Journal of Network and Computer Applications 136 (2019) 114–131

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Traffic-flow analysis for source-side DDoS recognition on 5G environments

Marco Antonio Sotelo Monge [a], Andrés Herranz González [a], Borja Lorenzo Fernández [a], Diego Maestre Vidal [a], Guillermo Rius García [a,b], Jorge Maestre Vidal [a,b,*]

[a] Complutense University of Madrid, Calle Profesor José García Santesmases s/n, Ciudad Universitaria, 28040, Madrid, Spain
[b] Indra Sistemas S.A., Avenida de Bruselas, 35, 28108, Alcobendas, Madrid, Spain

ARTICLE INFO

Keywords:
5G
Denial of service
Intrusion detection systems
Source-side detection
Knowledge acquisition

ABSTRACT

This paper introduces a novel approach for detecting the participation of a protected network device in flooding-based Distributed Denial of Service attacks. With this purpose, the traffic flows are inspected at source-side looking for discordant behaviors. In contrast to most previous solutions, the proposal assumes the non-stationarity and heterogeneity inherent in the emergent communication environment. In particular, the approach takes advantage of the monitorization and knowledge acquisition capabilities implemented in the SELFNET (H2020-ICT-2014-2/671672) project, which facilitates its implementation as a self-organizing solution on 5G mobile networks. Monitorization, feature extraction and knowledge acquisition tasks are carried out on centralized control plane, hence the proposed architecture minimizes the impact on operational performance and prompts the end-points mobility. The preliminary results observed when considering different metrics, adjustment parameters, and a dataset with traffic observed in 61 real devices proven efficiency when distinguishing normal activities from DDoS behaviors of different intensity. With an optimal granularity selection, the highest AUC reached values close to 1.0 when measured under the most intense attacks, hence demonstrating optimal TPR and FPR relationships by adapting to the instantiated use cases.

1. Introduction

The significant increase in Distributed Denial of Service (DDoS) attacks registered in the last years has warned the main organizations for cybersecurity (CCN-CERT, 2017), hence posing well-known threats to the information society. A significant example of this problem was observed in October 2016, when the DNS servers of the Dyn provider registered one of the most complex and mediatic DDoS campaigns (Almeida et al., 2017). This resulted in disabling dozens of services, web pages and social networks, some of them related with wide-spreading solutions, for example Twitter, Reddit, Github, Amazon or Spotify. This was achieved by exploiting a vulnerability present in millions of devices of different nature connected to the Internet of Things (IoT) (Bertino and Islam, 2017; Sicari et al., 2018). The threat was orchestrated from a botnet managed by the malware specimen Mirai (Kolias et al., 2017; Antonakakis et al., 2017), and the attack served to aggravate the uncertainty of many users about the safety of their network devices, which as a result of this incident or similar attacks, would ask themselves: are also my end-user or IoT devices taking part of remotely coordinated malicious campaigns? in this case, what are their purposes? to what extent are they contributing? or, how can I prevent such situations?.

Despite the importance of combating these threats by monitoring single source-side devices, this approach has barely been studied by the research community from the point of view of communication networks (Zargar et al., 2013; Yan et al., 2016), whose efforts usually aimed on analyzing network traffic at the intermediate/victim edges of the intrusion, or at deepening into identifying infections by remote control malware, typically related with botnets (Acarali et al., 2016). Fortunately, as a result of the emergence of novel communication network technologies (Software-Defined Networking (SDN), Network Function Virtualization (NFV), etc.) and the recent advances towards consolidating the fifth generation networks (5G) (Gavrilovska et al., 2016), the detection of DDoS attacks by analyzing traffic monitored at source-side acquires a new meaning, now playing an essential role in defining defensive strategies based on Self-Organizing Network (SON) deployments (Maestre Vidal et al., 2018). This requirement contrasts with a bibliography with predominance of obsolete approaches (Yan et al., 2016), that usually

* Corresponding author. Indra Sistemas S.A., Avenida de Bruselas, 35, 28108, Alcobendas, Madrid, Spain.
E-mail addresses: masotelo@ucm.es (M.A. Sotelo Monge), andrhe01@ucm.es (A. Herranz González), borjalor@ucm.es (B. Lorenzo Fernández), diemae01@ucm.es (D. Maestre Vidal), grius@ucm.es (G. Rius García), jmaestre@ucm.es (J. Maestre Vidal).

https://doi.org/10.1016/j.jnca.2019.02.030
Received 23 July 2018; Received in revised form 24 February 2019; Accepted 26 February 2019
Available online 4 March 2019

do not assume the characteristics expected in future networks, among them high heterogeneity, non-stationarity or reduction of operational expenditures.

In order to contribute to the development of solutions capable of dealing with the aforementioned problems, this paper introduces the FlowSentinel intrusion detection approach. FlowSentinel addresses the challenge of analyzing outbound traffic flows looking for traits of malicious activities, in particular those related with the involvement of a device as source-side of DDoS attempts. The discovery of suspicious activities is driven by estimating the monitored traffic behavior based on the study of aggregated metrics and the elaboration of prediction intervals. When they are surpassed, the discordance is tagged as anomalous, thus being reported as suspicious situations related with malicious resource depletion. With experimental purposes, FlowSentinel was originally built for Android systems. However, when adopting the proper implementation the method is scalable to alternative IoT technologies.

The first FlowSentinel implementation posed a portable solution, where the entire analytics were performed on the device itself, hence allowing users to install and run a defensive application that executed each data processing stage (Dro, 2018a). But due to the heterogeneity and non-stationarity inherent to the traffic output of a single mobile device, which usually relies on the user behavior, the analytics were adapted to changes in the distribution of monitored data, in this way gaining sophistication. Despite their effectiveness, these modifications implied important penalties in terms of quality of user experience, among them significantly CPU, memory and battery consumption. Another requirement of the new generation networks to be borne in mind is the compatibility of the traffic analysis with self-organization schemes, in this way allowing to diagnose the state of the network, making decisions and applying countermeasures without human operator supervision (closed-loop). Because of this, the current version of FlowSentinel has been designed and developed as an anomaly detection framework under the SELFNET project (H2020-ICT-2014-2/671672) (Dro, 2018d), from which inherits monitoring, correlation, analysis and decision-making capabilities. The solution facilitates its adaptation to more sophisticated self-protection approaches. As result, the main contributions of the performed research are:

- The in-depth review of the flooding-based DDoS landscape and the latest proposals for its mitigation in the bibliography.
- A novel method for flooding-based DDoS identification by analyzing source-side activities. Aiming on facilitating its interoperability with the emergent communication scenarios, the solution has been adapted to the non-stationarity inherent in large and heterogeneous environments.
- A detailed description about how the proposal was integrated in an advanced 5G multi-layered architecture for self-protective purposes, that describes a framework for further similar deployments.
- A labeled dataset for training/evaluation purposes with real traffic captures from 61 devices of different nature. It gathers 72,400 normal traffic samples, and 78,300 samples of flooding-based attacks.
- An evaluation methodology able to assess the effectiveness of similar proposals.
- An exhaustive discussion of the achieved results based on different dimensions (attack characterization, family of devices, monitorization granularity, etc.), that aims on facilitating the comparison of our findings with those of future publications

In order to facilitate the understanding of the proposal, the paper has been organized in the following eight sections: Section 1 introduces the DDoS problem and the main motivations of the performed research; Section 2 reviews de main traits of DDoS attacks and the different approaches for their mitigation; Section 3 details the assumed design principles and the FlowSentinel architecture; Section 4 describes the metrics considered at the different data processing stages; Section 5 proposes a novel DDoS detection strategy based on studying traffic flows; Section 6 defines its evaluation methodology; Section 7 discusses the obtained results; and finally, Section 8 presents the conclusions and future work.

2. Background

The principal traits of the flooding-based Denial of Service attacks and the most relevant countermeasures proposed by the research community are described throughout this section.

2.1. Flooding-based denial of service

Nowadays there are different procedures that intentionally may lead to deplete the resources of a network element, thereby denying its service. The performed research focuses on those based on flooding the victim with malicious requests (Zargar et al., 2013), which typically have been categorized in high-rate and low-rate DDoS attacks (Wei et al., 2013). This taxonomy considered as classification, criterion their request frequency. According to this classification, the first family of threats gathers techniques grounded in timely injecting a large volume of traffic/requests. On the other hand, low-rate DDoS intends to go unnoticed over the security measures by adopting incremental or activation/deactivation request injection patterns, that usually are less visible than conventional flooding-based intrusion attempts (Bhuyan et al., 2015; Li et al., 2015). Furthermore, the attacker may take advantage of reflection (Xiao et al., 2017) and/or amplification (MacFarland et al., 2017) to magnify the impact of the intrusion. A clear example of this is observed in the Link Flooding Attacks (LFA) (Wei et al., 2016; Wang et al., 2018b), where low rate request flows from regions with high traffic density are reused aiming on overflowing the computing capacity of intermediate network elements, thus resembling legitimate traffic and making the threat difficult to be discovered. Flooding-based denial of service was originally achieved from a single point of the network, which was commonly referred as conventional Denial of Service (DoS) attacks. But the trend of the current devices towards gaining computing capacity, the advances in cloud computing, and the tendency to implement self-scaling and load balancing mechanisms, entail that nowadays, the attacker requires a large number of end-point devices with traffic injection capabilities to reach their malicious purpose, this situation being typified as DDoS. Because of this, intruders usually resort to botnets (Matta et al., 2017) for acquiring offensive power. Botnets are increasingly extensive and adapt to the emergent network scenarios (Antonakakis et al., 2017). In addition, they have evolved towards robustness and evasion of mitigation techniques (Vormayr et al., 2017), which make them a dangerous shuttle of denial of service attempts. However, given the great bibliography related to the botnet problem, this paper does not deepen in them, thus suggesting publications like (Hoque et al., 2015) for understanding their most relevant features.

2.2. Countermeasures

Most of the efforts of the research community dedicated to the defense against DDoS assumed the aforementioned circumstances (Chadd, 2018). But given the complexity of the problem to be solved, the proposals are often divided into four different challenges (Vormayr et al., 2017): prevention, detection, mitigation and identification of sources. DDoS prevention focuses on avoiding the attack from reaching the victim, hence covering measures that range from applying filtering policies to traffic redistribution. Note that unlike mitigation, the prior identification of the intrusion with this purpose is not required. According to the bibliography, prevention approaches typically consider univocal features of the legitimate traffic (Luo et al., 2013), Turing tests (Wang et al., 2018a), security protocols (Khanna et al., 2011) or reputation-based systems (Wang et al., 2017b). But although they may

entail robust first lines of protection, they tend to significantly impact the hardened systems in terms of QoS/QoE, since these proactive actuations are not usually dependent of the risk level. A previous identification of potential DDoS threats may improve the balance between security and the cost of deploying countermeasures.

The research focused on DDoS detection requires studying features of the intrusion itself, either through the analysis of the traffic involved and/or the study of network-level behaviors (Zargar et al., 2013; Semerci et al., 2018). To this end, different analytic techniques were adopted, among them hidden Markov model (Holgado et al., 2017), artificial neural networks (Saied et al., 2016), change-point analysis (Behal et al., 2018), entropy-based analysis (Bhuyan et al., 2015), game theory (Wu and Wang, 2018), support vector machines (Al-Yaseen et al., 2017) or decision trees, the latter discussing the efficiency of different machine learning methods. But they rarely have been adapted to the characteristics of the emergent monitoring environments, lacking among others of adaptation to non-stationarity, robustness against adversarial attacks or effectiveness when dealing with heterogeneous data sources.

As highlighted in Yan et al. (2016), within this group, the source-side DDoS detection played a minority role in the bibliography. It was classically addressed by validating the destination hosts of the outgoing traffic (Ferguson and Senie, 2000), or by recognizing discordant traffic patterns at flow-level (Zargar et al., 2013). For example, D-WARD (Mirkovic and Reiher, 2005) proposed the construction of models that represented the normal usage of the traffic flowing through the protected system. It was based on classifying metrics extracted from traffic flows, from which it was possible to distinguish discordant behaviors. Another interesting contribution is illustrated in Gil and Poletto (2001), where the proportionality between outgoing and incoming traffic is studied. Both of them present the aforementioned limitations. With the advent of the SDN-based technologies (Sahoo et al., 2018; Imran et al., 2019), this detection paradigm (Yan et al., 2016) has been revised, leading to analyze the flow-tables inherent to the OpenFlow protocol. In this way, DDoS attacks originated in groups of compromised endpoints can be detected (Mehdi et al., 2011), which usually are IoT or end-user devices (Jin and Wang, 2013). But this requires conducting monitorization and feature extraction processes on data gathered in at least small/medium network regions, that is not possible in some context and is heavily constrained by privacy and data protection policies. In addition, they were not designed with the purpose of dealing with non-stationary/non-lineal monitoring environments.

Once the intrusion is recognized, the mitigation measures act. They involve the reactive and proactive deployment of some of the aforementioned prevention techniques, as well as the reinforcement of the network perimeters that displayed a higher risk level, which would lead to define quarantine regions (Maestre Vidal et al., 2018). They can also assume the instantiation of alternative countermeasures usually related to the active security model, among them deployment of honeypots and decoys (Wang et al., 2017a), or the redirection of malicious traffic (Thilak and Amuthan, 2018), typically towards sinkhole servers (Jhaveri et al., 2017). Mitigation actions typically entail collateral damage, that makes their actuation heavily dependent of the situational awareness considered at decision-making. Within this security paradigm are framed the main techniques for identifying the source of the threats. Their principal objective is to discover the intruder, for which the adoption of traceback measures, highlighting among them packet marking techniques, is frequent. In Alenezi and Reed (2014) some of the most popular source identification approaches are reviewed in detail. Note that because they often directly depend on the network topology (Kiremire et al., 2014), and the fact that in real scenarios they suffer restrictions imposed by the different data protection policies (Denning, 2014), their realistic goal frequently tends to be simplified as to get as close as possible to the intruder, in this way allowing to extend the range of action of the instantiated security actuators.

2.3. DDoS detection in 5G networks

Conventional DDoS detection methods have been carried out by placing security nodes on strategic points of the network infrastructure, typically categorized as: victim-end, core and source-end (Mirkovic and Reiher, 2005). Conducting this approach proven effectiveness in the past but entailed a raising number of restrictions as network architectures have gained both heterogeneity and complexity. When no separation of data and control planes was achieved, the analytical tasks were fully delegated to security servers (firewalls, IDS, IPS). Such approach entails rigidity since all the detection logic depends on the limitations of the network equipment, usually in terms of hardware/software features. It also entails a penalization in network performance, especially in situations where DDoS traffic analysis should be capable to handle higher data rates (Maimo et al., 2018b).

