A Measurement Approach for Inline Intrusion Detection of Heartbleed-Like Attacks in IoT Frameworks

Andrea Amodei, *Graduate Student Member, IEEE*, Domenico Capriglione, *Senior Member, IEEE*, Gianni Cerro, *Member, IEEE*, Luigi Ferrigno, *Senior Member, IEEE*, Gianfranco Miele, *Senior Member, IEEE*, and Giuseppe Tomasso, *Member, IEEE*

***Abstract*— Cyber security is one of the most crucial aspects of the Internet of Things (IoT). Among the possible threats, great interest is today paid toward the possible capturing of information caused by external attacks on both client and server sides. Whatever the IoT application, the involved nodes are exposed to cyberattacks mainly through the vulnerability of either the sensor nodes themselves (if they have the capabilities for networking operativity) or the IoT gateways, which are the devices able to create the link between the local nodes of the IoT network, and the wide area networks. Due to the low-cost constraints typical of many IoT applications, the IoT sensor nodes and IoT gateways are often developed on low-performance processing units, in many cases customized for the specific application, and thus not easy to update against new cyber threats that are continuously identified. In the framework of cyberattacks aimed at capturing sensitive information, one of the most known was the heartbleed, which, has allowed attackers to remotely read protected memory from an estimated 24%–55% of popular HTTPS sites. To overcome such a problem, which was due to a bug of the OpenSSL, a suitable patch was quickly released, thus allowing to avoid the problem in most of the cases. However, IoT devices may require more advanced mitigation techniques, because they are sometimes unable to be patched for several practical reasons. In this scenario, the article proposes a novel measurement method for inline detecting intrusions due to heartbleed and heartbleed-like attacks. The proposed solution is based on an effective rule which does not require decoding the payload and that can be implemented on a low-performance general-purpose processing unit. Therefore, it can be straightforwardly implemented and included in either IoT sensor nodes or IoT gateways. The realized system has been tested and validated on a number of experiments carried out on a real network, showing performance comparable (in some cases better) with the heavier machine learning-based methods.**

***Index Terms*— Cyber security, intrusion detection system (IDS), low-performance systems, rule-based, sensor nodes.**

Manuscript received 8 April 2023; accepted 21 May 2023. Date of publication 5 June 2023; date of current version 20 June 2023. This work was supported in part by the "Dipartimenti di Eccellenza," Italian Ministry of Education, University and Research Funding Program; and in part by the "Competence Center Cyber 4.0," funded by the Italian Ministry of Enterprises and Made in Italy. The Associate Editor coordinating the review process was Dr. Huang-Chen Lee. *(Corresponding author: Domenico Capriglione.)*

Andrea Amodei, Domenico Capriglione, Luigi Ferrigno, Gianfranco Miele, and Giuseppe Tomasso are with the Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, 03043 Cassino, Italy (e-mail: andrea.amodei@unicas.it; capriglione@unicas.it; ferrigno@unicas.it; g.miele@unicas.it; g.tomasso@unicas.it).

Gianni Cerro is with the Department of Medicine and Health Sciences "V. Tiberio," University of Molise, 86100 Campobasso, Italy (e-mail: gianni.cerro@unimol.it).

Digital Object Identifier 10.1109/TIM.2023.3282662

I. Introduction

NETWORK security monitoring and measurements are commonly used methodologies in information security operation centers. The network traffic is captured by means of suitable measurement probes and the related logs are monitored to detect any illegal activities within the network [1], [2].

Intrusion detection systems (IDSs) are automatic systems specifically designed for identifying threats that are able to potentially create damage to information systems as data leakages, Distributed Denial of Service (DDoS), Bad Data Injection, to cite a few, and in different contexts of application [3], [4], [5], [6], [7]. Recent trends of cyberattacks go toward Internet of Things (IoT) and operational technology (OT) infrastructure which will involve more and more targets including critical infrastructures, traditional manufacturing facilities, even smart home networks, in the next years. Due to the prevalence of employees managing these systems via remote access, which provides a very good entry point for cybercriminals, it is expected that attackers will target industrial sensors to cause physical damage that could result in assembly lines shutting down or services being interrupted [8].

Among the most popular cyberattacks, heartbleed vulnerability took the Internet by surprise in April 2014 and allowed attackers to remotely read protected memory from an estimated 24%–55% of popular HTTPS sites [9], [10]. This kind of attack was due to a bug in OpenSSL, the most popular open-source cryptographic library that implements the SSL and TLS protocols. In particular, the implementation of TLS heartbeat extension had a bug allowing attackers to remotely access private data from both clients and servers. Although a suitable patch was quickly released for overcoming such a bug, recent scientific and technical literature prove how the problem of heartbleed still persists in many contexts since systems that did not (or could not) upgrade to the patched version of OpenSSL are still affected by the vulnerability and open to attack. As matter of fact, any server or cloud platform should be relatively easy to patch. However, IoT devices may require more advanced mitigation techniques, because they are sometimes unable to be patched for practical reasons. Additionally, some companies often use a customized version of the vulnerable software. As an example, in the case of heartbleed, OpenSSL is an open-source library and some

companies might have modified it for their purposes. In such cases, a direct patch is not possible–the company must then reintroduce its custom code into the new version of the library. This is often the reason why web open-source software is not immediately updated by companies even if critical new bugs are found [11].

All these considerations, together with the fact that heartbleed-like attacks can be still dangerous, push again the research activity toward defining suitable intrusion detection strategies able to identify such kinds of attacks [12], [13], [14], [15], [16], [17], [18]. Moreover, the analysis of the literature has proved how, concerning heartbleed and heartbleed-like attacks, IDSs able to inline identify such kinds of attacks are not so far available. Generally, to develop an IDS, most of the solutions are based on Artificial Intelligence (AI) and Machine Learning approaches, which generally incur a significant computational burden [19], [20], [21], nevertheless in many application contexts concerning IoT, where devices and network systems are characterized by low costs and consequently by low resources for data processing and storage, the applicability of such techniques is not always feasible.

In this framework, starting from the preliminary results provided in [22] and on the basis of the past experience in the fields of measurements systems for network performance analysis [23], [24], [25], [26], [27], [28], [29], [30], in this article, the authors propose a novel measurement method designed for inline detecting on low-performance systems, and able to identify heartbleed-like attacks on the basis of very straightforward and easy to implement rules. It falls in the class of network-based IDS and it is implemented using very low-cost hardware and open-source software for data acquisition and analysis. To evaluate the performance of the proposed system, a suitable experimental setup has been realized for operating in real operating scenarios. The experimental characterization made on several kinds of real attacks and several kinds of real standard data traffic shows very good performance in terms of the ability to correct detecting the heartbleed-like attacks and in discriminating them from the standard traffic, thus keeping the number of false alarms low.

