maximum correlation in absolute value. The user-defined time interval $[-L, L]$ represents the amount of periodic patterns to be captured in the correlation. For the 1-hour resolution time series, we set $L = 168$ which corresponds to 1 week, and for 1-day resolution time series, we chose $L = 31$ which corresponds to 1 month. Formally, the temporal probing similarity is defined by:

$$\mathcal{T}_l(p_i, p_j) = \max_{-L \leq l \leq L} |\rho_l(p_i, p_j)| \in [0, 1], \tag{1}$$

where $\rho_l(p_i, p_j)$ is the unbiased Pearson correlation coefficient between the probing rates' time series of the ports p_i and p_j after introducing a lag l, defined by:

$$\rho_l(p_i, p_j) = \frac{1}{N - |l|} \sum_{k=\max(0,l)}^{\min(N, N+l)-1} \frac{X_k^{(p_i)} - \mu_{p_i}}{\sigma_{p_i}} \cdot \frac{X_{k-l}^{(p_j)} - \mu_{p_j}}{\sigma_{p_j}}, \tag{2}$$

where N is the length of the time series, $X_k^{(\bullet)}$ is the probing rate at the k^{th} time step, and $\mu_\bullet$ and $\sigma_\bullet$ are respectively the mean and the standard deviation of the time series $X^{(\bullet)}$.

A value of $\mathcal{T}_l(p_i, p_j)$ close to 1 means that, in average, the probing traffic received by the port p_i is highly correlated with the probing traffic received by the port p_j before or after a delay of l time steps.

B. Semantic Probing Similarity

One of the main purposes of ports is to conventionally identify network services during a network transport session.

The Internet Assigned Numbers Authority (IANA) designed a registry of services and port numbers in which three categories of ports are distinguished: *System Ports* assigned to the range $[0, 1023]$, *User Ports* assigned to the range $[1024, 49151]$, and the remaining range $[49152, 65535]$ is reserved for *Dynamic Ports* for private usage. Software and service providers usually choose port numbers from the first two ranges as default network gates to their deployed services.

To discover vulnerable services, network attackers generally follow two probing approaches: (i) wide range port probing which consists of scanning a large range of ports against a database of vulnerabilities, or (ii) target-specific service probing which consists of scanning a set of ports against one or many predetermined vulnerabilities. However, many end-users tend to use unconventional or alternative ports. This motivates network attackers to scan ports following strategies that take into consideration such alteration and obfuscation. For instance, the end-user may deploy the FTP service in port 22 (the SSH port). A network prober aiming to attack the insecure FTP service will first start the probing activity by scanning the port 21. Not receiving a reply, the attacker may assume that the FTP service (or its variant SFTP) is intentionally deployed by the victim in port 22 and subsequently scan the port 22 as well. In such case, the corresponding traffic initially intending to probe the FTP service is falsely recorded by the darknet as an SSH traffic. Hence, if it happens to be a common practice to scan ports 21 and 22, then they should be in the same feature space of the prediction model. That is, they should both contribute in the prediction of the probing rate of the target network service.

To overcome such issues, similar approach as [21] is adopted which consists of extracting and analyzing ports that are scanned consecutively by network probers. Since there are approximately 750 million source IP addresses (network probers) in our data set, and to reduce the computation complexity, we select randomly (uniformly) 5% of the IP addresses from the pool of the source IP address that sent at least two packets to at least two destination ports. Besides, we only take into account the IP addresses having a probing activity spanning in at most 24 hours, which is a large enough DHCP lease time [22]. Two ports are considered semantically similar if there is a significant amount of the IP addresses that sequentially scanned these two ports.

Formally, we consider the following:

- $\{D_1, D_2, \ldots, D_m\}$: the 24 hours traffic subsets of our data set,
- A: the set of selected source IP addresses as explained above,
- $p_i \overset{ip}{\longleftrightarrow} p_j$: the event representing an IP address ip having sent a probing packet to port p_i followed by a packet to port p_j, or vice-versa.

The semantic similarity between two ports p_i and p_j is defined as the average number of IP addresses that scan the pair of port numbers over time periods $\{D_i\}_{i\in[1,m]}$:

$$\mathcal{S}(p_i, p_j) = \frac{1}{m} \sum_{k=1}^{m} \sum_{\text{ip}\in A} s_{\text{ip}}^{(D_k)}(p_i, p_j), \tag{3}$$

where

$$s_{\text{ip}}^{(D_k)}(p_i, p_j) = \begin{cases} 1 & \text{if } p_i \overset{ip}{\longleftrightarrow} p_j \text{ in probing traffic } D_k \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

It is noteworthy that this metric is not bound to the interval $[0, 1]$ like the temporal similarity metric, and it depends on the number of selected IP addresses. To align it with the scale of the temporal probing similarity, we address the skewness of large values by log-scaling them and then by applying a min-max normalization:

$$\mathcal{S}_l(p_i, p_j) = \log(1 + \mathcal{S}(p_i, p_j)) \tag{5}$$

$$\hat{\mathcal{S}}(p_i, p_j) = \frac{\mathcal{S}_l(p_i, p_j) - \min\limits_{p_i,p_j} \mathcal{S}_l(p_i, p_j)}{\max\limits_{p_i,p_j} \mathcal{S}_l(p_i, p_j) - \min\limits_{p_i,p_j} \mathcal{S}_l(p_i, p_j)} \in [0, 1] \tag{6}$$

A score $\hat{\mathcal{S}}(p_i, p_j)$ close to 1 means that the port p_i and p_j are highly semantically similar, *i.e.* they are consecutively scanned by a significant amount of network probers.

C. Probing Affinity Metric

The affinity metric between two ports is defined as the harmonic mean of the temporal and the semantic probing similarities:

$$\mathcal{A}(p_i, p_j) = 2 \cdot \frac{\mathcal{T}(p_i, p_j) \times \hat{\mathcal{S}}(p_i, p_j)}{\mathcal{T}(p_i, p_j) + \hat{\mathcal{S}}(p_i, p_j)} \tag{7}$$

In this case, the harmonic mean is preferred over the arithmetic and the geometric means, because it penalizes port affinities having either a low temporal or a low semantic similarity score, and thus we consider affinities having both scores high. This has the effect to reduce the noise for a better port ranking and for an enhanced clustering separability.

D. Port Clustering and Ranking

To infer the feature space of the predictive model, we rely on the defined affinity metric to find groups of similar ports using two methods: (i) spectral clustering which constructs mutually exclusive groups of ports, and (ii) port ranking which allows the reuse of input ports for different target ports.