Novel defensive strategies in 5G networks strive to overcome the limitations of conventional detection/mitigation methods. This fact has been explicitly raised by the 5G PPP working group, considering three fundamental aspects: multi-tenancy support, self-adaptation to the network context and performance awareness (Dro, 2017a). Multi-tenancy security is intrinsically related to cloud computing environments, understood as the capability to allow different network operators to share the same physical infrastructure by isolating traffic between them, mainly implemented with encapsulation protocols such as VxLAN and GRE (Mamolar et al., 2018). In addition, self-adaptation to the network is the capability to adapt the defensive strategy to the changes observed in the topology or in the network traffic behavior (Maimo et al., 2018b). The latter characterized by non-stationary in 5G (Ditzler et al., 2015). Likewise, the impact on network performance should be maintained as minimum as possible.

Network security in 5G goes beyond traditional solutions since it has shifted towards the integration of the smartest management paradigms. Flooding-based DDoS countermeasures are no longer exposed to the aforementioned limitations but take advantage of the complexity of 5G architectures inherently tied to self-protective approaches, as it was approached by the SELFNET framework (Neves et al., 2017) when detecting and mitigating DDoS attacks conducted by botnets. Advanced monitoring schemes can coexist to fetch metrics for analytical purposes, embracing those related with physical or virtualized network elements. Victim-end, core and source-end detection can be easily accommodated as well, motivated by the operator's defensive strategy. Being SDN a key element of the 5G architecture, the separation of control and data planes accomplishes a triple purpose. First, the detection logic is delegated to the control plane which can conduct complex analytical tasks entirely implemented by software, such as flow-metrics discrimination or classification (Bhuyan et al., 2015). Second, it significantly reduces network overhead because the network nodes in the data plane are merely dedicated to pass traffic according with the forwarding rules enforced from the controller (Braga et al., 2010). In addition, SDN offers a centralized abstraction of the network elements, hence making awareness of topological/behavioral changes easily achievable. On the other hand, the responsive/preventive actions deployed in the network provide the same flexibility in 5G. Its operational behavior is closely related with the on-demand provisioning model inherited from cloud computing. Defensive actions can be put in place in diverse locations of the network infrastructure in the form of customized VNFs. Such is the case of NFV orchestration model to mitigate denial of service attacks based on DNS amplification (Lal et al., 2017) by scaling-out DNS services to handle malicious queries originated by attackers (Wang, 2019). Typical DDoS mitigation actions enforced as traffic diversion nodes can also be deployed by placing SDN/VNFs applications in crucial locations of the network, commonly closer to the source/destinations of the attack (Zhou and Guo, 2017).

The operational landscape pushed by the enabling technologies of 5G networks extends flooding-based DDoS detection capabilities as they can be transversally deployed (Mamolar et al., 2018). Heterogeneous

metrics gathered from multiple domains of the 5G architecture, ranging from those acquired at infrastructure/physical level (such as end to end latency) to the ones intrinsically related with higher-level operation (such as jitter) can be jointly analyzed under a self-adaptive model based on machine-learning, hence laying the architectural and functional distinctions from conventional networks.

3. Design principles

This section delves into the FlowSentinel design principles reviewing its objectives, assumptions and limitations. It also describes the current FlowSentinel architecture and the strategy for acquiring initial factual knowledge, in this way detailing the procedures for monitoring metric generation and knowledge inference.

3.1. Objectives

As highlighted in Section 2, the defense against flooding-based DDoS attacks may be approached from different perspectives, ranging from prevention to identification of sources (Vormayr et al., 2017). In addition, and given the complexity of the emerging network scenarios, they pose a large number of challenges, as is the case of deciding the most effective countermeasures and range of action (Zargar et al., 2013), to adapt to the nature of the data to be modeled (Bhuyan et al., 2015) or how implement previously agreed security management policies (Denning, 2014). In order to facilitate the understanding of the performed research, it should be clear that the main objective of this contribution has been the development of a novel flooding-based DDoS attacks detection strategy at source-side, that must to adapt to non-stationary processes in the data to be analyzed. Unlike similar proposals, only a single source of information is monitored, which is the protected device (Mehdi et al., 2011). The most relevant secondary goal is its integration as sensing/analysis model in a self-organizing solution for next generation networks, thus allowing both taking advantage of its dedicated analytical capabilities and fostering the definition of sophisticated uses cases able to reactively/proactively manage the protected network security. To this end, the SELFNET project (H2020-ICT-2014-2/671672) (Dro, 2018d) has been selected, which aligns with the European 5G PPP Security Work Group.

3.2. Assumptions

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

- The detection of the participation of an end-user or IoT device as source-side of DDoS attacks based on the study of metrics aggregated from its incoming/outgoing traffic is possible. In fact, this is the alternative hypothesis of the performed research, being the opposite its null hypothesis.
- Flooding-based DoS differ from normal activities in traits related with number of requests observed and traffic volume generated by the suspicious end-points. In the case of DDoS attacks, the number of involved clients also varies (Sotelo Monge et al., 2017a).
- The analysis of discordant behaviors in aggregated metrics at flow-level enables recognizing DDoS situations on conventional monitoring scenarios (Ozcelik and Brooks, 2015).
- By extracting and analyzing advanced metrics in a remote dedicated server it is possible to provide a passive-monitoring detection approach.
- As the information provided by incoming/outgoing traffic flows monitored from network devices largely depends on the traffic profile, its non-stationarity is assumed (since not for all traffic profiles stationarity can be guaranteed). The non-stationarity is also inherent to the emergent communication networks landscape (Gavrilovska et al., 2016).

3.3. Limitations

For diverse reasons, the performed research has not taken into account the following circumstances, most of them being postponed for future work:

- The protection of communication channels between monitoring agents and SON analytical components has not been addressed (Lateef et al., 2015). Consequently, at the performed experimentation it was assumed they were not compromised.
- Although the SELFNET framework provides a plethora of advanced incident correlation capabilities, the current implementation of FlowSentinel does not explore their full potential. Therefore this becomes an interesting line of future work.
- Nowadays there are different adversarial threats able to evade detection methods similar to those studied during the course of the performed research (Ozcelik and Brooks, 2015). But given the complexity that their development often entails, and aiming on facilitating the understanding of the main contribution of our research, their adoption is out of the scope of this publication. This includes hindering approaches like network address spoofing, being assumed that solutions similar to those described in Yaar et al. (2006) are deployed.
- The issues related with data protection inherent in the audition of user behaviors at communication networks have not been considered. Neither the implementation of the recent European General Data Protection Regulation (GDPR). Consequently, it is assumed that FlowSentinel have permission to monitor the incoming/outgoing traffic of their network devices for purely analytic purposes.
- The knowledge representation and data models implemented for management and storage of the factual knowledge acquired by FlowSentinel are not detailly specified throughout this paper.

3.4. Architecture

The FlowSentinel architecture is a SON solution (Fig. 1) grounded in the functional layers defined for the SELFNET framework (Dro, 2018d). At a glance, this 5G oriented architecture takes advantage of the decoupling of the control and data plane layers, promoted by SDN to allow a software-driven management model, being this a remarkable characteristic of the next-generation networks (Agiwal et al., 2016). The major benefit of this model is the inclusion of complex data processing tasks in the SON Autonomic Layer, which encompasses advanced data extraction features, machine learning algorithms, anomaly detection methods, among others, towards the accomplishment of self-protection capabilities for detecting and mitigating network threats in complex network contexts (Maimo et al., 2018a). In the lowest level of this 5G-oriented architecture, the physical infrastructure holds heterogeneous network nodes embracing end-user devices (D2D), such as mobile phones and personal computers, or IoT devices intended for machine to machine (M2M) communications (Demestichas et al., 2013; Palattella et al., 2016). All of them act as traffic generators and so the complexity of the monitored environment increases as the network grows.

On the other hand, the inclusion of virtualization capabilities driven by Network Functions Virtualization (NFV) leads to overcome scalability issues of the physical infrastructures (Hossain and Hasan, 2015) by provisioning resources on-demand. Such dynamic allocation is orchestrated by the Virtual Infrastructure Manager (implemented in Openstack (Dro, 2015a), which allows the instantiation of network elements easily configurable by software. This elastic provisioning model is a distinguishable aspect of 5G architectures, and is carried out, for instance, whenever a VNF node with the desired protocol stack (Taleb et al., 2016) is deployed in the network. Thereby, virtualization leads to the creation of configurable forwarding nodes compatible with the SDN paradigm in the SON Data Plane layer. The virtualized switching nodes, implemented in Open vSwitch (OVS) (Dro, 2015b), incorporate support

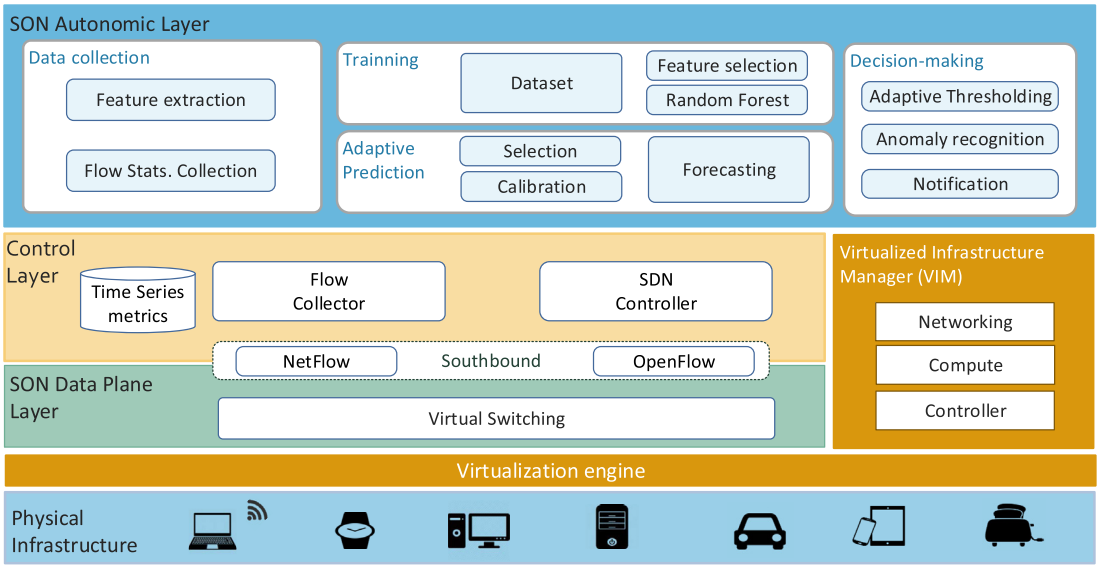


Fig. 1. Architecture for Flooding-based DDoS detection on 5G.

for the OpenFlow protocol (McKeown et al., 2008) in the southbound interface, thus handling the configuration messages with the SDN controller. In addition, the virtual switches provide support for NetFlow, a well-known protocol used for monitoring flow-based statistics (Hofstede et al., 2014a) built upon the matching of the source IP, destination IP, and protocol fields an IPv4 packet. Those statistics/counters are then analyzed in the autonomic layer when performing the detection tasks. NetFlow implements flow-sampling methods which have faced some scalability issues (Li et al., 2016) in large network deployments, as it was observed with other flow monitoring methods. Note that such scalability and efficiency concerns remain as a traffic engineering challenge, as evidenced by the research literature (Akyildiz et al., 2014; Su et al., 2015; Yang et al., 2017). Despite this drawback, the introduced architecture has opted for NetFlow since the detection strategy prioritizes the collection of accurate flow-based metrics collection, as proposed by (Giotis et al., 2014), rather than addressing specific efficiency or scalability requirements. Notwithstanding, we have mitigated those limitations by provisioning the OVS instances with larger allocation of memory and computing capacity, with the aim to enhance its performance. Both OpenFlow and Netflow interact with the OpenDaylight (ODL) controller (Medved et al., 2014) placed in the Control Layer. The OpenFlow interface manages the configuration of the flow-tables for packet forwarding in the data plane, whilst the Netflow collector gathers flow counters from the virtual switches. In addition, the proposed detection method relies on the representation of flow metrics as time series, being them stored in the database implemented by the Time Series Data Repository (TSDR) project (Dro, 2015c).

On the highest level of the architecture, the SON Autonomic layer is composed by data processing modules targeted on the core detection strategy implementation, which spans from the data collection to the notification of network threats. Flow statistics are gathered by querying the Time Series Metrics database under different granularities, and the resultant sampled metrics are aggregated by feature extraction methods (i.e. entropy measurement). Conducting a supervised machine learning approach requires a Training stage, being its main goal the construction of a classification model fitted for selecting the proper prediction algorithm. Such modeling is performed considering a set of time series features taken from the reference datasets. The Adaptive Prediction stage relies on that classification model to infer the most accurate prediction algorithm based on the time series characteristics extracted from the monitored observations. Once selected, the predictive algorithm is calibrated for minimizing the forecasting error. Then, the DDoS detection task is carried out by constructing the adaptive thresholds estimated on the basis of the predicted values. Hence, anomalies are inferred when the monitored observations are found outside the prediction boundaries, thus generating DDoS alerts notified as the outcome of this self-protective approach.

4. Denial of service indicators

Throughout the performed research different levels of information processing have been studied, which entailed the need for extracting very heterogeneous features that facilitate the analysis of the knowledge acquired from the monitored devices, that being analyzed as univariant time series. They are summarized in Table 1 and described throughout this section.

4.1. Training features

The first FlowSentinel analytic stage has focused on the extraction of traits that allow defining usage models adaptable to changes in the monitoring environment. For the automation of deciding the best suited modeling and prediction strategies, more than 100 metrics per time series sample were constructed by the tool TSFRESH, which has been developed under the iPREDICT (Dro, 2015d) project. This collection takes into account from basic statistical attributes (peaks,

Table 1
FlowSentinel DDoS indicators.

Symbol	Description
Cumulative	
nP_{in}	Incoming packets
nP_{out}	Outgoing packets
nB_{in}	Incoming bytes
nB_{out}	Outgoing bytes
Flow Disorder	
$H(nP_{in})$	Entropy packets per incoming flow
$H(nP_{out})$	Entropy packets per outoging flow
$H(nB_{in})$	Entropy bytes per incoming flow
$H(nB_{out})$	Entropy bytes per outgoing flow
Flow Divergence	
$nMSE(nP)$	Diff. Input and output packets
$nMSE(nB)$	Diff. Input and output bytes

maximum/minimum observations, mode, etc.) to correlation measures related with the evolution of the time series (white noise, trend, seasonality, autocorrelation coefficients, etc.). They were directly applied on the collection M3-Competition (Makridakis and Hibon, 2000) at the training stage of the system.