The rest of the article is organized as follows. Section II provides a detailed analysis of heartbleed and heartbleed-like attacks; Section III describes the preliminary experimental analysis carried out for designing and tuning the proposed measurement method; Section IV reports an example of implementation of the measurement method on a common low-performance system along with the experimental results achieved on a wide measurement campaign and allows comparing the proposed solutions with ML-based techniques; Section V discusses about the possible extensions of the proposed methodology to wider networks and, finally, Section VI draws the conclusions.

II. Data Leakage Due to Heartbleed Attacks

As aforementioned, in 2014 it was discovered one of the most relevant attacks which targeted web servers all over the world. Indeed due to the widespread of OpenSSL, the open-source library that implements one of the most notorious cryptographic protocols (transport layer security and security socket layer), the heartbleed attack caused the theft of millions and millions of sensitive data stored in servers that implemented a bug version of OpenSSL. The severity of the attack pushed in less than one day releasing a patch version for troubleshooting.

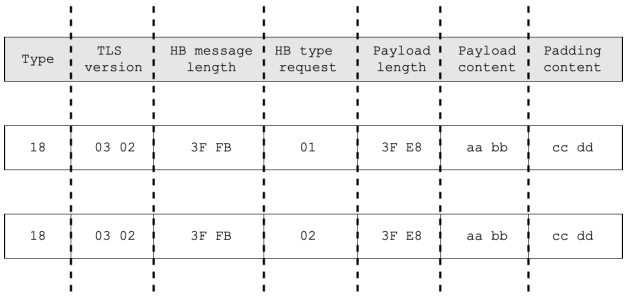


Fig. 1. Heartbeat request (HB type 01) and response (HB type 02) packets structural description.

A. Heartbleed

The reason that led the developers to adopt the TLS heartbeat eExtension was to avoid that the server side dropped the communication if a client did not send any message after an idle time, avoiding the authentication and renegotiation phases. Basically, the protocol adopts an echo mechanism in which the client sent a well-structured packet, known as *heartbeat request* and specifies the length. At the other end, the server replies with a packet, *heartbeat response*, having the same length and information value.

In detail, the heartbeat packet whether it was a request or response, is at most 16 384 bytes and has seven fields.

1) The first one is the one-byte type fixed-length and provides information about heartbeat type (0×18).
2) The second field is a two-byte field indicating the TLS version (03 01 to indicate TLSv1.0, 03 02 for TLSv1.1, 03 03 TLSv1.2).
3) The third field indicates the length of the heartbeat message.
4) The next one byte (fourth field) is the heartbeat type (0×01 for request message otherwise 0×02 for response).
5) Finally, there are two bytes to represent the length of the heartbeat payload message and payload (fifth and sixth fields), and padding (last field).

To have a correct message, the padding field must contain at least 16 bytes and the length of the heartbeat message must not exceed $2^{14} - 5$ value [9]. The packet structure is shown in Fig. 1. In particular, two aspects from the 1.0.1 and 1.0.1f OpenSSL versions that may cause some undesired consequences. The first one is related to the degree of freedom at the client side, as it has the possibility to choose the length of the heartbeat request packet in the allowable range, by declaring the packet length up to 65 536 bytes. The second one comes from the fact that the code implementing the protocol did not check the real length specified in the payload length field and the corresponding payload content.

The difference between the real packet size and the value indicated in the message represents the *heartbleed*, that was a flooding of information stored into the buffer. The structure

Type	TLS version	HB message length	HB type request	Payload length	Payload content	Padding content
18	03 02	00 03	01	FF FF		

Fig. 2. Heartbleed packet.

Type	TLS version	HB message length	HB type request	Payload length	Payload content	Padding content
18	03 02	3F FB	02	FF FF	aa bb	cc dd

Fig. 3. Heartbleed-like packet structural description.

of heartbleed-shaped packet is reported in Fig. 2. This type of attack ranks into heap buffer over-read attack.

B. Heartbleed-Like

Unlike canonical heartbleed packet, where an attacker usually specifies the length of the payload, using the maximum available value in hexadecimal form, that is $ffff_{hex}$, and keeping empty the rest of the packet, the *heartbleed-like* case was also evaluated in this article. Indeed, it is possible to send a packet having the same size as a heartbeat, but altering only the payload length. In Fig. 3 is reported the case in which the size is 16 384 bytes, but asking an echo heartbeat response containing 65 534 bytes. The reason that led us to study this new type of heartbleed attack is to trick a possible IDS based on the measurement of the packet size. More specifically, while keeping the length of the packet fixed, we have been ranging the payload length from the heartbeat to heartbleed. However, it might be intuitive that the most disruptive way to obtain the maximum data stored into the buffer is to keep the size of the packet as small as possible. Referring to our previous work [22], leveraging on the only assumption that the network traffic has a symmetric behavior between backward and forward, we upgraded the proposed rule to a more powerful one, as discussed in Section III.

III. Preliminary Experimental Assessment

The ability to distinguish malicious traffic, in the shape of heartbleed or heartbleed-like requests, needs a preliminary assessment, to evaluate peculiar features associated with attack data exchange and benign traffic. To this aim, this section describes a preliminary test set-up as well as the parameters measured through the use of a traffic monitoring software, to conclude with the analysis of acquired data, the presentation of the adopted figures of merit and definition of the detection rule.

A. Test Setup

The initial assessment has been carried out by adopting the test setup reported in Fig. 4. It is composed of a couple of personal computers (PCs), connected through an ethernet

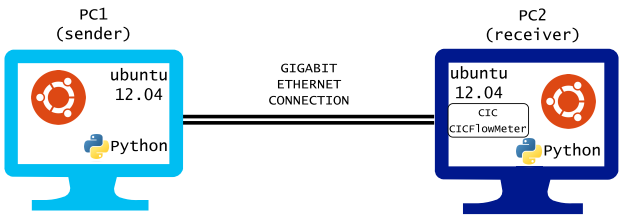


Fig. 4. Block diagram of the preliminary assessment set-up.

cable, according to the protocol IEEE 802.3. Both computers are running a Linux Operating System, namely Ubuntu 12.04. The OS choice is related to the installable OpenSSL version 1.0.1f, as described in Section II, to run heartbleed attacks. In detail, the old version is required on the attack target only: to make the system uniform, we adopted the same software features on both sides of the communication. In this way, symmetry can be observed and no preferential way can bias the obtained results. On the receiver side, an open source traffic monitoring software is also installed, i.e., CIC - CICFlowMeter [31], [32], which is generally able both to generate and monitor traffic by measuring some parameters of interest over the exchanged traffic. For our convenience, traffic was exchanged using Python scripts and CICFlowMeter was adopted only for acquiring and measuring the parameters of interest. In this way, bidirectional traffic between the sender (PC1) and receiver (PC2) is monitored on a flow-by-flow basis and 83 relevant parameters are measured for each measurement session. This set of parameters constitutes the basis for our analysis.