1) Spectral Clustering: The affinity matrix $A \in \mathbb{R}^{|\mathcal{D}|\times|\mathcal{D}|}$ is the input of the clustering algorithm, where the elements $A_{ij} = 1 - \mathcal{A}(p_i, p_j)$ if $i \neq j$, and $A_{ii} = 0$, p_i and p_j being two ports belonging to the set of destination ports $\mathcal{D}$. This matrix is symmetric and represents an undirected graph. Each vertex of this graph is a port belonging to $\mathcal{D}$ and an edge exists between two ports p_i and p_j if $A_{ij} > 0$, and in such case, its weight is A_{ij}. Therefore, partitioning the latter graph allows us to identify clusters of ports sharing similar probing characteristics. There exist many clustering algorithms to perform graph partitioning. One of the most robust algorithms is *Spectral Clustering* [23] widely used for this type of tasks. An advantage of the spectral clustering algorithm is that it does not make the strong assumption of convexity on the form of the clusters. Also, the coordinates of data points are not required; the distances between them are sufficient to partition the graph.

To determine the optimal number of clusters k (*i.e.* the optimal graph cuts), we use the silhouette method. This metric gives an insight about the quality of the clustering separation by measuring how similar each data point is to its own cluster comparing to those of the nearest neighboring cluster. Formally, for a given port $p_i \in \mathcal{D}$ belonging to a cluster C_h, the silhouette score is defined as follows:

$$s(p_i) = \frac{b(p_i) - a(p_i)}{\max(b(p_i), a(p_i))}, \tag{8}$$

where a is the average intra-cluster distance:

$$a(p_i) = \frac{1}{|C_h| - 1} \sum_{p_j \in C_h, p_j \neq p_i} \mathcal{A}(p_i, p_j) \tag{9}$$

and b is the average nearest-cluster distance:

$$b(p_i) = \min_{l \neq h} \frac{1}{|C_l|} \sum_{p_j \in C_l} \mathcal{A}(p_i, p_j). \tag{10}$$

Then, the optimal number of clusters k is the one that maximizes the average silhouette score:

$$k = \underset{2 \leq k_n \leq \frac{|\mathcal{D}|}{2}}{\operatorname{argmax}} \frac{1}{k_n} \sum_{1 \leq h \leq k_n} \left(\frac{1}{|C_h|} \sum_{p_i \in C_h} s(p_i) \right) \tag{11}$$

2) Port Ranking: As an alternative method for inferring the feature space of the prediction model for a given target port, we can simply consider the most similar ports, *i.e.* having the affinity score higher than a threshold τ. We choose τ giving the best coefficient of determination R^2 score through cross-validation by varying it between the mean and the maximum affinity scores, with steps equal to the standard deviation.

Formally, using port ranking, the port feature space of a given target port $p_i \in \mathcal{D}$ is:

$$C_\tau(p_i) = \{p_j \in \mathcal{D} | \mathcal{A}(p_i, p_j) > \tau\}, \tag{12}$$

and τ varies in:

$$\{\mu_\mathcal{A}, \mu_\mathcal{A} + \sigma_\mathcal{A}, \mu_\mathcal{A} + 2 \times \sigma_\mathcal{A}, \ldots, \max_{p_j \in \mathcal{D}} \mathcal{A}(p_i, p_j)\},$$

where

$$\mu_\mathcal{A} = \frac{1}{|\mathcal{D}| - 1} \sum_{\substack{p_j \in \mathcal{D} \\ p_j \neq p_i}} \mathcal{A}(p_i, p_j) \tag{13}$$

and

$$\sigma_\mathcal{A} = \sqrt{\frac{1}{|\mathcal{D}| - 1} \sum_{\substack{p_j \in \mathcal{D} \\ p_j \neq p_i}} (\mathcal{A}(p_i, p_j) - \mu_\mathcal{A})^2}. \tag{14}$$

VI. Prediction of Port Probing Rates

Monitoring the trend of the probing rates could help detect an ongoing stealth attack, or even predict an imminent threat. For instance, during the /0 *sipscan* performed by a large botnet, records of the UCSD network telescope showed a significant increase of the probing rates targeting ports 5060 and 80, during a period of 12 days, aiming to discover running SIP servers [24].

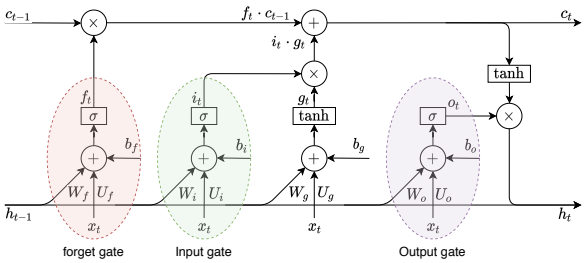


Fig. 4: LSTM Cell Architecture

Using the inferred clusters of ports sharing similar temporal and semantic probing characteristics, the probing rates of a target port can now be forecast by using the history of the probing activities of its similar ports, *i.e.* high ranked ports or ports belonging to the same cluster. This is carried by using predictive models capable of handling multivariate sequential data.

There exist an extensive collection of learning models for this purpose. For instance, in [14], the authors designed the non-stationary Vector Autoregressive (VAR) model to address the issue of non-stationarity of the probing rates time series. Even though the model showed satisfactory prediction results, its performance degrades in the case of ports having second order non-stationary probing time series such as 443 (https) and 3306 (mysql). Moreover, the autoregressive models are known for their flaw in capturing long-range dependencies and non-linear patterns embedded in the sequential data. To overcome these limitations, we used stacked Long Short-Term Memory (LSTM) neural networks.

A. LSTM Recurrent Neural Networks

Recurrent neural networks (RNNs) [25] are learning models used for processing sequential data. Their main characteristic is their "memory"; they are composed of cells allowing output values to be used as inputs while maintaining a state of what has been learned so far. The architecture of most RNN cells consist of three blocks of learnable parameters: (i) from the input to the hidden state, (ii) a self-loop hidden state which represents the memory, and (iii) from the hidden state to the output. However, the major issue with RNNs in their naive form is that, during the back-propagation of the errors in the training process, the gradients of long term signals become unstable (*i.e.* of smaller or larger magnitudes) and tend to vanish or explode [26]. This is why gated RNNs were introduced.

A widely used gated RNN cell that has proven performance is the Long Short-Term Memory (LSTM) designed by Hochreiter and Schmidhuber [27] as a capstone of their research on the *unstable gradient problem*. A LSTM cell, as illustrated in Fig. 4, is composed of three gates: the input, the forget and the output gates.

The flow of information throughout the LSTM cell is as follows. At time step t, the cell state c_{t-1} holds the memory; *i.e.* the information that has been learned so far, and the hidden state h_{t-1} carry short-term information.

The forget gate takes as input the previous hidden state h_{t-1} and the vector of features x_t that is part of the input sequence.