4.2. Basic metrics

The incoming/outgoing traffic flows from the protected IoT devices are monitored and structured in IPFIX format (Hofstede et al., 2014b), according to which each traffic flow f_t being a bunch of packets captured in certain time interval t, $t > 0$ that share the following properties: same source IP address (IP_{source}), IP destination ($IP_{destination}$) and port ($Port_{destination}$):

$$f_t = [IP_{source}, IP_{destination}, Port_{source}, Port_{destination}] \quad (1)$$

Let the set of traffic flows $F_i = \{f_1, f_2, \dots, f_j\}$, where each flow f_k, $0 < k \leq j$, was registered at the observation $t = i, i > 0$. The following time series (Ts) summarizes the last T, $t_1, \dots, t_T$, monitored observations:

$$Ts = \{\{f_1, f_2, \dots, f_{j1}\}_{(t=1)}, \{f_1, f_2, \dots, f_{j2}\}_{(t=2)}, \dots, \{f_1, f_2, \dots, f_{jT}\}_{t=T}\} \quad (2)$$

that are expressed as $Ts = \{F_t : t \in T\}$. Note that the $\varsigma_{(t_{ini}, t_{end})}$ timeslots:

$$\varsigma_{(1,2)}, \varsigma_{(2,3)}, \dots, \varsigma_{(T-1,T)};\ \varsigma_{(i,i+1)} = |(t_i) - (t_{i+1})| \quad (3)$$

delimit the traffic flows monitorization and establish the granularity of the analytic tasks to be performed, in this way serving as adjustment parameters that configure the sensitivity level of the detection methods. For example, when the granularity is high, the information to be processed is hardly filtered or softened, since it is often acquired from less instances (packets). As a result, these observations are more likely to pose outliers or noise. However, when the granularity is too low, it is possible that the analytic tasks overlook relevant situations. The first of these scenarios results in a more restrictive adjustment, where the detection of threats is prioritized in opposition to the generation of false positives. In the second case, the quality of the user experience is prioritized at the expense of decreasing the level of protection offered.

The following pair of measurements is taken per traffic flow: number of transferred packets nP and total amount of information transferred nB (bytes). Let the flow f_i related with the source-end with IP Address referred as $IP_{endpoint}$, its number of transferred packets is computed as follows:

$$nP(f_i) = \sum_{i=1}^{j} p_i, IP_{source}(p_i) = IP_{endpoint}(p_i)\ or\ IP_{destination}(p_i) = IP_{endpoint}(p_i) \quad (4)$$

consequently, p_i must belong to at least one of the following groups: input packets P_{in} and/or output packets P_{out}, that must satisfy the expressions:

$$p_i \in P_{in} \longleftrightarrow IP_{destination}(p_i) = IP_{endpoint}(p_i) \quad (5)$$

$$p_i \in P_{out} \longleftrightarrow IP_{source}(p_i) = IP_{endpoint}(p_i) \quad (6)$$

Similarly, its number of transferred bytes is calculated as indicated below:

$$nB(f_i) = \sum_{i=1}^{j} p_i(bytes), IP_{source}(p_i) = IP_{endpint}(p_i)\ or\ IP_{destination}(p_i) = IP_{destination}(p_i) \quad (7)$$

where B_{in} and B_{out} accumulate the number of bytes within incoming and outgoing packets. From $nP(f_i)$ and $nB(f_i)$ the aggregated metrics described in the next subsection are inferred. Note that the effectiveness of both nP and nB was continuously proven in the bibliography, highlighting their relevance at emergent communication environments (Wang et al., 2015) and at source-end DDoS defense (Mirkovic and Reiher, 2005). This led us to hypothesize that they settle a proper baseline for source-side flooding-based DDoS recognition on 5G environments, as will be demonstrated at the forthcoming sections.

4.3. Aggregated metrics

As suggested in the bibliography, the basic flow-level metrics are aggregated based on the relationship between outgoing and incoming traffic and their dispersion (Gil and Poletto, 2001). In the first case, the normalized Mean Squared Error (MSE) is considered, which is expressed as follows:

$$nMSE(X) = \frac{\frac{1}{n}\sum_{i=1}^{n}(x(a)_i - x(b)_i)^2}{\sigma^2} \quad (8)$$

where X is the trait to be analyzed, n is the total number of traffic flows with paired IP source and IP destination (i.e. the traffic incoming/outgoing traffic between a and b), $x(a)_i$ is the metric registered at the incoming traffic grouped at the flow a, and $x(b)_i$ is the metric registered at the outgoing traffic at b. A clear example is illustrated in the relationship $E_\tau(nP_{in}, nP_{out})$ that describes the difference between incoming packets $X_{in}(a) = nP_{out}(b)$ and outgoing packets $X_{out}(a) = nP_{in}(b)$ captured at the time interval τ.

On the other hand, the disorder degree of the observations is measured based on the normalized entropy described by Shannon. This decision is supported by previous research works related with conventional DDoS recognition, which successfully approached similar problems in the same way (Dro, 2018a). We hypothesized about this metric being also valid for detection at single source-side device monitorization. As is usual in the bibliography, the entropy implemented by FlowSentinel is inferred from the following expression:

$$H(X) = \frac{-\sum_{i=1}^{n} p_i \log_a p_i}{\log_a n} \quad (9)$$

where n is the total number of monitored flows captured at the time interval τ, and $p_1, p_2, \dots, p_n$ are the probabilities of the instances $x_1, x_2, \dots, x_n$ of the random variable X, the latest constructed from basic flow-level metrics. For example, let the disorder of bytes per flow $H(B)_\tau$ at the timeslot τ. If $H(B)_\tau = 0$ it is possible to assert that X_t is deterministic. On the opposite case, when $H(B)_\tau = 1$, X_t registers the maximum disorder degree.

5. Source-side floding-based DDoS detection

FlowSentinel bases its detection strategy on studying univariate time series built from aggregated metrics, which are deduced from both traffic monitored at the protected devices and collections of reference time series with training and validation purposes (at the performed experimentation, the M3-Competition (Makridakis and Hibon, 2000) dataset). To this end, three major data processing stages are distinguished: Training, Adaptive Prediction and Decision-making (see Fig. 2). At Training stage, the criteria that facilitates deciding the predictive models that best adapt to the data to be analyzed are defined from the reference samples. At the Adaptive Prediction stage, the modeling strategies are calibrated aiming on improving the forecasts to be made from the next observations. Therefore, the TSFRESH features extracted from the time series to be analyzed, lie in the grounds of the forecast models that drive the inference of next observations. Finally, FlowSentinel decides the significance of the registered prediction errors at the Decision-making stage, hence leading to discover unexpected behaviors (discordant) that provide suspicious activity indicators. The following describes in detail each FlowSentinel data processing stage.

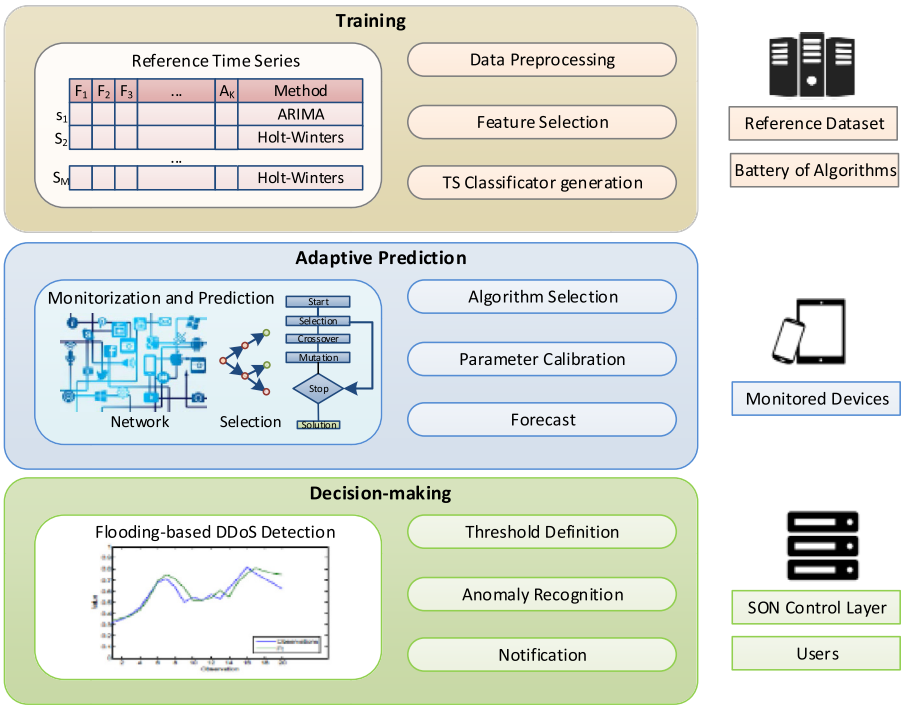


Fig. 2. FlowSentinel data processing stages.

5.1. Training and forecast method detection

The analytic dedicated server provides a battery of forecasting procedures implemented in the SELFNET analysis component (Sotelo Monge et al., 2017b), which gathers among others, models based on moving averages, autoregression or smoothing. In order to adapt the detection strategy to the non-stationarity of the monitoring environment, prior to infer the traffic behavior the best suited prediction method is selected and properly calibrated. Therefore, a collection of reference samples is required from which it is possible to extract the most relevant characteristics and build the classifier that stablishes the best forecast function (Dro, 2015d). The battery of predictive algorithms considered is summarized in Table 2. Note that for simplicity all they bring well-known solutions, being postponed the inclusion of more sophisticated approaches, like Extreme Learning Machines (ELM) and other Artificial Neural Network (ANN) families, to future work.

At the training stage, the classifier that decides the best prediction method is constructed (see Fig. 3). This illustration represents the complete FlowSentinel training process, which involves two major steps: 1) feature extraction and sample labeling, and 2) classifier construction. The first involves feature extraction, forecast algorithm selection and calibration, and from them, the second builds a model that represents the normal traits of the monitoring environment. They will serve for estimating the similarity of the monitored observation regarding the expected traffic patterns. The current version of FlowSentinel adopts as classification procedure the Random Forest approach described by Breiman (2001), where a classifier consisting of a collection of tree-structured predictors is implemented, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. In particular, the original solution implemented the variation of Classification And Regression Trees (CART) (Rutkowski et al., 2014) that choose which variable to split on using a greedy algorithm that minimizes error. This task typically requires specifying several adjustment parameters, for example, the maximum number of iterations to be performed (stop condition), the number of trees to construct, or their maximum depth. But as highlighted by Breiman, the number m of randomly selected attributes is the only adjustable parameter to which Random Forests are somewhat sensitive (assuming that there are no computational limitations, as is the case of the SELFNET Analytical dedicated server). This value determines the correlation between each pair of trees and the strength of each individual tree. By increasing the aforementioned parameter, both correlation and strength increase. When the correlation grows, the forest error rate increases; in the opposite, when the strength grows, the

Table 2
Battery of forecasting algorithms.

Method	Symbol	Type
Cumulative Moving Average (Mendes, 2015)	CMA	Moving Average
Simple Moving Average (Mulloy, 1994a)	SMA	Moving Average
Double Moving Average (Mulloy, 1994a)	DMA	Moving Average
Weighted Moving Average (Mulloy, 1994a)	WMA	Moving Average
Simple Exponential Smoothing (Aly et al., 2015)	EMA	Moving Average
Double Exponential Moving Average (Mulloy, 1994b)	DEMA	Moving Average
Triple Exponential Moving Average (Mulloy, 1994a)	TEMA	Moving Average
Simple Exponential Smoothing (Brown, 1957)	SES	Smoothing
Double Exponential Smoothing (Gardner, 2006)	DES	Smoothing
Triple Exponential Smoothing (Winters, 1960)	TES	Smoothing
Autoregressive Integrated Moving Average (Hillmer and Tiao, 1980)	ARIMA	Autoregression

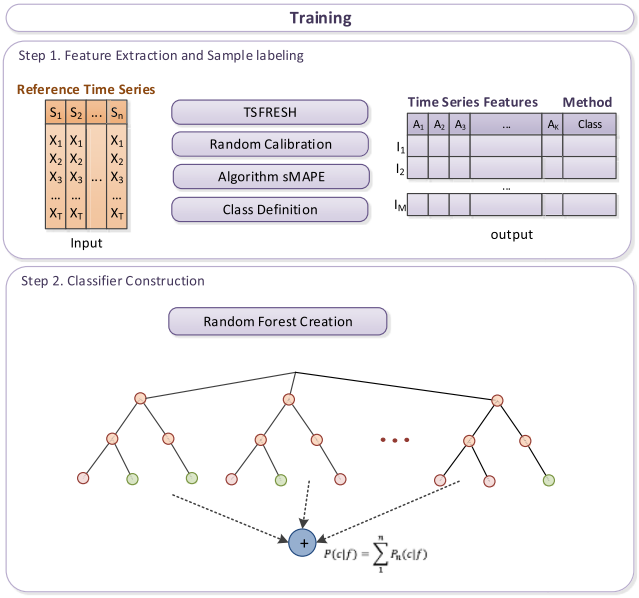


Fig. 3. FlowSentinel Training stage.

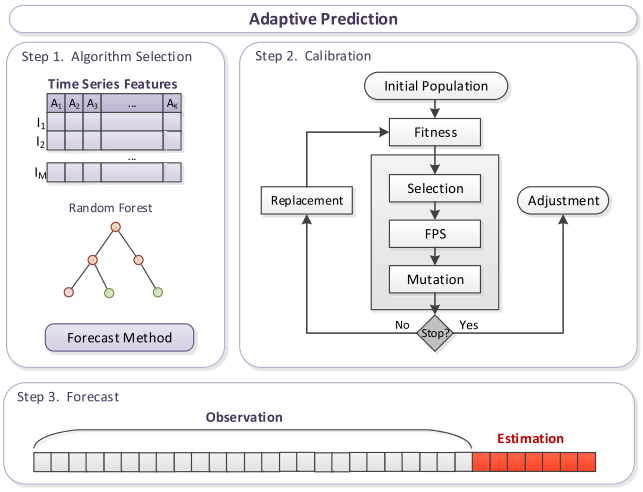


Fig. 4. FlowSentinel Adaptive Prediction stage.

forest error rate decreases, so the level of both features must be balanced. FlowSentinel addresses this problem by applying the solution proposed by Breiman (i.e. $m = \log M + 1$, where M is the number of features of samples within the dataset), hence postponing for future versions the implementation of alternative calibration strategies.