B. Traffic Generation and Measurement Conditions

The preliminary analysis has been carried out by considering several types of traffic, to take into account the most common data exchange conditions and to build up an attack detection rule which is robust and able to generalize its output. To this aim, the network traffic has been generated according to the following categories.

1) Benign Traffic.
   a) Heartbeat.
   b) Interactive.
   c) Non-real time.
   d) Latency sensitive.
2) Malicious Traffic.
   a) Heartbleed.
   b) Heartbleed-like.

In detail, 13 751 flows have been generated by means of suitably developed Python scripts and captured by CICFlowMeter, for a total time of several hours of data exchange per type of traffic. The size of each flow was variable, depending on the adopted protocol and the eventual presence of an attacking behavior. The traffic has been injected into a gigabit ethernet link and exploited the maximum speed the link allowed.

Details about the characteristics of malicious traffic are reported in Table I.

TABLE I
SIZE CHARACTERISTICS OF THE GENERATED MALICIOUS TRAFFIC

Traffic type	Minimum size	Maximum size	Step size
Heartbleed	4096	65536	4096
Heartbleed-like	16384	65536	4096

In such a preliminary phase, the data analysis has been performed offline. The measured parameters have been analyzed and processed subsequently.

C. Data Analysis

The performance of a good IDS is conditioned by the type of data and the method of acquisition. In our case, we chose to exploit the information brought by network traffic. Usually, network data collection can be performed in two ways.

1) Full Packet CAPture (PCAP) allows obtaining detailed information about packets belonging to the network, such as packet headers, packet size, protocol, flags. Moreover, it is also possible to read the packet payload containing private information or sensitive data.
2) NetFlow, unlike the aforementioned method, provides a measurement of some parameters about each flow. Examples of measurements are the number of bytes exchanged during a flow, the number of packets in both directions, and derived parameters (e.g., average, standard deviation, variance).

For our purpose, we chose to use the latter methodology measuring the network flow parameters using the previously described tool (i.e., CICFlowMeter tool).

Once the whole parameter set was acquired, two possible ways could be generally followed for identifying the most suitable parameters which should allow to discriminate between malicious and benign traffic.

1) Use an automatic feature selection method, typically employed in the field of Machine Learning and statistics, which is a commonly used procedure to select a subset of important features, instead of using the entire dataset, where redundant or irrelevant features may occur. Many advantages come from this procedure, some of the most important are the simplification of the model and shorter training times. Scikit-learn offers a wide choice of Feature Selection algorithms, depending on the classifier it is needed to use.
2) Analyze the physical meaning of the parameters returned by the measurement software and then consider a subset of them to propose a rule-based methodology. In the case of heartbleed and heartbleed-like attacks, both exhibit a behavior that privilege a larger amount of data in the download direction than in the upload one. Accordingly, we chose all measured parameters that could highlight the unbalance between traffic flow directions.

A summary of the obtained selections is reported in Table II. In particular, the Feature Selection approach is here shown in the case of the Random Forest Regressor (RFR), which is a meta estimator among a number of classifying decision trees (DTs) on multiple subsets of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

TABLE II
COMPARISON BETWEEN DATA-DRIVEN AND KNOWLEDGE-BASED TECHNIQUES

Methodology	Selected Features
RF Regressor	down_up_ratio, bwd_pkt_len_mean, init_bwd_win_byts bwd_blk_rate_avg, bwd_pkt_len_std, bwd_byts_b_avg, fwd_blk_rate_avg, bwd_pkts_b_avg, bwd_iat_max, bwd_iat_mean, bwd_seg_size_avg, flow_iat_max, totlen_bwd_pkts, fwd_iat_max, fwd_seg_size_avg, flow_iat_min, fwd_pkt_len_mean, subflow_bwd_pkts, bwd_iat_tot, subflow_bwd_byts, fwd_pkt_len_std, fwd_byts_b_avg, bwd_iat_std, fwd_act_data_pkts, fwd_pkt_len_max, fwd_pkts_b_avg, flow_iat_mean, tot_bwd_pkts, subflow_fwd_ pkts, bwd_iat_min, fwd_pkts_s, flow_pkts_s, flow_iat_std, fwd_iat_min, subflow_fwd_byts
Rule-based approach	down_up_ratio, bwd_pkt_len_max, bwd_pkt_len_mean, bwd_pkt_len_std, fwd_pkt_len_max, fwd_pkt_len_mean, fwd_pkt_len_std, init_bwd_win_byts, bwd_byts_b_avg

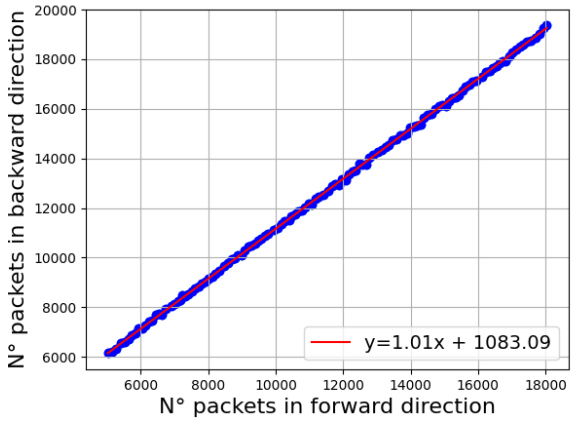
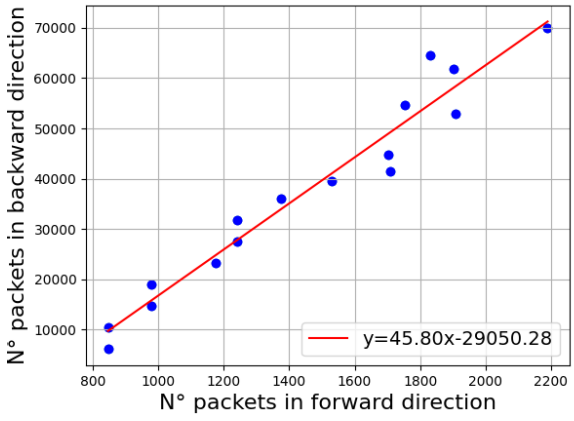


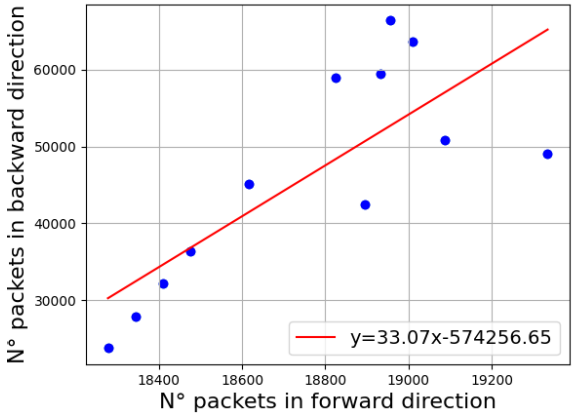
Fig. 5. Trend about the number of packets exchanged in forward and backward communication in heartbeat protocol.