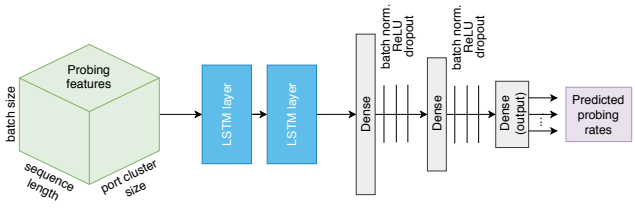


Fig. 5: LSTM neural network architecture

Then, it produces a filtering vector f_t through the sigmoid function ($\sigma(z) = \frac{1}{1+e^{-z}}$) which controls what information stored in the cell state c_{t-1} is to be forgotten:

$$f_t = \sigma(x_t \cdot U_f + h_{t-1} \cdot W_f + b_f). \tag{15}$$

Similar to the forget gate, the input gate generates a filter i_t which decides about information to be stored in the cell state:

$$i_t = \sigma(x_t \cdot U_i + h_{t-1} \cdot W_i + b_i). \tag{16}$$

The input data x_t combined with the previous hidden state h_{t-1} are squashed with the $\tanh$ activation function which acts as a regularizer and whose derivative is resistant to long range vanishing gradients.

$$g_t = \tanh(x_t \cdot U_g + h_{t-1} \cdot W_g + b_g) \tag{17}$$

The resultant *candidate state* g_t which holds the new information (aggregated from x_t and h_{t-1}) is passed through the input filter i_t in order to decide what information is to be added to the memory; the cell state c_t:

$$c_t = f_t c_{t-1} + i_t g_t. \tag{18}$$

The output gate controls which information stored in the cell state memory is to be retained to generate output values, and the hidden state is updated accordingly.

$$\begin{aligned} o_t &= \sigma(x_t \cdot U_o + h_{t-1} \cdot W_o + b_o) \\ h_t &= o_t \tanh(c_t) \end{aligned} \tag{19}$$

It is to note that $W_\bullet$, $U_\bullet$ and $b_\bullet$ are the set of learnable parameters during the training of the neural network.

B. Model Architecture

The stacked LSTM neural network used for modeling and predicting the probing rates is illustrated in Fig. 5. The model takes as input the probing rates' sequences of the ports belonging to the same cluster (or highly ranked) and outputs the probing rates' predictions of the ports of interest. The probing feature space is 3-dimensional (mini-batches of sequences of probing rates) and the feature values are normalized to zero mean and unit standard deviation.

Experimental studies strongly suggest to introduce depth by decomposing RNNs into multiple layers [28], [29]. Therefore, our architecture consists of 2 stacked LSTM layers having 256 and 128 LSTM cells, as depicted in Fig. 4. In our evaluation, two LSTM layers were enough to achieve accurate results while keeping the learning time limited. The LSTM layers are followed by two fully connected blocks of sizes 128 and 96 hidden units. Each block starts with a batch normalization layer which alleviates the problem of internal covariate shift and allows the use of higher learning rates for faster training [30]. The block also contains the ReLU activation ($ReLU(x) = \max(0, x)$) which reduces the likelihood of vanishing gradients during training. The latter is followed by dropout regularization which randomly sets a fraction of hidden units to 0 to prevents overfitting by simulating different network architectures (*i.e.* ensemble learning in a single network architecture) [31]. The output layer returns the probing rates' predictions and its size corresponds to the number of target ports. Finally, the hyperparameters related to training the model are discussed in the evaluation section.

VII. Application: Inferring Probing Traffic Anomalies

Network telescopes carry anomalous traffic related to a wide range of events like scans, worms, backscatter packets related to DoS attacks. Yet, network telescopes may also capture traffic whose intent is not detrimental, like packets related to misconfiguration and scans performed by organizations for statistical purposes. These types of background radiations are in general regular and easily discernible by rule-based or machine learning models. However, the risk of exploiting a new vulnerability, for instance, is rather unpredictable and can be considered as a traffic anomaly.

We consider a probing traffic targeting a network service as anomalous when one of the following conditions is verified:

- The prediction error is high: this occurs for instance when a probing pattern related to a new emerging threat is not recognized by the model.
- There is an abrupt increase in the probing rates: this happens when the forecast probing rate using the trained model does not follow the normal trend of the preceding probing rates.

These two conditions could be identified using the indicators introduced in the following sections.

A. Probing Anomaly Inference Indicator

To infer probing anomalies, we introduce an indicator relying on the model's prediction error. This error might be discerned when there is a significant disparity between (i) the predicted probing rate and (ii) the actual probing rate that may include irregular and malicious probes (e.g. DDoS attacks and worms traffic). The rational is, assuming that our model predicts accurately the probing rates, observing a significant prediction error could be an indicator of an abnormal traffic that has not been learned, and thus this error may represent an imminent threat.

For a given port p at a time step t, we use the following indicator:

$$E_t^{(p)} = \max\left(0, \frac{X_t^{(p)} - \widehat{X_t^{(p)}}}{\widehat{X_t^{(p)}}}\right), \tag{20}$$

where $\widehat{X_t^{(p)}}$ is the predicted probing rate using the trained model and $X_t^{(p)}$ is the probing rate recorded by the darknet.

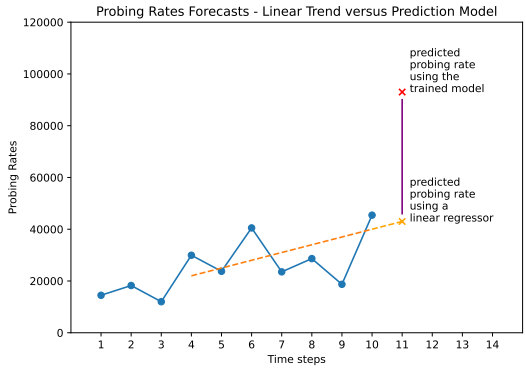


Fig. 6: The blue line represents the probing rates recorded by the darknet. The orange line represents the linear trend of the probing rates inferred on a window of 7 time steps (e.g. 7 days window). The purple line represents the difference between the forecast probing rate using the trained model and the probing rate using the linear regressor.

B. Early Warning Cyber Risk Indicator

To raise early cyber risk warnings, we use a monitoring indicator relying on the performance of the prediction model to accurately forecast future probing rates based on the dynamics incorporated in the past and current probes recorded by the darknet sensor. Thus, we consider the cyber risk high when there is a significant difference between (i) the forecast probing rate using our model and (ii) the probing rate produced using a simple linear model capturing the trend of the previous probing rates (fit in a preceding time window). Fig. 6 is an illustrative example.