Each training sample considered for Random Forest definition is represented by the 100 TSFRESH attributes extracted from the reference time series. The instance belongs to the class that represents the prediction algorithm that registered less significant forecasting errors when building its prediction model. The class of each sample is obtained by analyzing the time series with the complete prediction algorithm battery provided by the SELFNET Analyzer (Sotelo Monge et al., 2017b). The solution that resulted in the lower Symmetric Mean Absolute Percentage Error (sMAPE) becomes the label of the instance, being this value calculated as follows:

$$sMAPE = 200\% \sum_{t=1}^{n} \frac{|\hat{x_t} - x_t|}{|\hat{x_t}| + |x_t|} \tag{10}$$

where sMAPE is the prediction ($\hat{x_t}$) minus actuals (x_t) divided by the sum of forecasts and actuals. It is self-constrained by 200% error rate as typically considered in the bibliography. Note that the sMAPE criterion was previously adopted among others by the M3-Competition (Makridakis and Hibon, 2000), in this way enabling the evaluation of the effectiveness of different forecasting procedures. Finally, it should be remarked that one of the main disadvantages of the Random Forest classifiers is their trend to overfitting. FlowSentinel reduces this problem by including a pre-selection step conducted by the greedy algorithm for feature discrimination described in Hu et al. (2018) and its evaluation based on the significance of the prediction errors (Hall, 1999).

5.2. Adaptive prediction

Holte denoted in (Holte, 1993) that pattern recognition conventionally considered that the reference datasets applied for training purposes are representative of the expected observations at the monitored environment. The presence of gradual changes over time in the statistical characteristics of the class to which an observation belongs are the result of non-stationary fluctuations, which leads among others to the *concept drift* problem (Ditzler et al., 2015) (i.e. the models built at training do no longer represent the situations that they originally intended depict). From the standpoint of the research community, the stationarity in communication networks is questionable (Masugi, 2009), which only should be assumed in very specific circumstances (Markelov et al., 2017). Because of this, and in order to provide an effective defense against DDoS in any emerging scenario, FlowSentinel operates assuming non-stationarity. It is important to highlight that O Reilly et al. (2014) distinguished two major approaches to this problem: passive and active. The active solutions require the previous recognition of inflection points that foresee relevant changes on the monitored environment, from them updating the previously built models. Because of their modus operandi, the active solutions are usually refereed as *detection and response* methods. On the other hand, the passive approach assumes that the monitored feature distribution steadily varies over time, hence demanding the continuous recalibration of the analytic capabilities. Therefore, while active solutions focus on punctual drift distinction, the passive approaches proved greater effectiveness when forecasting gradual drift and recurring concepts (Widyantoro et al., 2003). The existence of stealthy flooding-based DDoS threats based on hiding abrupt variations in the data volume injected (Ficco and Rak, 2015) leads to hypothesize that with the proper data granularity, the second paradigm best suites the main objectives of the performed research, so it was implemented by FlowSentinel (the development of active/hybrid solutions is postponed for future work). FlowSentinel adapts to non-stationarity as illustrated in Fig. 4. Accordingly, the adaptive prediction capabilities are grounded in three major processing steps: 1) algorithm selection, 2) calibration and 3) forecast. The first step has as inputs the TFRESH metrics extracted from the monitored observations. From them, the Random Forest model build at training stage decides the most suitable forecasting algorithm. Then the optimal configuration of the algorithm is calculated based on the evolutionary computation paradigm. Finally, the forthcoming observations are predicted in a predefined time horizon. These steps are described in greater detail below.

5.2.1. Forecasting algorithm selection

The proposed adaptive prediction approach decides the most suitable forecast algorithm based on the study of the TSFRESH features extracted from the monitored time series. As Illustrated in Fig. 4, this set of characteristics serves as input of the Random Forest previously built at Training stage. The resultant class refers to the best forecast method, on which the expected behavior of the network is estimated. This procedure is repeated on each observation, so the prediction method will vary as the traffic distribution changes. For example, let the time series T_s that represent the number of incoming bytes (nB_{in}) of certain end-point. The Random Forest classifier initially decided that, the best suited algo-

rithm according with TSFRESH metrics extracted from T_s is the Simple Exponential Smoothing (SES) (Brown, 1957). But in the next T_{s+m} observations significantly gained in trend and seasonality. In this case the likelihood of budging from Simple Exponential Smoothing (SES) to Triple Exponential Smoothing (TES) (Winters, 1960) increases, since TES behaved more accurately than SES in similar circumstances, at Training stage.

5.2.2. Calibration

Most of the prediction methods in the implemented battery of algorithms required a previous configuration, where the proper calibration of its adjustment parameters plays an essential role in the achieved performance. Because of this, once the prediction method is selected, it is calibrated driven by a basic Genetic Algorithm (GA) (Kamrani et al., 2001). This solution is inspired by the biological evolutive theories and their genetic-molecular basis. Consequently, and in contrast to other proposals with similar purposes, the genetic algorithms are probabilistic algorithms that conduct the evolution of an initial population of individuals (observations) generated from initial factual knowledge, through actions with arbitrary results (i.e. genetic mutations and gen recombination) that try to get closer to the optimal solution in each iteration, hence resembling those of the biological evolution processes. Their main drawbacks are related with high resource consumption and not guarantee of finding and optimal solution, both of them extensively discussed in the bibliography (Elsayed et al., 2014).

FlowSentinel implements a GA as solution to the forecasting algorithm calibration problem after taking into consideration different reasons, highlighting among them: the fact that GAs already posed solutions to optimization problems previously proved with calibration purposes (Ruiz et al., 2016), that they are capable of operating on vectors of adjustment parameters of different nature, and that their operation adapts to the detail level in which calibrations must be calculated. Note that the latter is especially valuable when operating in real time scenarios, hence allowing to balance accuracy and performance to satisfy the implemented security management policies.

In the implemented GA, FlowSentinel considers as evolving population the set of candidate adjustments, where each individual raises a possible solution. Their genotype represents a vector of gens in which each position contains one of the adjustment parameters of the prediction method. For example, in the case of TES a collection of four characteristics would be constituted: data smoothing factor (α), trend smoothing factor (β), seasonal change smoothing factor (γ) and forecast horizon (θ) (Winters, 1960). The initial population is randomly calculated and only the best adapted individuals hold possibilities of persisting at future generations. Note that as in nature, the fitness of an individual ponders its ability to adapt to the environment, and therefore, the probability of procreation. Therefore, the fitness function of the implemented GA returns the sMAPE calculated when the prediction algorithm is calibrated according to the genotype of an individual.

In addition, the GA performs simple crossover and uniform mutation per iteration. The first of them selects a couple of parents per crossover by Fitness Proportionate Selection (FPS) (Goldberg and Deb, 1991), then randomly deciding a swapping point and exchanging their genetical contents pivoting on such point. Consequently, the descendant individual replaces the parent with lower fitness. At the mutation stage, an arbitrary gen of the descendant is replaced by a random value. Because of the gens may present different nature, this action is constrained by the boundaries stablished by the data range of the adjustment parameter. For example, the α parameter of the TES prediction function ranges in $0 \cdots 1$, so the random uniform mutations on α must be restricted to $0 \cdots 1$. The algorithm has two stop conditions: a predefined maximum number of iterations (worst case), and some individual reaching its optimal fitness, i.e. $sMAPE = 0$.

From the prediction algorithm selected by the Random Forest classifier, as well as from its calibration according to the adjustment indicated by the best individual of the final population, the next h observations of the time series to be analyzed are estimated. The final population is temporally stored for serving as initial population for the next execution of the GA, in this way usually gaining accuracy. Note that this decision is based on the fact that most of the time series will be similar, so large changes in the adjustment values are not expected. Table 3 summarizes the main steps of the implemented GA.

Table 3
Main features of the calibration GA.

Feature	Highlights
Individual	An individual represents a possible calibration. The genotype is a vector where each gen is an adjustment parameter of the prediction algorithm.
Initial Population	At the first launch, the initial population is generated by assigning randomly values to gens. The initial population of the next observation is the final population of the previous execution.
Fitness	The sMAPE obtained by a specific calibration (Hofstede et al., 2014b).
Selection	Fitness Proportionate Selection (FPS) (Goldberg and Deb, 1991)
Crossover	Swapping gens from an arbitrary pivot.
Mutation	Uniform mutation of an arbitrary gen.
Stop Condition	The stop condition is satisfied when a previously defined maximum number of iterations is reached, or when an optimal solution is discovered.

5.3. Decision-making

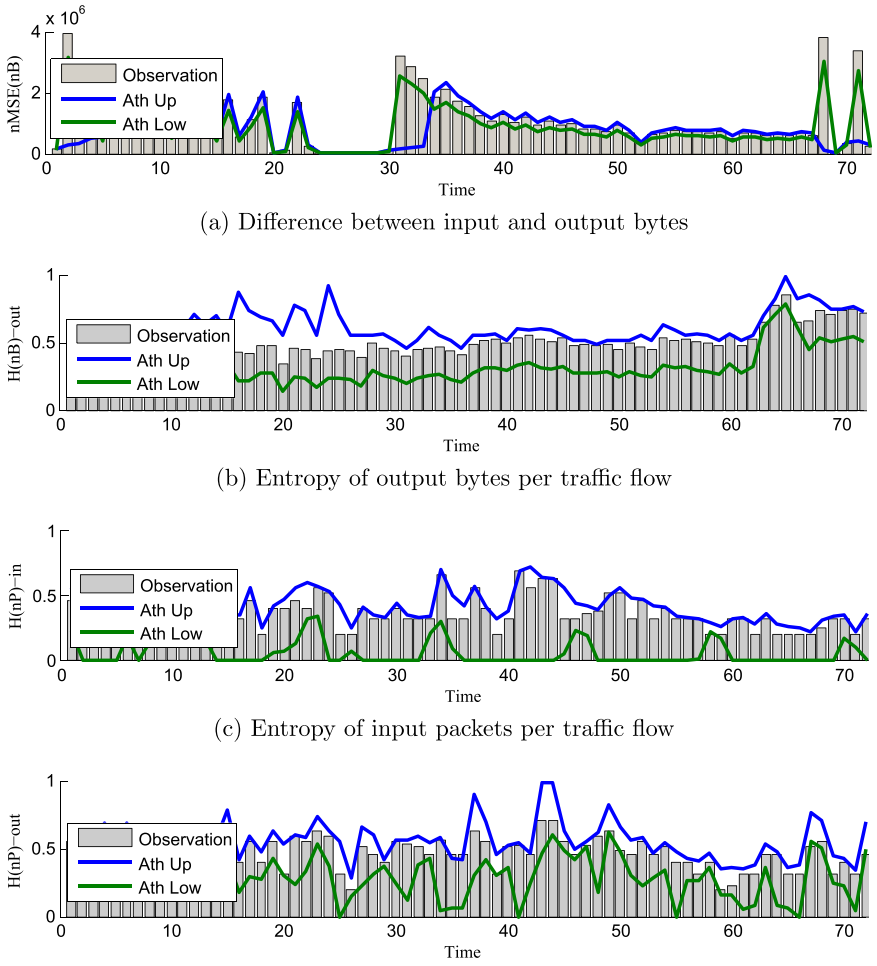
At the FlowSentinel classification stage, the natures of the time series of aggregated metrics constructed from the monitored traffic flows are decided. In this context, it is assumed that an observation is an outlier if it matches with an unexpected behavior, i.e. when the variation between a prognosis at certain time horizon and the observed value differ significantly. Because the projection of continuous values on time tends to yield errors, the main challenge of this process is to define their relevance, which is addressed by defining adaptive thresholds. In the aftermath, outliers are tagged as potential malicious behaviors, and normal situations are classified as legitimate, so the current FlowSentinel implementation acts as a binary classifier. The SELFNET Analyzer framework (Sotelo Monge et al., 2017b) provides advanced analytical capabilities related with building prediction intervals, most of them widely accepted by the research community for network traffic study. Of them, FlowSentinel integrates the adaptive thresholding methodology described in Makridakis et al. (1998), which defines the following prediction intervals:

$$Ath_{up} = \hat{x}_{n+1} + K\sqrt{\sigma^2(E_t)} \tag{11}$$

$$Ath_{low} = \hat{x}_{n+1} - K\sqrt{\sigma^2(E_t)} \tag{12}$$

where $\hat{x}_{n+1}$ is the forecast of certain aggregated metric x at $n + 1$ horizon, E_t is the Euclidean distance between $\hat{x}_{n+1}$ and x_{n+1}, and K is the adjustment parameter that configures the restrictiveness of the sensor. The equations distinguish an upper threshold Ath_{up} and a lower threshold Ath_{low}, both adapted to t. It is expected that the greater values of K, the higher noise tolerance, since this situation expands the margin of error between $\hat{x}_{n+1}$ and x_{n+1}. In the opposite case, FlowSentinel increases the protection level, which typically occurs at the expense of penalizing the false positive rate.

An example of the decision-making behavior in a real use case is illustrated in Fig. 5, where part of the time series of aggregated metrics monitored from the device L-D-40 (see SubSection 6.1) with 15 seconds of data granularity were analyzed. In particular, the evolution of four of the considered flooding-based DDoS indicators is observed: difference between input and output bytes (Fig. 5a), entropy of output



(a) Difference between input and output bytes

(b) Entropy of output bytes per traffic flow

(c) Entropy of input packets per traffic flow

(d) Entropy of output packets per traffic flow

Fig. 5. Example of outlier identification caused by flooding-bases DDoS.

bytes per traffic flow (Fig. 5b), entropy of input packets per traffic flow (Fig. 5c), and entropy of output packets per traffic flow (Fig. 5d). Only at the first of them discordances were detected, since the constructed adaptive thresholds were surpassed. Consequently, the proposal issued a notification of a potential threat.

6. Experimentation

In order to assess the effectiveness of FlowSentinel, different experiments have been conducted on traffic traces monitored from end-point devices of different nature. The gathered sample collection and the applied experimentation methodology are described below.

6.1. Dataset

The collection of samples gathered for FlowSentinel evaluation includes outgoing traffic captures from 61 different devices. Each sample was created from traffic monitorizations separated in time periods of 1, 3 and 5 days, comprising a total amount of 150 instances with 3 hours per device, so the dataset contains 72,400 samples of normal traffic. At the end of each normal traffic capture, the tools described in Dro (2017b, 2017c) launched various DDoS attacks; in particular, traffic injections based on UDP, HTTP or TCP flooding with low, medium and high intensities. Accordingly, the dataset provides 78,300 samples with malicious contents. The activities they represent are summarized in Table 4. They are daily user activities (general-purpose actions), synthetic web navigation with various automatization tools and multimedia streaming. The samples with traces labeled as general-purpose activities mainly coincide with miscellaneous office work and the use that the volunteers make day by day of their devices. The synthetic web browsing groups mainly contain traffic generated by random web navigation bots, which were separated according to the one used; A for Internet Noise (Dro, 2018b), B for Noiszy (Dro, 2018c) and C for TrackMeNot (Dro, 2018e). The streaming groups represent traffic from

Table 4
Monitored activities and endpoint devices at the performed experimentation.