It is possible to highlight that the automatic feature selection procedure identifies a higher number of parameters than the rule-based one. Therefore, it is expected that with ML methods, the corresponding models will be more complex than in the case of the rule-based one.

Our approach leveraged on the assumption that during the flows exchanged using the protocol, the number of heartbeat packets in forward and backward direction are the same, having a symmetric behavior. For proving that, we sent heartbeat packets whose lengths varied from the minimum allowed which is 4096 bytes to the maximum which is 16 384 bytes. In Fig. 5, the scatter plot showing the relationship between the exchanged packets is reported. The equation of linear regression describes how variable y changes as variable x changes (the slope is very close to 1). Indeed, (e.g.,) a server implements an echo mechanism with a client. On the contrary, during a buffer overread attack, the amount of packets received in backward is far greater than the packets sending in forward direction, showing an asymmetric behavior. Also in this case, to obtain it, according to I we manipulated the value of length of heartbeat payload, keeping empty payload for heartbleed attack, while to obtain heartbleed-like attack we sent a packet which having the same length than heartbeat, but varying the value in payload length from 16 384 to 65 536 bytes, keeping step of 4096 bytes. The trend of forward (x) and backward (y) packets are shown into Fig. 6, highlighting the slope greater than 45 for heartbleed and greater than 33 for heartbleed-like attack, thus confirming the expected asymmetric behavior.



(a)



(b)

Fig. 6. Trend about the number of packets exchanged in forward and backward communication in heartbleed and heartbleed-like attack. (a) Heartbleed attack. (b) Heartbleed-like attack.

D. Adopted Figures of Merit

To evaluate the capability of our methodology to discriminate between attack and benign conditions, and to compare the achieved performance with one provided by competitive solutions, some classical figures of merit, used in classification problems, have been employed. Particularly, considering the binary hypothesis testing and assuming H_0, the null hypothesis, as the attack presence and H_1, the alternative hypothesis as the attack absence, it is possible to define.

1) True Positive (TP) $= \sum_{i=1}^{N_t} \mathbb{1}_{(H_0|H_0)}$.
2) True Negative (TN) $= \sum_{i=1}^{N_t} \mathbb{1}_{(H_1|H_1)}$.
3) False Positive (FP) $= \sum_{i=1}^{N_t} \mathbb{1}_{(H_0|H_1)}$.
4) False Negative (FN) $= \sum_{i=1}^{N_t} \mathbb{1}_{(H_1|H_0)}$

where the $\mathbb{1}$ is the indicator function whose output is 1 if the expressed condition is true, or 0 otherwise. Furthermore, N_t stands for the total number of tests (one test corresponds to one flow, in our case).

Starting from TP, TN, FP, FN, the following quantities are considered:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4}$$

$$F1\text{-score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{5}$$

$$\text{AUC} = \frac{1}{2} - \frac{\text{FP}}{2(\text{TN} + \text{FP})} - \frac{\text{TP}}{2(\text{TP} + \text{FN})}. \tag{6}$$

E. Proposed Rule-Based Methodology

Considering selected measured parameters reported in Table II concerning the rule-based methodology, we proposed a flow-by-flow evaluation of the parameters of interest for developing the rules described in the following. In other words, considering the parameters measured by CICFlowMeter at each flow, they are used for evaluating the involved quantities. To these aims, we started the analysis evaluating the relationship (RL_1), which is a slight modification of the one tested in [22], where only heartbeat and typical heartbleed attacks were considered.

The relationship RL_1 is reported in the following equation:

$$\text{RL}_1 = \frac{(\text{bwd\_len\_max} - \text{bwd\_len\_mean}) \cdot \text{fwd\_len\_std}}{\text{bwd\_len\_std} \cdot (\text{fwd\_len\_max} - \text{fwd\_len\_mean})}. \tag{7}$$

The adopted parameters to describe RL_1 are.

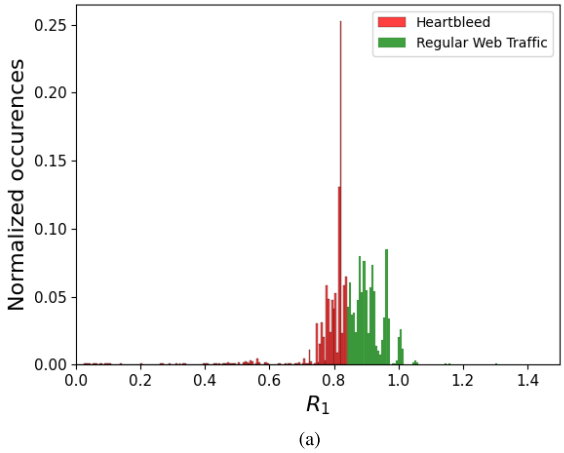
1) *bwd_len_max:* maximum value of the packet lengths (in bytes) in the backward direction (download case, considering PC1) - indicated as bwd_pkt_len_max in CICFlowMeter.
2) *bwd_len_mean:* average value of the packet lengths (in bytes) in the backward direction (download case, considering PC1) - indicated as bwd_pkt_len_mean in CICFlowMeter.
3) *bwd_len_std:* standard deviation value of the packet lengths (in bytes) in the backward direction (download case, considering PC1) - indicated as bwd_pkt_len_std in CICFlowMeter.
4) *fwd_len_max:* maximum value of the packet lengths (in bytes) in the forward direction (upload case, considering PC1) - indicated as fwd_pkt_len_max in CICFlowMeter.
5) *fwd_len_mean:* average value of the packet lengths (in bytes) in the forward direction (upload case, considering PC1) - indicated as fwd_pkt_len_mean in CICFlowMeter.
6) *fwd_len_std:* standard deviation value of the packet lengths (in bytes) in the forward direction (upload case, considering PC1) - indicated as fwd_pkt_len_std in CICFlowMeter.