Formally, we define the early warning cyber risk indicator for the network service related to a port p at time step t as follows:

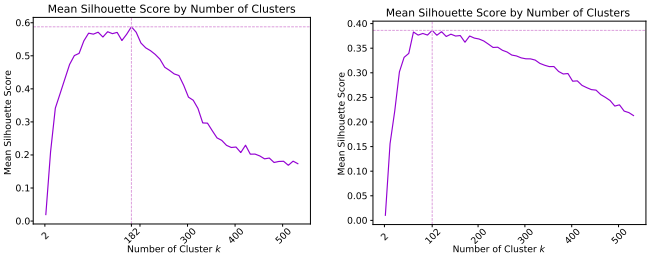
$$W_t^{(p)} = \max\left(0, \frac{\widehat{X_{t+1}^{(p)}} - L_{t+1}^{(p)}}{L_{t+1}^{(p)}}\right), \tag{21}$$

where $\widehat{X_{t+1}^{(p)}}$ the predicted probing rate using our trained model and $L_{t+1}^{(p)}$ the predicted probing rate using the linear model, both one step ahead-of-time.

A high value of $W_t^{(p)}$ means that the prediction model may have recognized a probing pattern in the port feature space and accordingly had forecast a probing rate that is significantly greater than the normal trend of probes, which could be an indicator of an imminent threat.

VIII. Results and Discussion

In this section, we report the results of port clustering, the performance of the prediction models, and we evaluate the performance of the monitoring indicators to infer anomalous traffic. We used 5 computation servers, each equipped with two Intel(R) Xeon(R) Silver 4114 CPU and 128 GB of RAM. Most of the computation tasks were parallelized across the CPU threads, and when possible, distributed across the computation servers. Also, the deep learning models for predicting the ports' probing rates were trained in a computer with the same latter configuration and equipped with 2 supplementary Nvidia GeForce GTX 1080 Ti GPUs. With this configuration, the data cleaning and preprocessing took approximately 2 months, and the training of the prediction models takes approximately between 10 and 20 days per cluster of ports, depending on the size of the cluster and the tuning of hyperparameters.



(a) 1 hour time resolution (b) 1 day time resolution

Fig. 7: Average Silhouette Score by Number of Clusters

(a) Remote Access Svc. (b) Web Servers (c) VPN Services

Fig. 8: Examples of port clusters. The text size is proportional to the number of packets that hit the given port.

A. Port Clustering

For the clustering of ports, the number of clusters k giving the best average silhouette score is selected from the range $[2, \lceil \frac{|\mathcal{D}|}{2} \rceil]$. Fig. 7 shows the silhouette scores for different time resolutions of the temporal probing similarity. The optimal number of clusters when using probing time series of 1 hour (resp. 1 day) time resolution is 182 (resp. 102).

Fig. 8 illustrates some examples of identified clusters. Note that the clusters in this figure does not include all of their ports, only representative ports are shown. Many clusters actually group ports that generally come as part of sequential scans like the cluster $\{3330, 3335, 3336, 3337, 3338, \dots\}$. Other clusters include only one port, however, these ports mostly belong to the user-defined ports' range of the IANA port categorization.

B. Prediction of Port Probing Rates

In the following, we evaluate and compare the performances of the LSTM model presented in Section VI-B and the VAR model described in [14], using different methods for defining the models' feature space that we denote: Rank. ($\mathcal{T}$), Rank. ($\mathcal{T}$&$\mathcal{S}$), Clust. ($\mathcal{T}$) and Clust. ($\mathcal{T}$&$\mathcal{S}$). Table II presents short descriptions of the evaluated models.

The experiments are conducted on a set of selected ports among the 30 most targeted ports shown in Fig. 9. The probing rates time series were split into two subsets: a training set (75% of the data) and a testing set (the remaining 25%) which

TABLE II: Description of the Predictive Models

Model	Description
LSTM Rank. ($\mathcal{T}$)	The stacked LSTM neural network described in Section VI-B. The feature space contains probing time series of the most similar ports to the target port (cf. V-D2) in terms of temporal similarity only.
LSTM Rank. ($\mathcal{T}$&$\mathcal{S}$)	The stacked LSTM neural network described in Section VI-B. The feature space contains probing time series of the most similar ports to the target port (cf. V-D2) in terms of the temporal and the semantic similarities (cf. V-C).
LSTM Clust. ($\mathcal{T}$)	The stacked LSTM neural network (cf. VI-B). The feature space contains probing times series of the ports belonging to the same cluster of the target port (cf. V-D1). The clustering is performed using as an affinity metric the temporal similarity only.
LSTM Clust. ($\mathcal{T}$&$\mathcal{S}$)	Same as above. The clustering is performed using the affinity metric combining the temporal and the semantic similarities (cf. V-C).
VAR Feature Selection	The non-stationary Vector Autoregressive model as described in [14].
VAR Rank. ($\mathcal{T}$&$\mathcal{S}$)	The Vector Autoregressive model. The feature vector contains probing rates of ports inferred using port ranking (cf. V-D2). Unlike the feature selection method, the feature vector is fixed in the beginning and not learned online.
VAR Clust. ($\mathcal{T}$&$\mathcal{S}$)	The Vector Autoregressive model. The feature vector is inferred using port clustering (cf. V-D1).

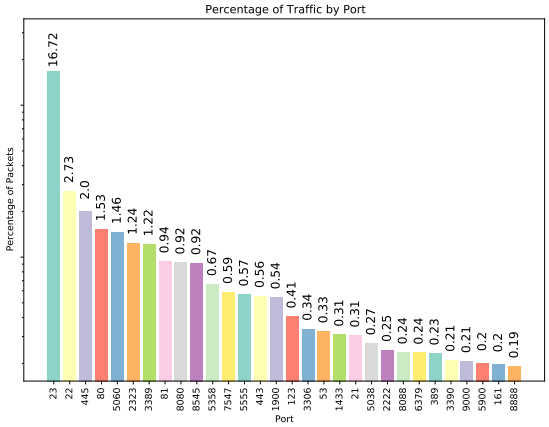


Fig. 9: Top 30 most targeted ports from January 2017 to July 2020. The percentages of packets are in log scale and the bars are annotated with the actual traffic percentages.

corresponds to the last 300 days of network telescope records. The training set has been used to optimize the models and to find the optimal hyperparameters. The R^2 scores in Tables IV and V are reported on the testing set.

1) LSTM Model Hyperparameters: To find the optimal hyperparameters reported in Table III, we perform a grid search with a 7-fold cross-validation.

To train the LSTM model, we optimize the mean absolute error MAE $= \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|$, where m is the size of the training batch, and y_i and $\hat{y}_i$ are respectively the observed and the predicted values of the i^{th} input probing sequence. Different input sequence lengths are evaluated. For the models trained using probing times series of 1 hour time resolution, sequences of 72 values constantly give the best performance.