No.	No.	Samples	p-ADF
Daily user habits	18	2700	0.103
Browser bot A	4	600	0.065
Browser bot B	14	2100	0.130
Browser bot C	4	600	0.008
Audio streaming	5	900	0.040
Video streaming	13	1950	0.065
Endpoint	**No.**	**Normal**	**Attack**
Desktop computer	24	3600	32,400
Notebook	18	2700	24,300
Smartphone	8	1200	10,800
Tablet computer	5	750	6750
Smartwatch	2	300	2700
Smart TV	1	150	1350

Table 5
Devices monitored at the experimentation.

ID	Report Date	Device	Activities	Characteristics
L-B-A-1	2018-04-01_02.05.48	PC	Browser bot A (P1)	I5-7400 x64 8 Gb Ram (Ubuntu 16.04)
L-B-A-2	2018-04-01_02.05.48	PC	Browser bot A (P1)	i3-8100 x64 4 GB Ram (Windows 10)
L-B-A-3	2018-03-31_20.01.06	PC	Browser bot A (P1)	i3-4170 x32 4 GB Ram (Windows 7)
L-B-A-4	2018-03-31_20.01.06	PC	Browser bot A (P1)	i7-6700K x64 16 GB Ram (Windows 10)
L-B-B-5	2018-04-04_17.51.41	PC	Browser bot B (P2)	i3-7350k x64 8 GB Ram (Windows 10)
L-B-B-6	2018-04-04_17.51.41	PC	Browser bot B (P2)	i5-6600K x64 8 GB Ram (Debian)
L-B-B-7	2018-04-04_17.51.41	PC	Browser bot B (P2)	i5-6600K x64 8 GB Ram (Debian)
L-B-B-8	2018-04-04_17.51.41	PC	Browser bot B (P2)	i3-8100 x64 4 GB Ram (Windows 10)
L-B-B-9	2018-04-04_17.51.41	PC	Browser bot B (P2)	i5-6600K x64 8 GB Ram (Debian)
L-B-B-10	2018-04-04_17.51.41	PC	Browser bot B (P2)	i3-7350k x64 8 GB Ram (Windows 10)
L-B-B-11	2018-04-04_17.51.41	PC	Browser bot B (P2)	I5-7400 x64 8 Gb Ram (Ubuntu 16.04)
L-B-B-12	2018-04-04_17.51.41	PC	Browser bot B (P2)	i5-6600K x64 8 GB Ram (Debian)
L-B-B-13	2018-04-05_20.47.19	PC	Browser bot B (P2)	I5-7400 x64 8 Gb Ram (Ubuntu 16.04)
L-B-B-14	2018-03-30_21.39.14	PC	Browser bot B (P2)	i3-8100 x64 4 GB Ram (Windows 10)
L-B-B-15	2018-03-30_21.39.14	PC	Browser bot B (P2)	i7-6700K x64 16 GB Ram (Windows 10)
L-B-C-16	2018-04-02_04.25.53	PC	Browser bot C (P3)	i5-6600K x64 8 GB Ram (Debian)
L-B-C-17	2018-04-02_04.25.53	PC	Browser bot C (P3)	I5-7400 x64 8 Gb Ram (Ubuntu 16.04)
L-B-C-18	2018-04-03_02.58.40	PC	Browser bot C (P3)	i3-7350k x64 8 GB Ram (Windows 10)
L-B-C-19	2018-04-03_02.58.40	PC	Browser bot C (P3)	i5-6600K x64 8 GB Ram (Debian)
L-D-20	2018-04-09_11.34.44	PC	Daily user habits (P0)	I5-7400 x64 8 Gb Ram (Ubuntu 16.04)
L-D-21	2018-04-09_11.34.45	PC	Daily user habits (P0)	i7-6700K x64 16 GB Ram (Windows 10)
L-A-22	2018-04-09_03.18.55	PC	Audio streaming (P4)	I5-7400 x64 8 Gb Ram (Ubuntu 16.04)
L-A-23	2018-04-07_15.31.07	PC	Audio streaming (P4)	i3-8100 x64 4 GB Ram (Windows 10)
L-V-24	2018-03-30_13.49.21	PC	Video streaming (P5)	I5-7400 x64 8 Gb Ram (Ubuntu 16.04)
L-V-25	2018-03-30_13.49.21	PC	Video streaming (P5)	i3-8100 x64 4 GB Ram (Windows 10)
L-V-26	2018-03-30_03.59.14	PC	Video streaming (P5)	i7-6700K x64 16 GB Ram (Windows 10)
L-V-27	2018-03-30_03.59.14	PC	Video streaming (P5)	i5-6600K x64 8 GB Ram (Debian)
L-V-28	2018-03-29_15.47.07	PC	Video streaming (P5)	i5-6600K x64 8 GB Ram (Debian)
L-V-29	2018-03-29_15.47.07	PC	Video streaming (P5)	i3-7350k x64 8 GB Ram (Windows 10)
L-B-B-30	2018-03-30_04.02.19	Notebook	Browser bot B (P2)	Toshiba i7-6500 x64 8 Gb Ram (Win10)
L-B-B-31	2018-03-31_07.44.03	Notebook	Browser bot B (P2)	Hp 470 G4 4 GB Ram (Win10)
L-B-B-32	2018-03-31_07.44.03	Notebook	Browser bot B (P2)	Acer Aspire 3 4 GB Ram (Windows 7)
L-D-33	2018-03-30_13.19.31	Notebook	Daily user habits (P0)	ASUS K541UA-GQ610T (Windows 8)
L-D-34	2018-03-30_13.19.31	Notebook	Daily user habits (P0)	Toshiba i7-6500 x64 8 Gb Ram (Win10)
L-D-35	2018-03-29_18.06.20	Notebook	Daily user habits (P0)	Hp 470 G4 4 GB Ram (Win10)
L-D-36	2018-03-29_18.06.20	Notebook	Daily user habits (P0)	Toshiba i5-6200 8 GB Ram (Ubuntu)
L-D-37	2018-03-31_07.44.03	Notebook	Daily user habits (P0)	Acer Aspire 3 4 GB Ram (Windows 7)
L-D-38	2018-03-31_07.44.03	Notebook	Daily user habits (P0)	Lenovo i3-6006U 8 GB Ram (Windows 10)
L-D-39	2018-03-31_07.44.03	Notebook	Daily user habits (P0)	Toshiba i5-6200 8 GB Ram (Ubuntu)
L-D-40	2018-03-31_07.44.03	Notebook	Daily user habits (P0)	ASUS K541UA-GQ610T (Windows 8)
L-D-41	2018-04-08_15.37.35	Notebook	Daily user habits (P0)	Lenovo i3-6006U 8 GB Ram (Windows 10)
L-V-42	2018-03-30_00.11.09	Notebook	Video streaming (P5)	Toshiba i7-6500 x64 8 Gb Ram (Win10)
L-V-43	2018-04-08_11.40.26	Notebook	Video streaming (P5)	ASUS K541UA-GQ610T (Windows 8)
L-V-44	2018-04-08_11.40.26	Notebook	Video streaming (P5)	Hp 470 G4 4 GB Ram (Win10)
L-V-45	2018-03-30_19.18.02	Notebook	Video streaming (P5)	Acer Aspire 3 4 GB Ram (Windows 7)
L-B-B-46	2018-03-29_21.50.49	Smartphone	Browser bot B (P2)	BQ Aquaris B5 (Android)
L-B-B-47	2018-03-29_21.50.49	Smartphone	Browser bot B (P2)	Asus Zenfone Laser (Android)
L-D-48	2018-04-05_17.08.14	Smartphone	Daily user habits (P0)	Samsung Galaxy A3 (Android)
L-D-49	2018-04-10_16.20.20	Smartphone	Daily user habits (P0)	Xiaomi Redmi Note 4 (Android)
L-D-50	2018-04-11_15.00.52	Smartphone	Daily user habits (P0)	Samsung Galaxy A3 (Android)
L-D-51	2018-04-11_15.00.52	Smartphone	Daily user habits (P0)	Huawey Honor 6X (Android)
L-D-52	2018-04-06_19.42.03	Smartphone	Daily user habits (P0)	Xiaomi Redmi Note 4 (Android)
L-D-53	2018-04-06_19.42.03	Smartphone	Daily user habits (P0)	Asus Zenfone Laser (Android)
L-D-54	2018-04-03_16.21.08	Tablet	Daily user habits (P0)	Asus 380M 1 GB-Ram (Android)
L-D-55	2018-04-04_16.38.49	Tablet	Daily user habits (P0)	Asus Zenpad Z500M 4 GB Ram (Android)
L-D-56	2018-04-04_16.38.49	Tablet	Daily user habits (P0)	Xiaomi MiPad 2 GB Ram (Android)
L-V-57	2018-04-08_19.30.14	Tablet	Video streaming (P5)	Asus 380M 1 GB-Ram (Android)
L-V-58	2018-04-08_19.30.14	Tablet	Video streaming (P5)	Asus Zenpad Z500M 4 GB (Android)
L-V-59	2018-04-08_19.30.14	Tablet	Video streaming (P5)	BQ Aquaris M10 2 GB (Android)
L-A-60	2018-04-08_16.12.34	SmartWatch	Audio streaming (P4)	Xiaomi Amazfit 2 DualCore 1.2 GHz 128 MB
L-A-61	2018-04-08_16.12.34	SmartWatch	Audio streaming (P4)	Xiaomi Amazfit 2 DualCore 1.2 GHz 128 MB
L-V-62	2018-04-08_22.06.58	Smart TV	Video streaming (P5)	Samsung UE43M5575AU Full HD Smart TV

devices that throughout the sampling mostly made use of multimedia streaming services, hence emphasizing those related with audio contents (Spotify, Apple Music, etc.) and video (YouTube, Twitch, etc.). Table 4 indicates the 6 families of devices that have been considered during the experimentation: desktop computers, notebooks, smartphones, tablet computers, smartwatches and Smart TVs. Note that given that in terms of traffic modeling, the type of the end-point had less impact than its usage model, the conducted study primarily focused on their behavior. It also displays the average p-value of the Augmented Dickey-Fuller test (ADF) that assess the no-stationarity of each traffic profile (Cheung and Kon, 1995). The p-values lower than 0.05 resemble stationary processes, which leads us to assume that most of the endpoints behave as non-stationary data sources. Finally, it is important to highlight that with the motivation of encouraging the research in this research topic, the full dataset is available at (Dro, 2018a). The monitored devices are enumerated in Table 5 and the collected samples are summarized in Fig. 6.

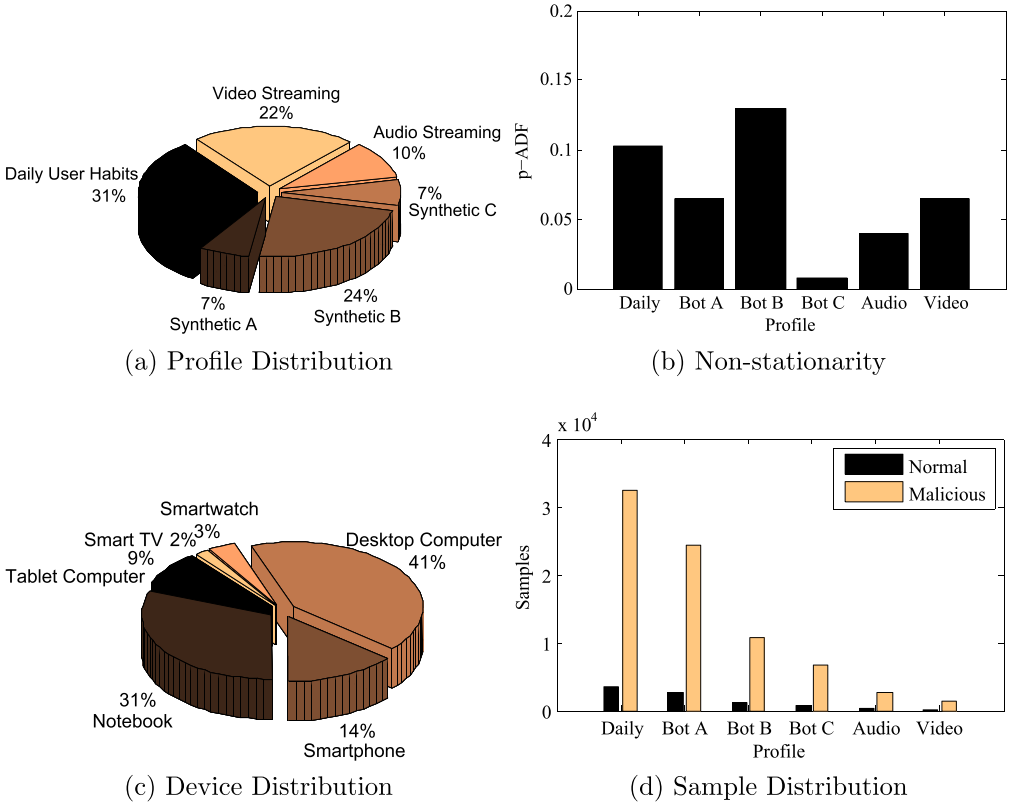


Fig. 6. Summary of the dataset.

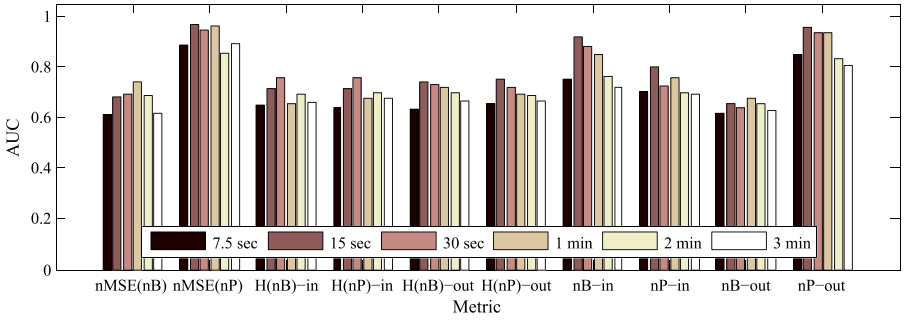
Table 6
AUC registered per observation granularity when varying K.