In Fig. 7, the values obtained for RL_1 in the preliminary test campaign are reported.

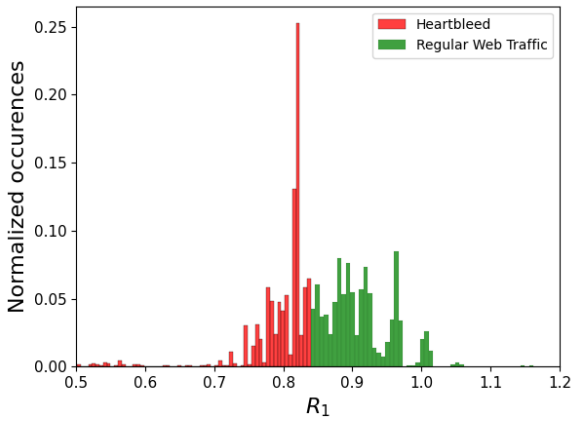
A second proposed relationship, namely RL_2, is simply defined as

$$\text{RL}_2 = \text{down\_up\_ratio} \tag{8}$$

where the adopted quantity, down_up_ratio stands for the ratio between the number of packets in the download direction (from PC2 to PC1) and the same quantity in the upload direction (from PC1 to PC2).

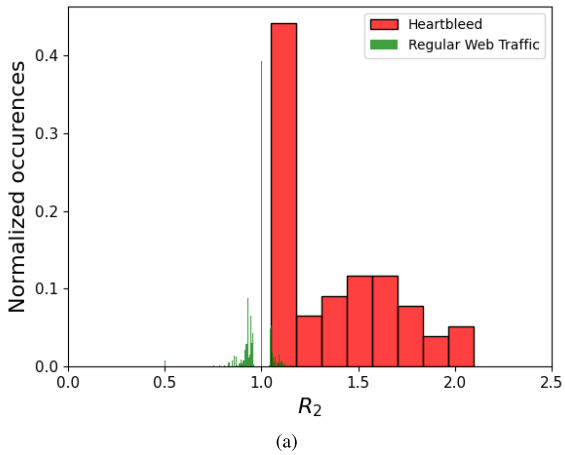


(a)

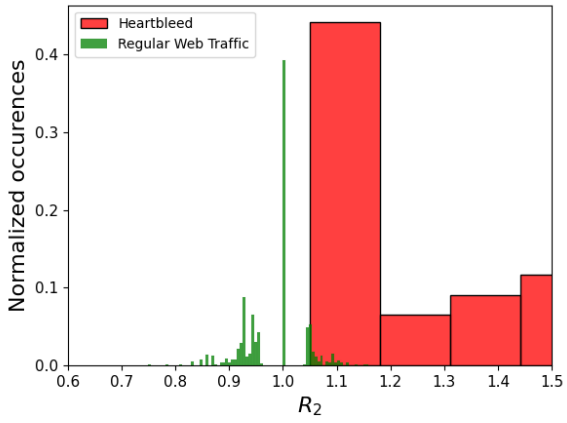


(b)

Fig. 7. Normalized histogram of RL_1 values in benign and malicious traffic conditions. (a) Normalized histogram of RL_1 values. (b) Normalized histogram of RL_1 values zoom-in.



(a)



(b)

Fig. 8. Normalized histogram of RL_2 values in benign and malicious traffic conditions. (a) Normalized histogram of RL_2 values. (b) Normalized histogram of RL_2 values zoom-in.

In Fig. 8, the values obtained for RL_2 in the preliminary test campaign are reported.

The third and last relationship RL_3 is based on the following equation:

$$RL_3 = \frac{\text{bwd\_len\_mean} - \text{fwd\_len\_mean}}{\text{fwd\_len\_mean}} \tag{9}$$

where the involved quantities have the same meaning explained referring to rule RL_1.

In Fig. 9, the values obtained for RL_3 in the preliminary test campaign are reported.

Starting from the analysis of the histograms of RL_1, RL_2, and RL_3, the relationships have been chosen according to the *asymmetry* concept as well as their good separability property when benign traffic and malicious cases are considered.

Consequently, the following rules are identified:

$$R_1 = \{RL_1 \in [0.00, 0.84]\} \tag{10}$$

$$R_2 = \{RL_2 \in [1.05, 2.10]\} \tag{11}$$

$$R_3 = \{RL_3 \in ([0.16, 1.32]||[6.80, 18.00])\}. \tag{12}$$

The "true" value of each rule identifies possible heartbleed attacks. As shown in Figs. 7–9, the separability property is not perfectly respected for all considered relationships: therefore, a too high number of false positives could arise from the single application of each rule. Consequently, we chose to apply a logical AND relation among all the truth values of the proposed rules for definitely detecting either heartbleed or heartbleed-like attacks.

IV. Example of On-Field Application

In this section, a real system composed of IoT nodes and a central IoT gateway is realized and the methodology is evaluated on-field. Particularly, the adopted implementation, the experimental results and their comparison against traditional ML techniques are discussed.

A. Experimental Setup

To make the proposed measurement method really applicable to the context of IoT frameworks, here we propose a typical network configuration of IoT applications, whose block diagram is proposed in Fig. 10.

In the proposal, we supposed to have some IoT nodes which have limited networking capabilities and, to reach the Internet, they need to pass through an IoT gateway, where the protection method against heartbleed and heartbleed-like attacks is implemented. The malicious user which would like to steal data from nodes is depicted as "Attacker." The WAN

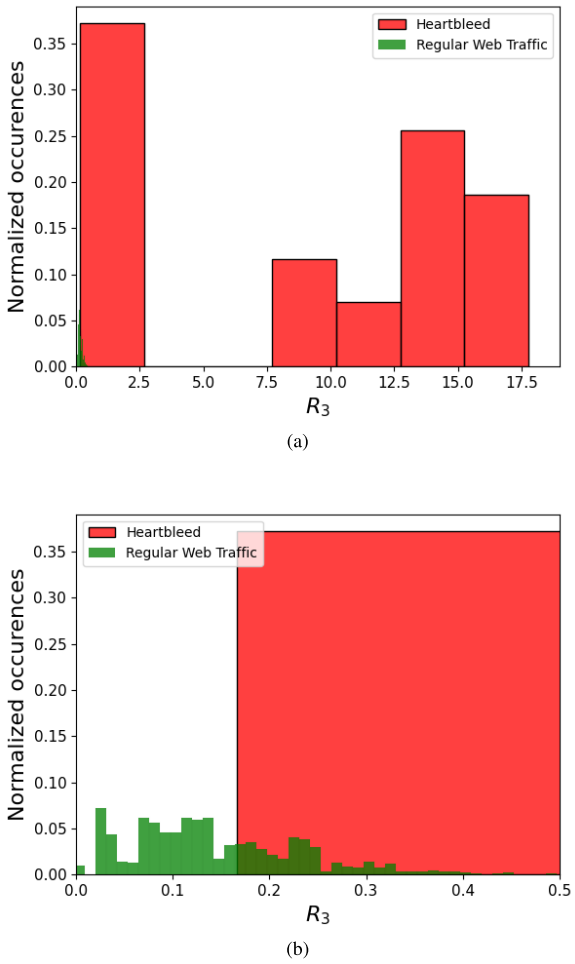


Fig. 9. Normalized histogram of RL_3 values in benign and malicious traffic conditions. (a) Normalized histogram of RL_3 values. (b) Normalized histogram of RL_3 values zoom-in.