TABLE III: List of hyperparameters of the LSTM Model

Hyperparameter	Values
Sequence length	24, 48, 72, 144 (1 hour time res.) 3, 7, 15 and 30 (1 day time res.)
Training epochs	700 with early stopping
Dropout rate	0.1, 0.3 and 0.5
Optimizer	RMSProp and Adam
Learning rate	0.001, 0.01 and 0.1
Batch size	16, 32 and 64

We also noticed that the performance of the model tends to decrease when choosing larger time steps. For 1 day forecasting models, the sequence length of 30 days constantly produces the best scores. For the number of training epochs, an early stopping strategy was implemented. We found that the early stopping callback was triggered between 300 and 500 epochs for most of the models, and that there was no need to go beyond 700 epochs, as the model starts to overfit. As for the remaining hyperparameters, namely the dropout rate, the optimizer, the learning rate and the batch size, the optimal values depend on the training data.

2) LSTM vs. VAR: As shown in Table IV (resp. Table V), the LSTM Rank. ($\mathcal{T}$ & $\mathcal{S}$) model outperforms the other LSTM and VAR models for 7 (resp. 11) ports out of the 13 ports, with a mean score R^2 of 0.79 (resp. 0.83). This is likely due to the memory property of the LSTM enabling the learning of long range dependencies and complex repeated patterns. Also, because of the non-linearities introduced in its architecture, the LSTM is able to learn hidden patterns and non-linear mappings between the probing rates history sequences and the target probing rates.

However, there are few cases when the VAR shows better results, for instance the telnet service for 1 hour resolution. This is due to the low stochasticity of the variance in the corresponding time series. Knowing that the VAR does not require a large autoregressive order [14], the trainable parameters are few, leading the VAR to produce more stable predictions, in contrast to the LSTM model which has a considerable amount of trainable parameters that introduce a noise in the predicted values.

Figures 10 and 11 show examples of the predicted probing rates using the "LSTM Rank. ($\mathcal{T}$ & $\mathcal{S}$)" model versus the observed probing rates. We observe that the probing time series are non-stationary in terms of first and second order statistics with some extreme probing rate values. It is clear that the LSTM tend to learn the trend of the time series more than the variance. This is more visible when comparing for instance the predictions for port 22 (ssh) 1 hour ahead-of-time (Fig. 10) and 1 day ahead-of-time (Fig. 11); the aggregation of the hourly probing rates into daily probing rates creates a trend in the time series and reduces the local variances, thus leading the LSTM to perform better for 1 day forecasts ($R^2 = 0.87$).

Finally, it is noteworthy that the VAR model cannot rely on a feature space defined by the port ranking and the port clustering methods. The reason is that the feature space is fixed during the training, unlike the feature selection method which allows the model to learn features while training.

3) Port Ranking vs. Port Clustering: To define the feature space of the predictive models, we used the port ranking and

TABLE IV: R^2 Scores of the Probing Rates Prediction Models (1 hour ahead-of-time forecasting). The best scores are in bold text.

	LSTM Rank. ($\mathcal{T}$ & $\mathcal{S}$)	LSTM Rank. ($\mathcal{T}$)	VAR Feature Selection	LSTM Clust. ($\mathcal{T}$)	LSTM Clust. ($\mathcal{T}$ & $\mathcal{S}$)	VAR Rank. ($\mathcal{T}$ & $\mathcal{S}$)	VAR Clust. ($\mathcal{T}$ & $\mathcal{S}$)
23 (telnet)	0.94	0.95	**0.97**	0.86	0.91	0.61	0.34
2323 (telnet alt.)	0.95	0.95	**0.96**	0.91	0.89	0.43	0.41
22 (ssh)	**0.79**	0.74	0.73	0.63	0.67	0.55	0.39
2222 (ssh alt.)	**0.75**	0.73	0.68	0.57	0.67	0.27	0.62
445 (microsoft-ds)	0.92	**0.93**	**0.93**	0.86	0.92	0.21	0.35
80 (http)	**0.73**	0.71	0.57	0.64	0.55	0.49	0.21
443 (https)	**0.67**	0.62	0.59	0.51	0.64	0.60	0.33
5060 (sip)	0.81	0.78	0.76	0.86	**0.92**	0.49	0.53
5555 (softether-vpn)	0.59	**0.62**	0.58	0.66	0.44	0.26	0.43
3306 (mysql)	**0.74**	0.73	0.65	0.67	0.65	0.55	0.23
1900 (micrsoft-ssdp)	0.78	0.68	0.69	**0.84**	0.79	0.62	0.63
1433 (mssql)	**0.80**	**0.80**	0.67	0.64	0.52	0.35	0.40
1883 (mqtt)	**0.83**	0.82	0.61	0.63	0.58	0.34	0.21
Mean R^2	0.79	0.77	0.72	0.71	0.70	0.44	0.39

TABLE V: R^2 Scores of the Probing Rates Prediction Models (1 day ahead-of-time forecasting). The best scores are in bold text.

	LSTM Rank. ($\mathcal{T}$ & $\mathcal{S}$)	VAR Feature Selection	LSTM Rank. ($\mathcal{T}$)	LSTM Clust. ($\mathcal{T}$ & $\mathcal{S}$)	LSTM Clust. ($\mathcal{T}$)	VAR Rank. ($\mathcal{T}$ & $\mathcal{S}$)	VAR Clust. ($\mathcal{T}$ & $\mathcal{S}$)
23 (telnet)	**0.93**	0.91	0.91	0.89	0.86	0.71	0.67
2323 (telnet alt.)	**0.92**	0.90	0.89	0.87	0.87	0.76	0.66
22 (ssh)	**0.87**	0.83	0.80	0.77	0.78	0.55	0.61
2222 (ssh alt.)	**0.81**	0.79	0.77	0.78	0.74	0.43	0.27
445 (microsoft-ds)	0.86	**0.87**	0.82	0.75	0.72	0.70	0.57
80 (http)	**0.71**	0.63	0.62	0.61	0.59	0.47	0.31
443 (https)	**0.72**	0.61	0.68	0.58	0.55	0.23	0.17
5060 (sip)	0.83	**0.85**	0.84	0.73	0.73	0.51	0.70
5555 (softether-vpn)	**0.77**	0.69	0.67	0.65	0.67	0.34	0.28
3306 (mysql)	**0.81**	0.73	0.73	0.71	0.69	0.39	0.41
1900 (micrsoft-ssdp)	0.79	**0.80**	0.76	0.75	0.73	0.44	0.45
1433 (mssql)	**0.87**	0.86	0.84	0.82	0.83	0.68	0.61
1883 (mqtt)	**0.84**	0.77	0.75	0.78	0.76	0.47	0.49
53 (dns)	**0.86**	0.69	0.72	0.76	0.71	0.50	0.48
Mean R^2	0.83	0.79	0.78	0.75	0.73	0.51	0.48

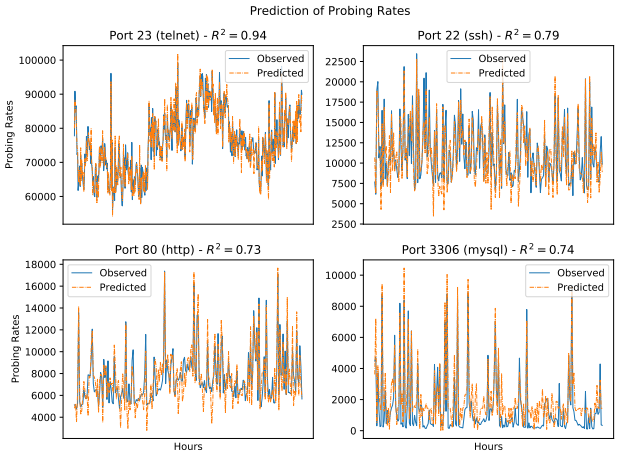


Fig. 10: Prediction of probing rates 1 hour ahead-of-time.