Indicator	Observation granularity					
	7.5 Sec.	15 Sec.	30 Sec.	1 Min.	2 Min.	3 Min.
nP_{in}	0.75	0.79	0.72	0.75	0.69	0.69
nP_{out}	0.84	0.95	0.93	0.93	0.83	0.80
nB_{in}	0.61	0.65	0.63	0.67	0.65	0.62
nB_{out}	0.75	0.91	0.87	0.84	0.76	0.72
$H(nP_{in})$	0.63	0.71	0.75	0.67	0.69	0.67
$H(nP_{out})$	0.65	0.74	0.72	0.69	0.68	0.66
$H(nB_{in})$	0.64	0.71	0.75	0.65	0.69	0.65
$H(nB_{out})$	0.63	0.73	0.72	0.71	0.69	0.66
nMSE(nP)	**0.88**	**0.96**	**0.94**	**0.95**	**0.85**	**0.89**
$nMSE(nB)$	0.60	0.68	0.68	0.74	0.68	0.61

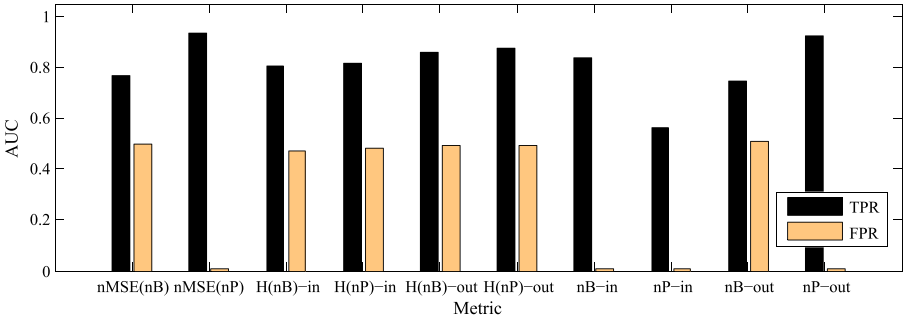
Best(15 s)		
TPR	FPR	Y
0.56	0.01	0.55
0.92	0.01	0.91
0.74	0.51	0.23
0.83	0.01	0.82
0.81	0.48	0.33
0.87	0.49	0.38
0.80	0.47	0.33
0.85	0.49	0.36
0.93	**0.01**	**0.92**
0.76	0.50	0.26

6.2. Evaluation methodology

The effectiveness of the proposed method has been tested by adopting an experimental evaluation methodology, in which the impact on effectiveness was measured when varying the following adjustment parameters: metric, restriction level, granularity and attack intensity. Assuming that the main goal of the performed research intends to provide a novel and effective flooding-based DDoS detection approach, the results described in this paper focus on proving and discussing the accuracy achieved when analyzing IoT traffic of different nature. Because of this, the monitored activities were labeled as *legitimate* (normal) and *malicious* (discordant). In analogy with previous publications, this task was addressed considering FlowSentinel as a binary classifier, hence based on observing its sensitivity and specificity



(a) AUC per metric and granularity



(b) Accuracy on the best granularity

Fig. 7. Results when varying the data granularity.

(Makridakis et al., 1998). The first of them determines the ability to properly point out anomalies as *malicious*. On the other hand, the specificity measures the ability of recognizing normal activities as *legitimate*. From their representation in the ROC (Receiver Operating Characteristic) space, several effectiveness indicators have been extracted, standing out for relevance the Area Under the Curve (AUC), and the True Positive Rates (TPR) and False Positive Rates (FPR) achieved from the best sensor adjustment in terms of K. As is frequent in the bibliography, the latter coincides with the position of the ROC curve that displays the better Youden index (Bantis et al., 2014). Based on these criteria, the following three experiments have been accomplished:

1. *Impact of granularity*. At this experiment, the accuracy of the sensor was measured when studying traffic flows captured in time intervals of 7.5 s, 15 s, 30 s, 1 min, 2 min and 3 min.
2. *Impact of the end-points activity*. Configured with the best granularity of the previous test, this experiment analyzes the accuracy of FlowSentinel based on the activities usually performed by the compromised devices.
3. *Impact of attack intensity*. This test verifies under what circumstances the flooding-based DDoS threats are more easily detectable by the sensor, hence distinguishing protocol (HTTP, TCP and UDP) and the flooding capabilities of the attack.

7. Results and discussion

The following describes and discusses the obtained results when varying the data granularity, usage mode of the inspected devices and attack intensity.

7.1. Impact of granularity

At this experiment, the accuracy of the sensor was measured when studying traffic flows captured in time intervals of 7.5 s, 15 s, 30 s, 1 min, 2 min and 3 min; hence only focusing on the monitorization interval and the adjustment of the K parameter for adaptive thresholding calibration. The accuracy achieved per metric and configuration is illustrated in Table 6 and Fig. 7a, where the effectiveness of FlowSentinel is expressed in terms of AUC. This performance indicator was calculated via trapezoidal approximation with 0.005 estimated error. The best studied granularity was 15 seconds per observation (see 7b), which provides the most accurate results (AUC = 0.96, TPR = 0.93 and FPR = 0.01). When the granularity is lower (i.e. the duration of the observation is smaller), the FlowSentinel accuracy worsens. For example, when 7.5 seconds the best registered AUC was 0.88. Similarly, as the level of detail falls, the effectiveness of the proposal decreases, hence reaching AUC = 89.2 when 3 minutes per observations. This is due to the fact that with small observations the information they compile tends to be less significant, hence being more likely to infer noise. On the contrary, when the observation is too large, the first observations of the attack may go unnoticed among legitimate traffic. In this case, the adaptation of FlowSentinel to non-

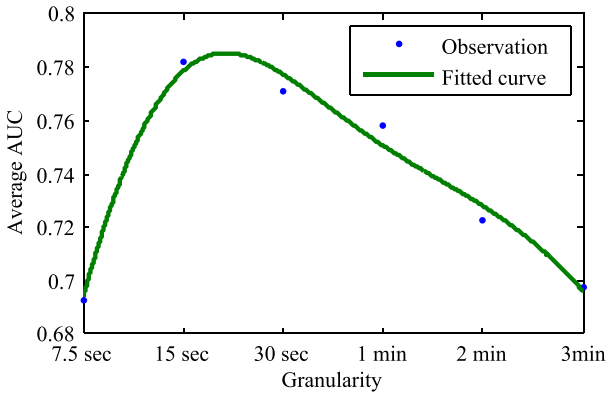


Fig. 8. Gaussian estimation of average AUC per granularity.

Table 7
AUC registered per traffic profile at 15 s granularity.

Indicator	Traffic usage profile					
	P_0	P_1	P_2	P_3	P_4	P_5
nP_{in}	0.86	0.84	0.86	0.94	0.70	0.84
nP_{out}	0.96	0.96	0.96	0.97	0.94	0.96
nB_{in}	0.79	0.71	0.51	0.93	0.81	0.67
nB_{out}	0.92	0.88	0.95	0.93	0.92	0.93
$H(nP_{in})$	0.73	0.54	0.80	0.84	0.57	0.67
$H(nP_{out})$	0.73	0.76	0.79	0.72	0.63	0.67
$H(nB_{in})$	0.70	0.58	0.79	0.83	0.56	0.67
$H(nB_{out})$	0.71	0.78	0.80	0.64	0.64	0.66
nMSE(nP)	**0.96**	**0.97**	**0.97**	**0.97**	**0.96**	**0.96**
nMSE(*nB*)	0.84	0.76	0.36	0.91	0.86	0.72

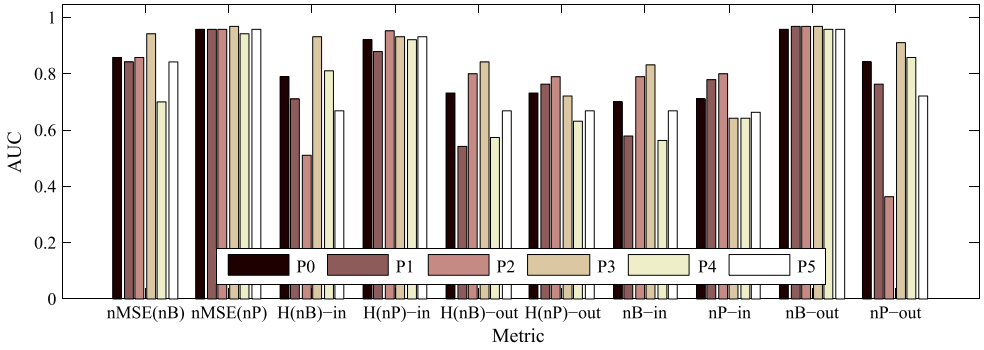


Fig. 9. AUC per metric and behavioral profile.

stationarity readjusted the analytic algorithms, so if the attack is not initially detected, it may be considered part of the normal activity of the network. These situations are highlighted in Fig. 8, where a Gaussian approximation with 0.005 confidence interval was plotted from the average AUC per granularity; in particular, a turning point is clearly distinguished in the improvement obtained by decreasing the observation interval.

Finally, it is worth to highlight the accuracy achieved by metrics directly related to the total incoming (nP_{in}) and outcoming (nP_{out}) packet, which divergence (*nMSE*(*nP*)) behaved as the most accurate DDoS indicator at the performed experimentation. In contrast with the classical entropy-based DDoS detection solutions focused on intermediate/victim edge audition, these metrics proved not to be as effective at single source-side monitorization.

7.2. Impact of traffic profile

With the purpose of facilitating the understanding of the achieved results, the impact of the device usage mode on the FlowSentinel effectiveness has been studied when assuming the best granularity of the previous experimentation, as it is illustrated in Table 7. The six traffic activity profiles described in Table 4 and Fig. 9 were analyzed, hence leading to the following best results: user daily habits P_0 (AUC = 0.96), A synthetic traffic P_1 (AUC = 0.97), B synthetic traffic P_2 (AUC = 0.97), C synthetic traffic P_3 (AUC = 0.97), audio streaming P_4 (AUC = 0.96) and video streaming P_5 (AUC = 0.96). Note that similarly to the previous tests, the best metric is often the difference between incoming and outgoing packets (*nMSE*(*nP*)), which is closely followed by the total number of incoming packets (nP_{in}) and the total number of outgoing packages (nP_{out}). Again entropy-based metrics have not been effective enough. As can be observed in the box plot described in Fig. 10, the best metrics provided similar results regardless of the usage profile. Consequently, since no significant variations have been recorded between traffic profiles, it is possible to conclude that FlowSentinel was capable of self-calibrating according to the traffic distribution inherent to each type of endpoint, in this way posing an effective solution regardless the nature of the device.

7.3. Impact of attack type

Table 8 and Fig. 11a summarize the accuracy obtained by threat group, which include flooding-based low-rate attacks on HTTP (*H*), TCP (*T*), UDP (*U*) protocols. As illustrated in Fig. 11c, they have been clustered based on intensity, hence distinguishing three subsets: high inten-

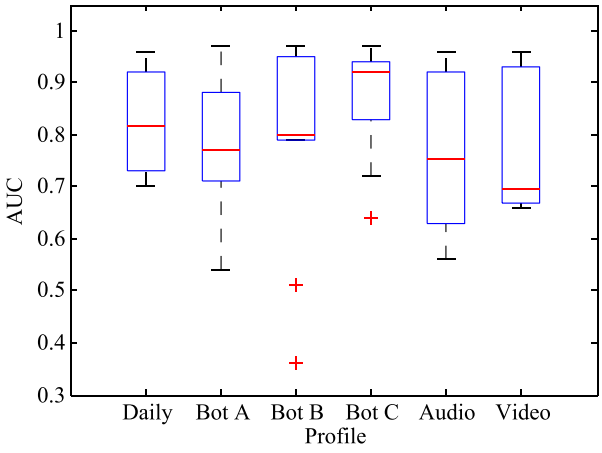
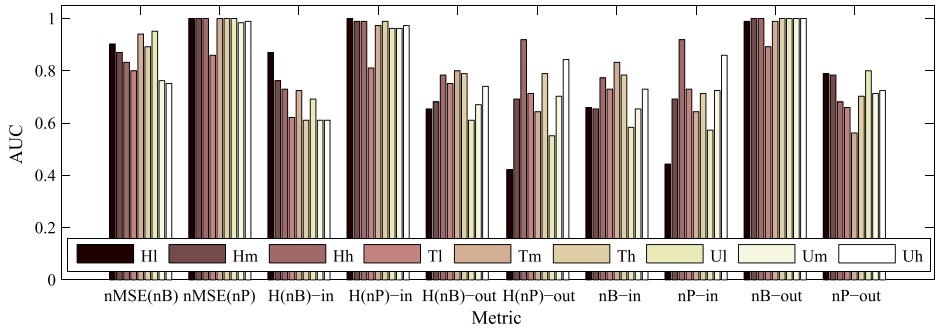


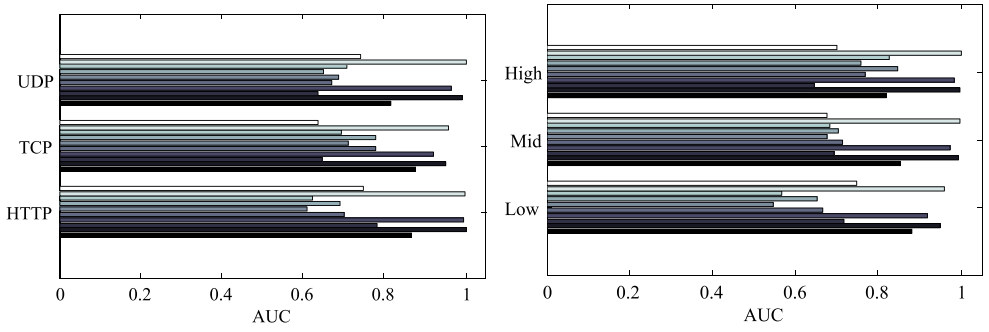
Fig. 10. Distribution of results per behavioral profile.

Table 8
AUC registered per attack type at 15 s granularity.