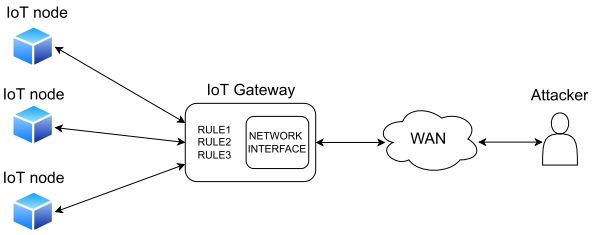


Fig. 10. Network configuration where an IoT gateway implements protection rules acting as a firewall for the IoT nodes.

network cloud only recalls the possibility that the malicious user can belong to an external network, able to reach the IoT node through the IoT gateway. The choice of the edge device is not discriminant in our approach, IoT nodes or old machines can act similarly whenever they only have to manage heartbeat or simple traffic requests. The whole task is accomplished by the center-sided IoT gateway, which is aimed to protect any low-end device, whose traffic is preliminarily filtered and, consequently, verified. Furthermore, the IoT gateway does not care about the device typologies it has to manage and it does not deal with any patch fixing. The proposed approach is widely applicable to protect heartbleed but also other attacks dealing with databreach. It is worth noting that the attack is not directly blocked in the requesting phase, but

TABLE III
LIST OF THE NUMBER OF FLOW TYPOLOGIES EXCHANGED DURING THE TEST

Type	Number of flows
Heartbeat	12363 (exact)
Heartbleed	16 (typologies)
Heartbleed-like	12 (typologies)
Regular traffic	548 (exact)

double-checked with the pair request/response. The attack is not finally accomplished since the low-end device response never reaches the attacker's end, but it is blocked by the gateway itself.

As for the Attacker and the IoT node (attacked), two light terminals running Ubuntu 12.04 LTS operating system, (vulnerable to heartbleed-like attacks because of the unpatched OpenSSL version) have been considered.

As for the IoT gateway, to prove the implementation of the proposed method on low-cost platforms, a Raspberry Pi 4 Model B, equipped with Ubuntu 22.04 LTS (the OpenSSL version is not relevant in this case, since it only needs to implement the rule and running CICFlowMeter) has been adopted. It is equipped with 3.10 Python version and, flow-by-flow, it runs the described measurement method for the heartbleed and heartbleed-like inline detection.

To compare the proposed method with alternative solutions, the same hardware has been adopted for implementing very popular classifiers, as better described in the following.

B. Experimental Results

The experimental test is composed of four acquisitions. The first one involved the exchange of network flows using heartbeat protocol where each heartbeat request has been increasing its size by 1 byte per flow starting from 4021 to 16 348 bytes. In the second one, the canonical heartbleed attack is launched. Meanwhile, in the third one, the heartbleed-like attack had the canonical size of the heartbeat packet, but the length specified in the length field packet has been increased from 16 384 to 65 536 bytes with 4096 bytes per step. In the end, the last one involved the regular traffic as described in Section III. In Table III are showed the exact number of flows exchanged during the test. As regular traffic and heartbeat regard, the reported number is the total number of flows; as heartbleed and heartbleed-like are concerned, the number is related to flow typologies, each of them has been repeated 30 times to have a statistical meaning of the test.

According to the key performance indicators introduced in the previous section [see (1)–(6)], Table IV reports the achieved results, in which the progression of performance with the addition of AND condition is reported. In particular, a general improvement is obtained by adding the first AND condition, while slight modifications (in the increasing direction) are observed passing from the AND between two rules to the same condition among the three ones.

It is relevant to highlight that if we use only the RL_1 rule, we obtain a high value of sensitivity but a low value of precision (recall), leading to a low value of F1-score. The latter one, despite a little decrease in sensitivity, increases

TABLE IV
PERFORMANCE TREND BY THE APPLICATION OF THE PROPOSED RULE IN INCREASING AND LOGIC

Rule	Accuracy	Precision	Sensitivity	Specificity	F1-score	AUC
R_1	0.874	0.798	1.000	0.747	0.888	0.874
R_1 & R_2	0.980	0.996	0.964	0.996	0.978	0.980
R_1 & R_2 & R_3	0.981	0.998	0.964	0.999	0.981	0.982

TABLE V
PERFORMANCE COMPARISON ON RASPBERRY PI, USING RF FEATURE SELECTION AS INPUT FOR CLASSIFIERS

Method	Accuracy	Precision	Sensitivity	Specificity	F1-score	AUC
DT	1.000	1.000	1.000	1.000	1.000	1.000
GNB	0.950	1.000	0.890	1.000	0.940	0.990
K-NN	0.900	0.890	0.890	0.910	0.890	0.970
MLP	0.900	0.890	0.890	0.910	0.890	0.970
RF	1.000	1.000	1.000	1.000	1.000	1.000
RBA	**0.982**	**0.999**	**0.964**	**0.999**	**0.981**	**0.982**

evaluating also RL_2 and RL_3, justifying an improvement of the AUC score. Actually, the RL_1 and RL_2 conditions are strictly necessary. The third rule could, in the specific case, be omitted but it is the only rule taking into account the difference between the mean values in both directions. There could be cases where the total number of packets is similar while their mean length changes. In heartbleed detection, rule 3 could also be neglected but we believe that in heartbleed-like attacks it is good to keep it, since its implementation is straightforward due to the fact that needed quantities are already taken to implement RL_3 and therefore the computational additional burden is really negligible.

C. Comparison With Traditional ML Techniques

Nowadays AI is exploited in several fields, including cyber security. Typically, Machine Learning and Deep Learning solutions are proposed in the literature for developing IDSs, thus representing a reasonable benchmark for comparing the proposed method with widespread and competitive ones. Considering that the development of Deep Learning techniques generally requires more expensive computational unit, in the following we consider only Machine Learning algorithms.