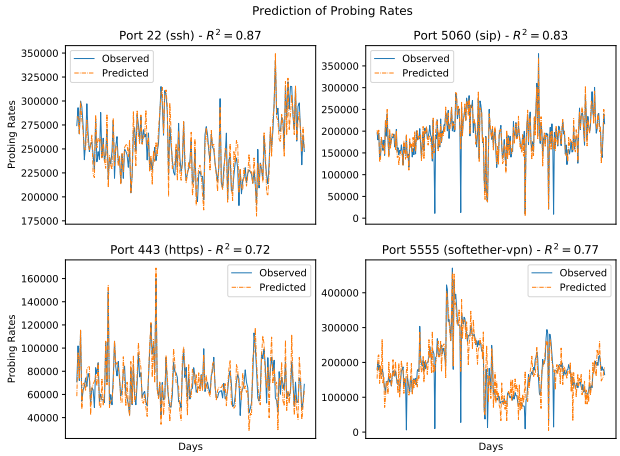


Fig. 11: Prediction of probing rates 1 day ahead-of-time.

port clustering methods described in Section V-D. The scores show that the port ranking method produces better results than port clustering because the spectral clustering assumes exclusive groups of ports. In such manner, it may exclude from the feature space a port carrying salient information that may contribute in the prediction of multiple probing rates. Contrastingly, port ranking relaxes the latter constraint, allowing the inclusion of all ports having a high probing similarity score with the target port.

4) Contribution of the Probing Semantic Similarity: We evaluated for both port ranking and port clustering methods the contribution of the semantic similarity between ports in the definition of the feature space. Tables IV and V show that the affinity metric plays an important role in reducing the noise within the feature space. Specifically, the semantic similarity combined with the temporal similarity outrun the temporal similarity when used alone. However, when performing short term predictions (on 1 hour horizon), and when using port

clustering for the inference of the feature space, the contribution of the semantic similarity is minor.

C. Anomaly Inference

To evaluate the inference of anomalies, we inspect the temporal correspondences between the indicators of anomalous probing rates introduced in VII-B and VII-A and the published vulnerabilities. It is noteworthy that the anomalies could be due to other events like DoS attacks. However, we focus the attention on the particular case of anomalies related to vulnerability disclosures because we can rely on a large public database for validation.

Thanks to the NVD CVE database [1], we extract the set of vulnerabilities bound to the network stack (*i.e.* the "attack vector" set to "network") for a given network service, using predefined regular expressions on the "description data" field. The extraction process does not take into consideration the version of the service affected by the vulnerability. This is due to the passive nature of darknets which prevents accurate service identification.

Also, our approach would omit to identify other possible correlations (between a port number and the unconventional service running on it) making our evaluation a worst case analysis. Instead, the vulnerabilities affecting a same service are aggregated. The reason is that we aim at assessing the cyber risk, and not to classify probing anomalies or to detect vulnerable services. Then, aggregated vulnerabilities related to a network service are labeled by the conventional port number (e.g. FTP and its secure variants are labeled by "port 21"). Next, we take as values of the sequence the sum of the CVSSv3 base scores of the vulnerabilities that are published in each time interval. The sequences are extracted using a time resolution of one day. Let's denote $\text{CVE}^{(p)}$ the latter sequence.

There might be a time shift between the publication of the vulnerability and the corresponding abnormal traffic given by the monitoring indicator. The reason is that a vulnerability could be exploited before or after its disclosure. To tackle this issue, we propose the use of Dynamic Time Warping (DTW) known for its ability to manage sequences varying in speed by dynamically realigning them, *i.e.* finding temporal correspondences between the data points of the two input sequences [32].

Particularly, for a given network service represented by a port p, we calculate the DTW distances between the sequence of published vulnerabilities $\text{CVE}^{(p)} = \{\text{CVE}_1^{(p)}, \text{CVE}_2^{(p)}, \dots\}$ and the sequence of the probing anomaly indicator $E^{(p)} = \{E_1^{(p)}, E_2^{(p)}, \dots\}$. Fig. 12 shows the obtained DTW distances. It is to note that the DTW similarity is not symmetric. The scores to focus on are in the diagonal, and the other scores serve as a validation basis. As the DTW distances show, there is a stronger relationship between the published vulnerabilities related to each network service and its corresponding probing anomaly indicator as proved by the smaller distances in the diagonal.

In addition, we evaluate the relationship between the probing anomaly indicator $E^{(p)}$ and the indicator $A^{(p)}$ expressed

[1] https://nvd.nist.gov/vuln/data-feeds

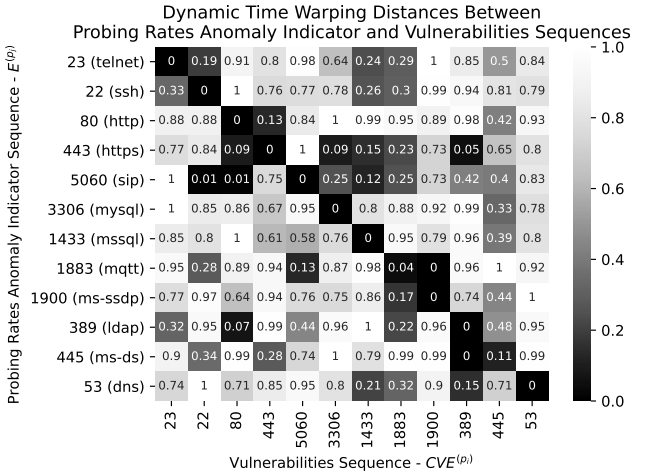


Fig. 12: The value in the i^{th} row and the j^{th} column is $\text{DTW}(E^{(p_j)}, \text{CVE}^{(p_i)})$. The DTW distances in the same row are scaled to values between 0 and 1 because the scale of the distances depends on the target port. The scores are reported for the period from August 2019 to June 2020 (i.e. testing set). The used time resolution in the input sequences is 1 day.

in (22). The latter compares the observed (the actual) probing rate with the moving average of the previous probing rates. Indeed, the deviation from the moving average is an easy and common technique to identify outliers and subsequently anomalous traffic.