Indicator	Attack type								
	H_l	H_m	H_h	T_l	T_m	T_h	U_l	U_m	U_h
nP_{in}	0.90	0.87	0.83	0.80	0.94	0.89	0.95	0.76	0.75
nP_{out}	1.00	1.00	1.00	0.86	1.00	1.00	1.00	0.98	0.99
nB_{in}	0.87	0.76	0.73	0.62	0.72	0.61	0.69	0.61	0.61
nB_{out}	1.00	0.99	0.99	0.81	0.97	0.99	0.96	0.96	0.97
$H(nP_{in})$	0.65	0.68	0.78	0.75	0.80	0.79	0.61	0.67	0.74
$H(nP_{out})$	0.42	0.69	0.92	0.71	0.64	0.79	0.55	0.70	0.84
$H(nB_{in})$	0.66	0.65	0.77	0.73	0.83	0.78	0.58	0.65	0.73
$H(nB_{out})$	0.44	0.69	0.92	0.73	0.64	0.71	0.57	0.72	0.86
nMSE(nP)	**0.99**	**1.00**	**1.00**	**0.89**	**0.99**	**1.00**	**1.00**	**1.00**	**1.00**
nMSE(nB)	0.79	0.78	0.68	0.66	0.56	0.70	0.80	0.71	0.72



(a) AUC per metric and attack



(b) Accuracy on attack type (c) Accuracy on attack intensity

Fig. 11. Results when varying the attack intensity.

sity (h), medium (m) and low (l). For example, the symbol T_l refers to the group of TCP attacks of low intensity. In general terms, the effectiveness was better than at previous tests, where the metrics *nMSE*(*nP*), nP_{in} and nP_{out} outstand. The best AUC ranged from 0.99 to 1.0 regardless the intrusion subset. This obvious improvement is empowered by a fundamental characteristic of the test: the K adjustment factor that was applied by FlowSentinel for configuring its restriction level now is set to detect a specific menace; this did not happen at the second experiment, where the same threshold distance was configured for all the DDoS methods. In Fig. 11b the results are compared per attack family, where no significant differences were observed between the best metrics, which is also demonstrated in Fig. 12.

Consequently, it is possible to deduce that the proposed method has been able to adapt to each attack group. However, as the threat specificity decreases FlowSentinel tends to lose precision. This must be taken into account when proposing general-purpose self-organizing defenses, where it might be advisable to deal with the different intrusion categories separately.

During the experimentation, the nature of the injected attack did not demonstrate to play a crucial role in the effectiveness of the proposal. This is a direct result of the modus operandi of the launched threats: flooding the victim with a vast amount of information to be processed; which overall, could be managed regarding the protocol involved (HTTP, TCP, UDP). Nonetheless, there are considerations that could be inferred from these results, and that somehow, may support further research.

Firstly, it is worth to highlight that aggregated metrics solely based on studying incoming traffic were more effective than those uniquely based on outgoing traffic. This is due to the fact that, the response to malicious requests was not required, thus a priori being more significant. Metrics considering both input and output traffic were able to lead FlowSentinel more accurately, since the exchange of control traffic allowed to distinguish merely downloading/uploading data streams from targeted attacks. Finally, *nP*-based metrics clearly outperformed *nB*-based metrics. This was not a surprising fact, since DDoS injection tools typically randomize the generation of the packets payloads.

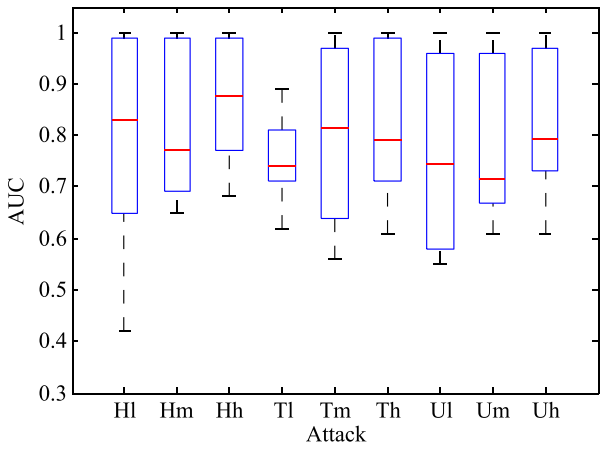


Fig. 12. Distribution of results per attack type.

8. Conclusions

This contribution revealed a research line aimed on detecting flooding-based DDoS attacks by analyzing source-side traffic flows from protected devices, in this way supporting the development of defensive SON solutions grounded on endpoint monitorization. To this end, an autonomic architecture with operability on the emerging communication networks and a novel intrusion detection approach adaptable to non-stationary processes, were introduced. Their effectiveness has been proven at an extensive experimentation, where traffic from 61 devices of different nature was monitored and analyzed looking for DDoS traits. The obtained results confirm the alternative hypothesis of the research, thus demonstrating that it is possible to raise similar solutions to the challenges inherent to our object of study. Another interesting finding is that the proposal behaved almost indistinctly on different network usage profiles, whether they pose legitimate or malicious activities.

But not all the metrics were equally effective. For example, those based on studying the proportionality between incoming and outgoing traffic yielded promising results, from which it follows that solutions like (Hoque et al., 2015; Dro, 2018a) could be successfully accommodated for operating on heterogeneous and non-stationary network environments. In contrast, the classical entropy-based approaches for DDoS detection at intermediate/victim edges were not as effective. This fact brings uncertainty about their accuracy when acting at source-side observations, more particularly at non-stationary contexts.

One of the main drawbacks of the proposal observed during the experimentation is its proven tendency to loss precision as its specificity grows, i.e. when it is trained to act against a larger variety of attacks. This feature raises an interesting line of future research that leads to encourage outlining ensemble learning methods for providing a general-purpose defensive solution. Throughout the paper alternative ways of improvement have been highlighted, being of special interest those based on expanding the diversity of metrics, prediction algorithms and adjustment parameters. It is expected that as a result of these enhancements, a greater effectiveness may be registered. This also contributes to the clearer understanding of the studied traffic profiles, as well as to assess their impact on intrusion detection at forthcoming communication networks.

Acknowledgements

The authors want to thank the support of the SELFNET (A Framework for Self-Organized Network Management in Virtualized and Software Defined Networks) project, which was funded by the European Commission Horizon 2020 Programme under Grant Agreement number H2020-ICT-2014-2/671672.

References

Acarali, D., Rajarajan, M., Kmminos, N., Herwono, I., 2016. Survey of approaches and features for the identification of HTTP-based botnet traffic. J. Netw. Comput. Appl. 76, 1–15.
Agiwal, M., Roy, A., Saxena, N., 2016. Next generation 5G wireless networks: a comprehensive survey. IEEE Commun. Surv. Tutor. 18 (3), 1617–1655.
Akyildiz, I., Lee, A., Wang, P., Luo, M., Chou, W., 2014. A roadmap for traffic engineering in SDN-OpenFlow networks. Comput. Network. 71, 1–30.
Al-Yaseen, W., Othman, Z., Nazri, M., 2017. Real-time multi-agent system for an adaptive intrusion detection system. Pattern Recogn. Lett. 85, 56–64.
Alenezi, N., Reed, M., 2014. Uniform DoS traceback. Comput. Secur. 45 (1), 17–26.
Almeida, V., Doneda, D., Abreu, J., 2017. Cyberwarfare and digital governance. IEEE Internet Comput. 21 (2), 68–71.
Aly, A., Salem, N., Mahmoud, M.A., Woodall, W., 2015. A reevaluation of the adaptive exponentially weighted moving average control chart when parameters are estimated. Qual. Reliab. Eng. Int. 31 (8), 1611–1622.
Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., Zhou, Y., 2017. Understanding the Mirai botnet. In: Proceedings of the 26th USENIX Security Symposium. Vancouver, BC, Canada.
Bantis, L., Nakas, C., Reiser, B., 2014. Construction of confidence regions in the ROC space after the estimation of the optimal Youden index-based cut-off point. Biometrics 70 (1), 212–223.
Behal, S., Kumar, K., Sachdeva, M., 2018. D-face: an anomaly based distributed approach for early detection of ddos attacks and flash events. J. Netw. Comput. Appl. 11, 49–63.
Bertino, E., Islam, N., 2017. Botnets and internet of things security. Computers 50 (2), 76–79.
Bhuyan, M., Bhattacharyya, D., Kalita, J., 2015. An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. Pattern Recogn. Lett. 51 (1), 1–7.
Braga, R., Mota, E., Passito, A., 2010. Lightweight ddos flooding attack detection using nox/openflow. In: Local Computer Networks (LCN), 2010 IEEE 35th Conference on. IEEE, pp. 408–415.
Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.
Brown, R., 1957. Exponential smoothing for predicting demand. Oper. Res. 5 (1), 145.
CCN-CERT, 2017. IA-16/17 CyberThreats-Trends, 2017 Edition. https://www.ccn-cert.cni.es/en/reports/.
Chadd, A., 2018. Ddos attacks: past, present and future. Netw. Secur. 2018 (7), 13–15.
Cheung, Y., Kon, S., 1995. Lag order and critical values of the augmented Dickey-Fuller test. J. Bus. Econ. Stat. 13 (3), 277–280.
Demestichas, P., Georgakopoulos, A., Karvounas, D., Tsagkaris, K., Stavroulaki, V., Lu, J., Xiong, C., Yao, J., 2013. 5G on the horizon: key challenges for the radio-access network. IEEE Veh. Technol. Mag. 8 (3), 47–53.
Denning, D., 2014. Framework and principles for active cyber defense. Comput. Secur. 40, 108–113.
Ditzler, G., Roveri, M., Alippi, C., Polikar, R., 2015. Learning in nonstationary environments: a survey. IEEE Comput. Intell. Mag. 10 (4), 12–25.
Open Source Software for Creating Private and Public Clouds, 2015a. https://www.openstack.org.
Open vSwitch, 2015b. https://www.openvswitch.org/.
Opendaylight TSDR, 2015c. https://wiki.opendaylight.org/view/Project_Proposals:Time_Series_Data_Repository.
TSFRESH: Time Series Feature Extraction Based on Scalable Hypothesis Tests, 2015d. https://github.com/blue-yonder/tsfresh.
5g Ppp Architecture Working Group View on 5g Architecture, 2017a. https://5g-ppp.eu/wp-content/uploads/2017/07/5G-PPP-5G-Architecture-White-Paper-2-Summer-2017_For-Public-Consultation.pdf. version 2.0.
Low Orbit Ion Cannon (LOIC), 2017b. https://sourceforge.net/projects/loic/files/.
WarChild DoS Test Suit, 2017c. https://github.com/Souhardya.
DroidSentinel, 2018a. https://github.com/borjalor/DroidSentinel.
Internet Noise, 2018b. http://makeinternetnoise.com.
Noiszy, 2018c. https://noiszy.com.
SELFNET: Self-Organized Network Management in Virtualized and Software Defined Networks, 2018d. http://www.SELFNET-5g.eu.
TrackMeNot: Resisting Surveillance in Web Search, 2018e. https://cs.nyu.edu/trackmenot/.
Elsayed, S., Sarker, R., Essam, D., 2014. A new genetic algorithm for solving optimization problems. Eng. Appl. Artif. Intell. 27, 57–69.
Ferguson, P., Senie, D., 2000. Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing. https://tools.ietf.org/html/rfc2827.
Ficco, M., Rak, M., 2015. Stealthy denial of service strategy in cloud computing. IEEE Trans. Cloud Comput. 3 (1), 80–94.
Gardner, E., 2006. Exponential smoothing: the state of the art Part II. Int. J. Forecast. 22 (4), 637–666.
Gavrilovska, L., Rakovic, V., Atanasovski, V., 2016. Visions towards 5G: technical requirements and potential enablers. Wireless Pers. Commun. 87 (3), 731–757.
Gil, T., Poletto, M., 2001. MULTOPS: a data-structure for bandwidth attack detection. In: Proceedings of the 10th USENIX Security Symposium, vol. 10, Washington, DC, US.
Giotis, K., Argyropoulos, C., Androulidakis, G., Kalogeras, D., Maglaris, V., 2014. Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. Comput. Network. 62, 122–136.