In particular, among the most known ML Algorithms, we selected those widely employed in such a field of application: DT, Gaussian Naive Bayes (GNB), *K*–Nearest Neighbor (K–NN), MultiLayer Perceptron (MLP), Random Forest (RF).

Performance comparison among the above-mentioned ML algorithms and the proposed rule-based approach (RBA) is reported in Tables V and VI. It has been carried out using RF feature selection as input for classifiers. In particular, in Table V the same figures of merit, previously defined in (1)–(6), have been considered. Instead, Table VI reports a further analysis that has been carried out with the aim of comparing the memory usage and the execution time among the proposed method against the considered ML-based classifiers.

Analyzing these results, it is possible to note that the ML algorithms that have shown the better performance are RF and DT, which have obtained the maximum score for all the considered key performance indicators, even if the former has shown an execution time that is almost double of the latter.

TABLE VI
COMPARISON AMONG RULE AND CLASSIFIERS ON BOTH CPU TIME AND MEMORY USAGE, USING RF FEATURE SELECTION

Method	Time [s]	Memory Occupancy [kB]
DT	0.309	51.280
GNB	0.100	51.280
K-NN	0.300	61.280
MLP	0.133	173.280
RF	0.538	119.280
Proposed Rule	**0.040**	**13.691**

As for the proposed RBA, it has shown a very good performance, comparable with DT and RF, in terms of precision and specificity and a little bit worse, but better than the other considered ML algorithms, in terms of accuracy, sensitivity, and F1-score. But the main interesting achievement concerns the execution time and memory usage. In fact, it has shown an execution time that is more than 5 times shorter than that experienced when the DT algorithm has been considered. A similar remark can be made concerning memory usage. In fact, the proposed rule has a memory usage that is 5 times lower than DT, which is the classifier that has shown the lowest memory usage.

Summarizing, the achieved results confirm the very good performance of the proposed RBA concerning popular ML techniques, and a significant improvement in terms of execution times and memory requirements. These last considerations confirm the suitability of the proposed approach to be implemented also on low-performance systems and for inline detection.

These features become particularly attractive whenever an IDS able to detect several kinds of attacks has to be developed.

V. TOWARD A WIDE APPLICABILITY

Proposed results and methodology allow discussing about the possible extension of the work in more general and complex frameworks of cyber security, by analyzing both the level of generalization and the likelihood of the considered kind of attacks.

As for the level of generalization, two raising issues have to be considered: 1) the capability to keep low-complexity methods to detect further cyberattacks and 2) the management of such a kind of attacks in case of larger networks.

The response to 1) is related to the attack typology. The proposed attack detection system and its low complexity can be kept whenever the attack deals with anomalies in data flow rates or traffic unbalancing. When the attack moves to other typologies, such as ransomware or other ones working on target data more than on flow alteration, a different methodology has to be investigated, hopefully keeping the computational complexity low. Regarding 2), in the case of networks characterized by a high number of nodes, the proposed solution is still applicable by splitting the detection points. In other words, each IoT gateway could implement the proposed intrusion detection solution for managing its subset nodes.

As for the likelihood of the considered attacks in today's scenario, actually, the computation and technology are moving toward a dichotomous mechanism, where powerful and centralized computing centers coordinate pervasive very low-power devices, which are placed everywhere. These last ones capture data from nature and society about almost everything and, periodically, send their acquisition to central servers. The possibility of having sensitive data in temporary buffers is not that rare, as well as the possibility of their system updates being carried out only in nonservice mode. Therefore, the authors see a wide impact of the faced topic in the IoT world, especially considering the lightweight found solution, eligible for implementation in low-power devices (even small gateways coordinating peripheral nodes).

VI. Conclusion

In the framework of IoT applications, cyber-attacks represent a significant threat to developers of systems and applications. Indeed, the typical application constraints of low-cost, low-powerful architecture, and, at the same time, the required networking operativity make such systems intrinsically vulnerable to cyber criminals. Moreover, the use of low-powerful platforms for developing IoT sensor nodes or IoT gateways, often makes it impossible to update the systems with suitable patches for correcting the frequently identified vulnerabilities.

Therefore, among data leakage attacks, heartbleed and heartbleed-like still represent significant threats to IoT applications. In this context, the article has presented a novel and rule-based measurement method that can be developed on low-cost platforms for inline detecting the presence of such kinds of cyberattacks. The low complexity of the proposed solution makes it directly implementable on IoT gateways and allows well-discriminating malicious traffic from the benign one (that could involve an IoT sensor node belonging to the IoT gateway segment).

In particular, the implementation of the proposed method on a very popular platform (i.e., Raspberry Pi 4), has proved that, compared with solutions based on popular ML-based techniques, the proposed solution keeps similar performance (in several cases also better) in terms of the typical figures of merit adopted in the context of IDSs, but, due to the lowest complexity, it requires very lower execution times and memory requirements. These features become particularly attractive, looking at the long-term goal of the research activity of developing an IDS able to detect several kinds of attacks.

Indeed, future developments will extend the proposed method to further kinds of cyberattacks typically affecting IoT applications, like DDoS and Bad Data Injection.

References

[1] D. A. Kumar and S. Venugopalan, "Intrusion detection systems: A review," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 8, pp. 1–15, 2017.

[2] K. Yu, K. Nguyen, and Y. Park, "Flexible and robust real-time intrusion detection systems to network dynamics," *IEEE Access*, vol. 10, pp. 98959–98969, 2022.

[3] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments," *Energy Rep.*, vol. 7, pp. 8176–8186, Nov. 2021.

[4] H. Sarjan, A. Ameli, and M. Ghafouri, "Cyber-security of industrial Internet of Things in electric power systems," *IEEE Access*, vol. 10, pp. 92390–92409, 2022.

[5] A. Chidukwani, S. Zander, and P. Koutsakis, "A survey on the cyber security of small-to-medium businesses: Challenges, research focus and recommendations," *IEEE Access*, vol. 10, pp. 85701–85719, 2022.

[6] A. Huseinovic, S. Mrdovic, K. Bicakci, and S. Uludag, "A survey of denial-of-service attacks and solutions in the smart grid," *IEEE Access*, vol. 8, pp. 177447–177470, 2020.

[7] G. Bernieri, M. Conti, and F. Pascucci, "A novel architecture for cyber-physical security in industrial control networks," in *Proc. IEEE 4th Int. Forum Res. Technol. Soc. Ind. (RTSI)*, Sep. 2018, pp. 1–6.