$$A_t^{(p)} = \max\left(0, \frac{X_t^{(p)} - \frac{1}{m}\sum_{i=0}^{m} X_{t-i}^{(p)}}{\frac{1}{m}\sum_{i=0}^{m} X_{t-i}^{(p)}}\right) \tag{22}$$

In the above expression, m is the number of probing rates falling in the chosen moving average window. To compare the performances of the indicators $E^{(p)}$ and $A^{(p)}$ for inferring the anomalous traffic related to vulnerabilities, we use the following score:

$$S_E^{(p)} = 1 - \frac{\text{DTW}(E^{(p)}, \text{CVE}^{(p)})}{\text{DTW}(A^{(p)}, \text{CVE}^{(p)})}, \tag{23}$$

We note that $S^{(p)} \leq 1$. A value of $S^{(p)}$ close to 1 means that, by using the monitoring indicator $E^{(p)}$, we can infer anomalous traffic that is related to vulnerability disclosures more precisely than using the simpler monitoring indicator $A^{(p)}$, which considers the darknet traffic as being anomalous when it diverges from the moving average probing rate. Thus, monitoring the probing activities using the $E^{(p)}$ indicator is a more accurate approach to infer anomalies related to vulnerabilities. In other terms, when the indicator $E^{(p)}$ takes large values, the cyber-risk related to vulnerability disclosure is elevated.

Fig. 13 depicts the obtained scores for different network services. The $E^{(p)}$ indicator outperforms the standard indicator $A^{(p)}$ in inferring anomalous probing activities related to vulnerabilities with an increase of accuracy ranging between 11% and 71%, depending on the target port.

As for the early warning cyber risk indicator $W^{(p)}$ introduced in Section VII-B, we compare the indicator $W^{(p)}$

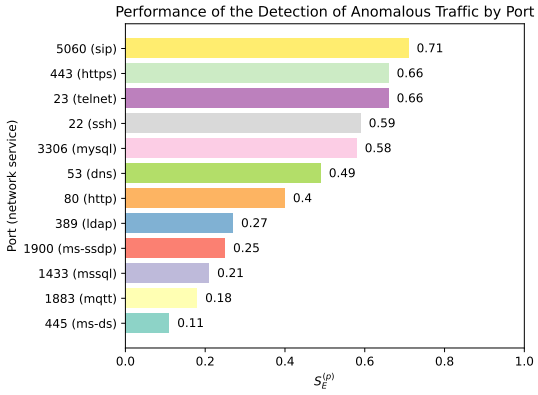


Fig. 13: Performance of inferring anomalous traffic related vulnerabilities by network service. The scores are reported for the period from August 2019 to June 2020. The moving average window used for $A^{(p)}$ is 7 days.

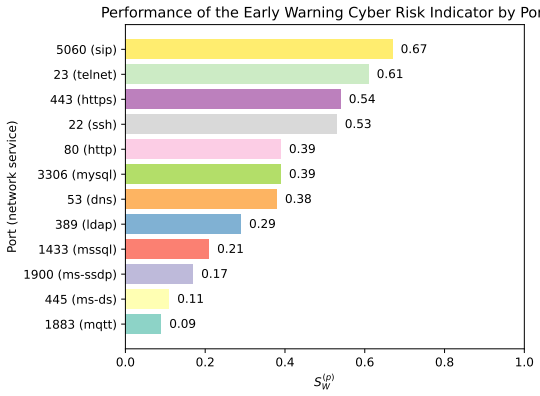


Fig. 14: Performance of forecasting anomalous traffic related vulnerabilities by network service. The scores are reported for the period from August 2019 to June 2020. The moving average window used for $A^{(p)}$ is 7 days.

with the indicator $A^{(p)}$ expressed in Equation 22, using the following score:

$$S_W^{(p)} = 1 - \frac{\text{DTW}(W^{(p)}, \text{CVE}^{(p)})}{\text{DTW}(A^{(p)}, \text{CVE}^{(p)})}, \tag{24}$$

Fig. 14 shows the obtained scores for different network services. Similarly, the $W^{(p)}$ indicator outperforms the standard indicator $A^{(p)}$ in forecasting anomalous probing activities related to vulnerabilities with an increase of accuracy ranging between 9% and 67%. Also, we observe that the early warning indicator $W^{(p)}$ is able to accurately forecast anomalies 1 hour and 1 day ahead-of-time without a significant loss in performance when compared to the traffic anomaly indicator $E^{(p)}$. This may have a practical implication for network security operators as it allows to assess the cyber-security risk early and thus proactively implement defense means and strategies.

IX. Conclusion

In this research work, we exploited large scale Internet traffic captured by two network telescopes to model and monitor the probing rates at the service level. We designed a framework inferring the most correlated network services through clustering and ranking of ports to model and forecast the probing rates using LSTM neural networks. The designed affinity metric measuring the temporal and the semantic similarities between ports demonstrated its efficacy in reducing the dimensionality and the noise within the feature space of the predictive models. Even though the stochasticity and the non-stationarity of the probing time series, our approach have proven to produce better predictions than the non-stationary vector autoregressive model. Also, we described how the probing rates model could be leveraged to improve the monitoring of the probing activities recorded by a network telescope. We proposed new indicators relying on the prediction models, that are efficient in inferring anomalous traffic related to the exploit of vulnerabilities and in raising an early warning when the cyber risk is high.

As a limitation of this work, darknet traffic includes misconfiguration packets and periodic regular scans performed by many organizations. It would be beneficial to isolate such non-malicious traffic to discern malicious probing activities and assess to which extent these probing activities are predictable.

As for future work, other than addressing the aforementioned limitation, it is worth investigating to which extent the attention-based models [33] could improve the predictions of probing rates, while allowing interpretability of the model.

Acknowledgment

This research work is supported by Lorraine Université d'Excellence program of Université de Lorraine. The authors would like to thank Frédéric Beck from High Security Laboratory of Inria Nancy-Grand Est for providing the network telescope data and computation servers.

References

[1] K. Bissell, R. M. LaSalle, and P. Dal Cin, "Ninth annual cost of cybercrime study," 2019, (Last accessed on May 27, 2022). [Online]. Available: https://www.accenture.com/us-en/insights/security/cost-cybercrime-study

[2] I. Security, "Cost of a data breach report," 2020, (Last accessed on May 27, 2022). [Online]. Available: https://www.ibm.com/security/digital-assets/cost-data-breach-report/

[3] Cisco, "Cisco annual internet report, 2018–2023," 2020, (Last accessed on May 27, 2022). [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[4] D. Moore, C. Shannon, G. Voelker, and S. Savage, "Network Telescopes: Technical Report," Cooperative Association for Internet Data Analysis (CAIDA), Tech. Rep., 2004.