Goldberg, D., Deb, K., 1991. A comparative analysis of selection schemes used in genetic algorithms. Found. Genet. Algorithms 1, 69–93.
Hall, M.A., 1999. Correlation-Based Feature Selection for Machine Learning.
Hillmer, S., Tiao, G., 1980. An ARIMA-model-based approach to seasonal adjustment. J. Am. Stat. Assoc. 77 (377), 63–70.
Hofstede, R., Celeda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., Pras, A., 2014a. Flow monitoring explained: from packet capture to data analysis with netflow and ipfix. IEEE Commun. Surv. Tutor. 16 (4), 2037–2064.
Hofstede, R., Celeda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., Prass, A., 2014b. Flow monitoring explained: from packet capture to data analysis with NetFlow and IPFIX. IEEE Commun. Surv. Tutor. 16 (4), 2037–2064.
Holgado, P., Villagrá, V., Vázquez, L., 2017. Real-time multistep attack prediction based on Hidden Markov Models. IEEE Trans. Dependable Secure Comput., https://doi.org/10.1109/TDSC.2017.2751478.
Holte, R., 1993. Very simple classification rules perform well on most commonly used datasets. Mach. Learn. 11 (1), 63–90.
Hoque, N., Bhattacharyya, D., Kalita, J., 2015. Botnet in DDoS attacks: trends and challenges. IEEE Commun. Surv. Tutor. 17 (4), 2242–2270.
Hossain, E., Hasan, M., 2015. 5G cellular: key enabling technologies and research challenges. IEEE Instrum. Meas. Mag. 18 (3), 11–21.
Hu, B., Li, X., Sun, S., Ratcliffe, M., 2018. Attention recognition in EEG-based affective learning research using CFS + KNN algorithm. IEEE ACM Trans. Comput. Biol. Bioinform 15 (1), 38–45.
Imran, M., Durad, M., Khan, F., Derhab, A., 2019. Toward an optimal solution against denial of service attacks in software defined networks. Future Gener. Comput. Syst. 92, 444–453.
Jhaveri, M., Cetin, O., Ganan, C., Moore, T., Van Eeten, M., 2017. Abuse reporting and the fight against cybercrime. ACM Comput. Surv. 49 (4) (2).
Jin, R., Wang, B., 2013. Malware detection for mobile devices using software-defined networking. In: Proceedings of the 2nd GENI Research and Educational Experiment Workshop, pp. 81–88 Salt Lake City, UT, US.
Kamrani, A., Wang, R., Gonzalez, R., 2001. A genetic algorithm methodology for data mining & intelligent knowledge acquisition. Int. J. Forecast. 40 (4), 361–377.
Khanna, S., Venkatesh, S., Fatemieh, O., Gunter, C., 2011. Adaptive selective verification: an efficient adaptive countermeasure to thwart DoS attacks. IEEE/ACM Trans. Netw. 20 (3), 715–728.
Kiremire, A., Brust, M., Phoha, V., 2014. Using network motifs to investigate the influence of network topology on PPM-based IP traceback schemes. Comput. Network. 72 (1), 14–32.
Kolias, C., Kambourakis, G., Stavrou, A., Voas, J., 2017. DDoS in the IoT: Mirai and other botnets. Computers 50 (7), 80–84.
Lal, S., Taleb, T., Dutta, A., 2017. Nfv: security threats and best practices. IEEE Commun. Mag. 55 (8), 211–217.
Lateef, H., Imran, A., Imran, M., Giupponi, L., Dohler, M., 2015. LTE-advanced self-organizing network conflicts and coordination algorithms. IEEE Wireless Commun. 22 (3), 108–117.
Li, C., Yang, J., Wang, Z., Li, F., Yang, Y., 2015. A lightweight DDoS flooding attack detection algorithm based on synchronous long flows. In: Proceedings of the IEEE Global Communications Conference (GLOBECOM), pp. 1–6 San Diego, CA, US.
Li, Y., Miao, R., Kim, C., Yu, M., 2016. FlowRadar: a better NetFlow for data centers. In: Proceedings of the 13th Usenix Symposium on Networked Systems Design and Implementation, pp. 311–324 Santa Clara, CA, US.
Luo, H., Lin, Y., Zhang, H., Zukerman, M., 2013. Preventing DDoS attacks by identifier/locator separation. IEEE Netw. 27 (6), 60–65.
MacFarland, D., Shue, C., Kalafut, A., 2017. The best bang for the byte: characterizing the potential of DNS amplification attacks. Comput. Network. 116, 12–21.
Maestre Vidal, J., Orozco, A., Villalba, L.J.G., 2018. Adaptive artificial immune networks for mitigating DoS flooding attacks. Swarm Evolut. Comput. 38, 94–108.
Maimo, L., Gomez, A., Clemente, F., Perez, M., Perez, G., 2018a. A self-adaptive deep learning-based system for anomaly detection in 5G networks. IEEE Access 6, 7700–7712.
Maimo, L.F., Gomez, A.L.P., Clemente, F.J.G., Gil Perez, M., Martinez Perez, G., 2018b. A self-adaptive deep learning-based system for anomaly detection in 5g networks. IEEE Access 6, 7700–7712.
Makridakis, S., Hibon, M., 2000. The M3-Competition: results, conclusions and implications. Int. J. Forecast. 16 (4) 451–176.
Makridakis, S., Wheelwright, S., Hyndman, R., 1998. Forecasting: Methods and Applications. John Wiley & Sons.
Mamolar, A.S., Pervez, Z., Calero, J.M.A., Khattak, A.M., 2018. Towards the transversal detection of ddos network attacks in 5g multi-tenant overlay networks. Comput. Secur. 79, 132–147.
Markelov, O., Duc, V., Bogachev, M., 2017. Statistical modeling of the Internet traffic dynamics: to which extent do we need long-term correlations? Phys. A Stat. Mech. Appl. 485, 48–60.
Masugi, M., 2009. Applying a recurrence plot scheme to analyze non-stationary transition patterns of IP-network traffic. Commun. Nonlinear Sci. Numer. Simul. 14 (4), 1418–1430.
Matta, V., Di Mauro, M., Longo, M., 2017. DDoS attacks with randomized traffic innovation: botnet identification challenges and strategies. IEEE Trans. Inf. Forensics Secur. 12 (8), 1844–1859.
McKeown, N., Anderson, T., Balakrishan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J., 2008. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. 38 (2), 69–74.
Medved, J., Varga, R., Tkacik, A., Gray, K., 2014. Opendaylight: towards a model-driven sdn controller architecture. In: Proceedings of the 15th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–6 Zhangjiajie, China.
Mehdi, S., Khalid, J., Khayam, S., 2011. Revisiting traffic anomaly detection using software defined networking. In: Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID), pp. 161–180 Menlo Park, CA, US.
Mendes, P., 2015. Combining data naming and context awareness for pervasive networks. J. Netw. Comput. Appl. 50, 114–125.
Mirkovic, J., Reiher, P., 2005. D-ward: a source-end defense against flooding denial-of-service attacks. IEEE Trans. Dependable Secure Comput. 2 (3), 216–232.
Mulloy, P., 1994a. Smoothing data with faster moving averages. Stocks Commod. 12 (1), 11–19.
Mulloy, P., 1994b. Smoothing data with less lag. Tech. Anal. Stocks Commod. 12 (1).
Neves, P., Cale, R., Costa, M., Gaspar, G., Alcaraz-Calero, J., Wang, Q., Nightingale, J., Bernini, G., Carrozzo, G., Valdivieso, A., et al., 2017. Future mode of operations for 5g: the selfnet approach enabled by sdn/nfv. Comput. Stand. Interfac. 54, 229–246.
O Reilly, C., Gluhak, A., Imran, M., Rajasegarar, S., 2014. Anomaly detection in wireless sensor networks in a non-stationary environment. IEEE Commun. Surv. Tutor. 16 (3), 1413–1432.
Ozcelik, I., Brooks, R., 2015. Deceiving entropy based DoS detection. Comput. Secur. 48 (1), 234–245.
Palattella, M., Dohler, M., Grieco, A., Rizzo, G., Torsner, J., Engel, T., Ladid, L., 2016. Internet of things in the 5G era: enablers, architecture, and business models. IEEE J. Sel. Area. Commun. 34 (3), 510–527.
Ruiz, G., Bandera, C., Temes, T., Gutierrez, A., 2016. Genetic algorithm for building envelope calibration. Appl. Energy 168, 691–705.
Rutkowski, L., Jaworski, M., Pietruczuk, L., Duda, P., 2014. The CART decision tree for mining data streams. Inf. Sci. 266, 1–15.
Sahoo, K., Puthal, D., Tiwary, M., Rodrigues, J., Sahoo, B., Dash, R., 2018. An early detection of low rate ddos attack to sdn based data center networks using information distance metrics. Future Gener. Comput. Syst. 89, 685–697.
Saied, A., Overill, R., Radzik, T., 2016. Detection of known and unknown DDoS attacks using Artificial Neural Networks. Neurocomputing 172, 385–393.
Semerci, M., Cemgil, A., Sankur, B., 2018. An intelligent cyber security system against ddos attacks in sip networks. Comput. Network. 136, 137–154.
Sicari, S., Rizzardi, A., Miorandi, D., Coen-Porisini, A., 2018. Reato: reacting to denial of service attacks in the internet of things. Comput. Network. 137, 37–48.
Sotelo Monge, M., Maestre Vidal, J., Villalba, L., 2017a. Entropy-based economic denial of sustainability detection. Entropy 19 (12) (649).
Sotelo Monge, M., Maestre Vidal, J., Villalba, L., 2017b. Reasoning and knowledge acquisition framework for 5G network analytics. Sensors 17 (10) (2405).
Su, Z., Wang, T., Xia, Y., Hamdi, M., 2015. CeMon: a cost-effective flow monitoring system in software defined networks. Comput. Network. 92, 101–115.
Taleb, T., Ksentini, A., Jantti, R., 2016. Anything as a service for 5G mobile systems. IEEE Netw. 30 (6), 84–91.
Thilak, K., Amuthan, A., 2018. Cellular automata-based improved ant colony-based optimization algorithm for mitigating ddos attacks in vanets. Future Gener. Comput. Syst. 82, 304–314.
Vormayr, G., Zseby, T., Fabini, J., 2017. Botnet communication patterns. IEEE Commun. Surv. Tutor. 19 (4), 2768–2796.
Wang, Z., 2019. An elastic and resiliency defense against ddos attacks on the critical dns authoritative infrastructure. J. Comput. Syst. Sci. 99, 1–26.
Wang, R., Jia, Z., Ju, L., 2015. An entropy-based distributed DDoS detection mechanism in software-defined networking. In: Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 310–317 Helsinki, Finland.
Wang, K., Du, M., Maharjan, S., Sun, Y., 2017a. Strategic honeypot game model for distributed denial of service attacks in the Smart grid. IEEE Trans. Smart Grid 8 (5), 2474–2482.
Wang, Y., Chen, I., Cho, J., Swami, A., Chan, K., 2017b. Trust-based service composition and binding with multiple objective optimization in service-oriented mobile ad hoc networks. IEEE Trans. Serv. Comput. 10 (4), 660–672.
Wang, C., Miu, T., Luo, X., Wang, J., 2018a. SkyShield: a sketch-based defense system Against application layer DDoS attacks. IEEE Trans. Inf. Forensics Secur. 13 (3), 559–573.
Wang, L., Li, Q., Jiang, Y., Jia, X., Wu, J., 2018b. Woodpecker: detecting and mitigating link-flooding attacks via sdn. Comput. Network. 147, 1–13.
Wei, W., Chen, F., Xia, Y., Jin, G., 2013. A rank correlation based detection against distributed reflection DoS attacks. IEEE Commun. Lett. 17 (1), 173–175.
Wei, L., Li, Q., Jiang, Y., Wu, J., 2016. Towards mitigating Link Flooding Attack via incremental SDN deployment. In: Proceedings of the IEEE Symposium on Computers and Communications (ISCC), pp. 397–402 Messina, Italy.
Widyantoro, D., Ioerger, T., Yen, J., 2003. Tracking changes in user interests with a few relevance judgments. In: Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM), pp. 548–551 New Orleans, LA, US.
Winters, P., 1960. Forecasting sales by exponentially weighted moving averages. Manag. Sci. 6 (3), 324–342.
Wu, H., Wang, Z., 2018. Multi-source fusion-based security detection method for heterogeneous networks. Comput. Secur. 74, 55–70.
Xiao, L., Wei, W., Yang, W., Shen, Y., Wu, X., 2017. A protocol-free detection against cloud oriented reflection DoS attacks. Soft Comput. 21 (13), 3713–3721.

Yaar, A., Perring, A., Song, D., 2006. StackPi: new packet marking and filtering mechanisms for DDoS and IP spoofing defense. IEEE J. Sel. Area. Commun. 24 (10), 1853–1863.
Yan, Q., Yu, F., Gong, Q., Li, J., 2016. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges. IEEE Commun. Surv. Tutor. 18, 602–622.
Yang, X., Han, B., Sun, Z., Huang, J., 2017. SDN-based DDoS attack detection with cross-plane collaboration and lightweight flow monitoring. In: Proceedings of the 2017 IEEE Global Communications Conference (GLOBECOM 2017) Singapore, Singapore.
Zargar, S.T., Joshi, J., Tipper, D., 2013. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Commun. Surv. Tutor. 15 (4), 2046–2069.
Zhou, L., Guo, H., 2017. Applying nfv/sdn in mitigating ddos attacks. In: Region 10 Conference, TENCON 2017-2017 IEEE. IEEE, pp. 2061–2066.

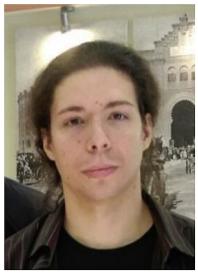
Marco Antonio Sotelo Monge holds a Bsc. in Computer Science Engineering degree from the Universidad Continental (Peru) and M.Sc. in Computer Science from the Universidad Complutense de Madrid (Spain). He is PhD in Computer Science from the same university. He is working as a full-time Researcher for the Group of Analysis, Security and Systems (GASS) at UCM. He is currently participant in the European projects SELFNET (H2020-ICT-2014-2/671672) and RAMSES (H2020-FCT-04-2015/700326). His main research interests are 5G, SDN/NFV, artificial intelligence and information security.

Andrés Herranz González was graduated in Computer Engineering by Universidad Complutense de Madrid (Spain) in 2018. He is passionate about artificial intelligence, data science and researcher. During the last year in the university, while he has been working in the Innovation area of Everis (NTT Data), he has been collaborating in the research group GASS in the SELFNET (H2020-ICT-2014-2/671672) European project with the Department of Software Engineering and Artificial Intelligence (DISIA) of the Universidad Complutense de Madrid.

Borja Lorenzo Fernández graduated in Computer Science Engineering degree by Universidad Complutense de Madrid (Spain) in 2018. Interested in computer security, research and hacking. Actually, he collaborates with the research group GASS in the SELFNET (H2020-ICT-2014-2/671672) European project at the Department of Software Engineering and Artificial Intelligence (DISIA) of the Universidad Complutense de Madrid. Simultaneously he is obtaining professional experience as an auditor in the Hacking Department at Innotec system (Entelgy).

Diego Maestre Vidal is studying his last year in Computer Science Engineering degree by Universidad Complutense de Madrid (Spain). He is collaborating with group GASS in the project SELFNET (H2020-ICT-2014-2/671672), he also got a degree of Administration of Systems and Networks. Since 2017, he is lecturer on Logix5 Smart Solutions. His main research interests are Network Security, Artificial Intelligence and Patter Recognition.

Guillermo Rius García graduated in the Bachelor Program in Computer Science Engineering at Universidad Complutense de Madrid in 2018, where he co-authored a degree dissertation which awarded him several prizes by reputed consulting companies such as Management Solutions and Sopra Steria. He further complemented his academic background with his collaboration with research group GASS at the Department of Software Engineering and Artificial Intelligence (DISIA), from the Faculty of Computer Science and Engineering at Universidad Complutense de Madrid, where he participated as fellow researcher in SELFNET (H2020-ICT-2014-2/671672) European project. Additionally, he currently combines his research activity at the aforementioned research group with his professional life at INDRA transportation department, where he works as data scientist.



Jorge Maestre Vidal (https://jmaestrevidal.com) is Senior Specialist in Cybersecurity (senior researcher) at Indra, and member of the the Department of Software Engineering and Artificial Intelligence (DISIA) of the Faculty of Computer Science and Engineering at the Complutense University of Madrid (UCM), Spain. He received a Computer Science Engineering degree from the UCM in 2012, master degree in Research in Computer Science in 2013, and PhD in Computer Science in 2018. In 2016 he was Visiting Research at Instituto de Telecomunicações (IT), Aveiro, Portugal. His academic experience includes teaching and direction of final degrees projects. In addition, he participated in projects funded by private organizations (Banco Santander, Safelayer Secure Communications S.A., etc.) and public institutions (EDA, FP7, Horizon 2020, Plan Nacional de I + D + i, Spanish Ministry of Defense, etc.). He was recently participant in the European projects SELFNET (H2020-ICT-2014-2/671672) and RAMSES (H2020-FCT-04-2015/700326), and he is an occasional collaborator with the 5G-PPP Security WG. His main research interests are Artificial Intelligence, Information Security and the emerging Communication Technologies, where he has significant background proved by publications in several research journals (Knowledge-Based Systems, Swarm and Evolutionary Computation, Journal of Network and Computer Applications, etc.), conferences (ARES, EuroS&P, ICIT, RAID, etc.), participation at international research projects (H2020, COST, CYTED), experience as peer-reviewer (Elsevier, MDPI, IEEE, Adelaide, etc.) and member of different organizing/technical committees (ICSP-AS, SDN-NGAS, ICQNM, AIR, etc.). He is also evaluator of the National Fund for Scientific and Technological Development (FONDECYT) of the Chilean National Commission for Scientific and Technological Research (CONICYT).