[5] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Towards a forecasting model for distributed denial of service activities," in *IEEE 12th International Symposium on Network Computing and Applications*, 2013, pp. 110–117.

[6] E. Bou-Harb, M. Debbabi, and C. Assi, "A time series approach for inferring orchestrated probing campaigns by analyzing darknet traffic," in *10th International Conference on Availability, Reliability and Security*, 2015, pp. 180–185.

[7] L. Müller, M. Luckie, B. Huffaker, kc claffy, and M. Barcellos, "Spoofed traffic inference at ixps: Challenges, methods and analysis," *Computer Networks*, vol. 182, p. 107452, 2020.

[8] P. Richter and A. Berger, "Scanning the scanners: Sensing the internet from a massively distributed network telescope," in *Proceedings of the Internet Measurement Conference*. Association for Computing Machinery, 2019, p. 144–157.

[9] C. Shannon and D. Moore, "The spread of the witty worm," *IEEE Security Privacy*, vol. 2, no. 4, pp. 46–50, 2004.

[10] B. Irwin, "A network telescope perspective of the conficker outbreak," in *2012 Information Security for South Africa*, 2012, pp. 1–8.
[11] Z. Durumeric, M. Bailey, and J. A. Halderman, "An internet-wide view of internet-wide scanning," in *23rd USENIX Security Symposium (USENIX Security 14)*. USENIX Association, Aug. 2014, pp. 65–78.
[12] S. Torabi, E. Bou-Harb, C. Assi, M. Galluscio, A. Boukhtouta, and M. Debbabi, "Inferring, characterizing, and investigating internet-scale malicious iot device activities: A network telescope perspective," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018, pp. 562–573.
[13] S. Torabi, E. Bou-Harb, C. Assi, E. B. Karbab, A. Boukhtouta, and M. Debbabi, "Inferring and investigating iot-generated scanning campaigns targeting a large network telescope," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2020.
[14] M. Zakroum, A. Houmz, M. Ghogho, G. Mezzour, A. Lahmadi, J. François, and M. E. Koutbi, "Exploratory data analysis of a network telescope traffic and prediction of port probing rates," in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2018, pp. 175–180.
[15] F. Soro, I. Drago, M. Trevisan, M. Mellia, J. Ceron, and J. J. Santanna, "Are darknets all the same? on darknet visibility for security monitoring," in *2019 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2019, pp. 1–6.
[16] A. Houmz, G. Mezzour, K. Zkik, M. Ghogho, and H. Benbrahim, "Detecting the impact of software vulnerability on attacks: A case study of network telescope scans," *Journal of Network and Computer Applications*, p. 103230, 2021.
[17] E. Bou-Harb, M. Husák, M. Debbabi, and C. Assi, "Big data sanitization and cyber situational awareness: A network telescope perspective," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 439–453, 2019.
[18] F. Soro, M. Allegretta, M. Mellia, I. Drago, and L. M. Bertholdo, "Sensing the noise: Uncovering communities in darknet traffic," in *2020 Mediterranean Communication and Computer Networking Conference (MedComNet)*, 2020, pp. 1–8.
[19] D. Cohen, Y. Mirsky, M. Kamp, T. Martin, Y. Elovici, R. Puzis, and A. Shabtai, "Dante: A framework for mining and monitoring darknet traffic," in *Computer Security – ESORICS 2020*. Springer International Publishing, 2020, pp. 88–109.
[20] T. Ban, L. Zhu, J. Shimamura, S. Pang, D. Inoue, and K. Nakao, "Detection of botnet activities through the lens of a large-scale darknet," in *Neural Information Processing*. Springer International Publishing, 2017, pp. 442–451.
[21] L. Evrard, J. François, and J. Colin, "Attacker behavior-based metric for security monitoring applied to darknet analysis," in *IFIP/IEEE Symposium on Integrated Network and Service Management*, 2019, pp. 89–97.
[22] M. Khadilkar, N. Feamster, M. Sanders, and R. Clark, "Usage-based dhcp lease time optimization," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*. Association for Computing Machinery, 2007, p. 71–76.
[23] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. MIT Press, 2001, p. 849–856.
[24] A. Dainotti, A. King, K. Claffy, F. Papale, and A. Pescapé, "Analysis of a "/0" stealth scan from a botnet," *IEEE/ACM Transactions on Networking*, vol. 23, no. 2, pp. 341–354, 2015.
[25] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, p. 533–536, 1986.
[26] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen netzen," 04 1991.
[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.
[28] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.
[29] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *Proceedings of the Second International Conference on Learning Representations*, 2014.
[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, 2015, p. 448–456.
[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
[32] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, p. 561–580, 2007.
[33] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," 2020.

BIOGRAPHY

Mehdi Zakroum *(Member, IEEE)* received the B.S. degree in Mathematics from University of Montpellier, France, and the M.S/Engineering degree in Computer Science and Data Science from Polytechnic School of the University of Lille, France. Currently, he is a Ph.D. candidate affiliated with TICLab at the International University of Rabat (Morocco), LORIA (France), and the RESIST team, which is a joint team between INRIA and University of Lorraine (France). His current research focuses on network monitoring and cyber-threat inference and predictability using machine learning techniques.

Dr. François Jérôme *(Member, IEEE)* received the Ph.D. degree in computer science from the University of Lorraine, France, in December 2009. He was then appointed as a Research Associate with the University of Luxembourg. He is currently a Research Scientist within RESIST Team, Inria. His main research areas are focused on the use of data analytics techniques for security and also its coupling with network softwarization. In 2019, he received the IEEE Young Professional Award in Network and Service Management.

Isabelle Chrisment received the Ph.D. degree in Computer Science from the University of Nice-Sophia Antipolis, France, in 1996, and the Habilitation degree from Henri Poincare University, Nancy, in 2005. She is a Professor of Computer Science at TELECOM Nancy Engineering School, University of Lorraine, France. Since 2014, she has been the Scientific Team Leader of the RESIST Team (formerly, MADYNES Team), a joint team between Inria and the University of Lorraine. Her main research area is related to network monitoring and security, and especially, within dynamic and large-scale networks.

Mounir Ghogho *(Fellow, IEEE)* received the M.S. degree in 1993 and the Ph.D. degree in 1997 from the National Polytechnic Institute of Toulouse, France. He was an EPSRC Research Fellow with the University of Strathclyde (Scotland), from Sept 1997 to Nov 2001. In Dec 2001, he joined the school of Electronic and Electrical Engineering at the University of Leeds (England), where he was promoted to full Professor in 2008. While still affiliated with Leeds University, in 2010 he joined the International University of Rabat (UIR) where he is currently Dean of the College of Doctoral Studies and Director of the ICT Research Laboratory (TICLab). He was awarded the UK Royal Academy of Engineering Research Fellowship in 2000 and the IBM Faculty Award in 2013. He was elevated to the grade of IEEE Fellow in 2018.