[8] N. Weinberg. *7 Hot Cybersecurity Trends (and 2 Going Cold)*. Accessed: Oct. 24, 2022. [Online]. Available: https://www.csoonline.com/article/3262972/7-hot-cybersecurity-trends-and-2-going-cold.html

[9] Z. Durumeric et al., "The matter of heartbleed," in *Proc. Conf. Internet Meas. Conf.*, Nov. 2014, pp. 475–488.

[10] M. Carvalho, J. DeMott, R. Ford, and D. A. Wheeler, "Heartbleed 101," *IEEE Secur. Privacy*, vol. 12, no. 4, pp. 63–67, Jul. 2014.

[11] T. A. Nidecki. *The Heartbleed Bug—Old Bugs Die Hard*. Accessed: Oct. 24, 2022. [Online]. Available: https://www.acunetix.com/blog/web-security-zone/heartbleed-bug/

[12] D. E. Geer and P. Kamp, "Inviting more heartbleed," *IEEE Secur. Privacy*, vol. 12, no. 4, pp. 46–50, Jul. 2014.

[13] Z. Hu, P. Chen, M. Zhu, and P. Liu, "A co-design adaptive defense scheme with bounded security damages against heartbleed-like attacks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4691–4704, 2021.

[14] Y. Wang, H. Wang, X. Hei, W. Ji, and L. Zhu, "Petri net modeling and vulnerability analysis of the heartbleed," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Oct. 2021, pp. 155–160.

[15] J. Sigholm and E. Larsson, "Cyber vulnerability implantation revisited," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2021, pp. 464–469.

[16] M. Kherbache, K. Amroun, and D. Espes, "A new wrapper feature selection model for anomaly-based intrusion detection systems," *Int. J. Secur. Netw.*, vol. 17, no. 2, pp. 107–123, 2022.

[17] R. Panigrahi et al., "Intrusion detection in cyber–physical environment using hybrid Naïve Bayes–decision table and multi-objective evolutionary feature selection," *Comput. Commun.*, vol. 188, pp. 133–144, Apr. 2022.

[18] M. Madou, "Now is the time to strengthen cyber defences," *Netw. Secur.*, vol. 2022, no. 8, Aug. 2022.

[19] S. H. Fern, A. Amir, and S. N. Azemi, "Multi-class imbalanced classification problems in network attack detections," in *Proc. 6th Int. Conf. Electr., Control Comput. Eng.* Cham, Switzerland: Springer, 2022, pp. 1057–1069.

[20] M. S. Milosevic and V. M. Ciric, "Extreme minority class detection in imbalanced data for network intrusion," *Comput. Secur.*, vol. 123, Dec. 2022, Art. no. 102940.

[21] D. Krishnan, "Detection of denial-of-service attacks using stacked LSTM networks," in *Proc. Data Analytics Manage.* Singapore: Springer, 2022, pp. 229–239.

[22] A. Amodei, D. Capriglione, L. Ferrigno, G. Miele, G. Tomasso, and G. Cerro, "A rule-based approach for detecting heartbleed cyber attacks," in *Proc. IEEE Int. Symp. Meas. Netw. (MN)*, Jul. 2022, pp. 1–6.

[23] L. Angrisani, D. Capriglione, L. Ferrigno, and G. Miele, "A methodological approach for estimating protocol analyzer instrumental measurement uncertainty in packet jitter evaluation," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 5, pp. 1405–1416, May 2012.

[24] L. Angrisani, D. Capriglione, L. Ferrigno, and G. Miele, "An internet protocol packet delay variation estimator for reliable quality assessment of video-streaming services," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 5, pp. 914–923, May 2013.

[25] L. Angrisani, D. Capriglione, L. Ferrigno, and G. Miele, "Internet protocol packet delay variation measurements in communication networks: How to evaluate measurement uncertainty?" *Measurement*, vol. 46, no. 7, pp. 2099–2109, Aug. 2013.

[26] L. Angrisani, E. Atteo, D. Capriglione, L. Ferrigno, and G. Miele, "An efficient experimental approach for the uncertainty estimation of QoS parameters in communication networks," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, May 2010, pp. 1186–1191.

[27] D. Capriglione, G. Cerro, L. Ferrigno, and G. Miele, "How to quantify $trust$ in your network emulator?" in *Proc. Int. Conf. Wired/Wireless Internet Commun.* Cham, Switzerland: Springer, 2018, pp. 171–182.

[28] D. Capriglione, G. Cerro, L. Ferrigno, and G. Miele, "The effect of hardware/software features on the performance of an open–source network emulator," in *Proc. Int. Conf. Wired/Wireless Internet Commun.* Cham, Switzerland: Springer, 2019, pp. 233–245.

[29] L. Angrisani, D. Capriglione, G. Cerro, L. Ferrigno, and G. Miele, "Experimental analysis of software network emulators in packet delay emulation," in *Proc. IEEE Int. Workshop Meas. Netw. (MN)*, Sep. 2017, pp. 1–6.

[30] L. Angrisani, D. Capriglione, L. Ferrigno, and G. Miele, "Measurement of the IP packet delay variation for a reliable estimation of the mean opinion score in VoIP services," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2016, pp. 1–6.

[31] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy*, 2017, pp. 253–262.

[32] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Privacy*, 2016, pp. 407–414.

Andrea Amodei (Graduate Student Member, IEEE) received the master's degree in computer science engineering from the University of Cassino and Southern Lazio, Cassino, Italy, in 2020, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Information Engineering.

His activity research focuses on methods for detecting cyber-attacks, leveraging both network traffic and side channel information as the electromagnetic field and current signal.

Domenico Capriglione (Senior Member, IEEE) is currently an Associate Professor of electrical and electronic measurements with the University of Cassino and Southern Lazio, Cassino, Italy. His current research interests include measurements on RF and telecommunication systems, measurements for cyber security, DSP-based measurement systems, network measurements, and measurement of electromagnetic compatibility.

Dr. Capriglione has been served as the Chair for the IEEE I&M Technical Committee TC-37-Measurements and Networking, since 2016.

Gianni Cerro (Member, IEEE) is currently a Research Fellow with the Department of Medicine and Health Sciences "Vincenzo Tiberio" University of Molise, Campobasso, Italy. His research interests include magnetic localization systems for biomedical and industrial applications, cognitive radio systems for new generation communication technologies, measurements in telecommunication networks, sensor networks for environmental monitoring, and measurement characterization of medical devices, such as brain–computer interfaces.

Luigi Ferrigno (Senior Member, IEEE) has been the Scientific Manager of the Industrial Measurements Laboratory, University of Cassino, Cassino, Italy, since 2004, and has been a Full Professor of electric and electronic measurement, since 2020. In 2008, he was a Founding Member of the University spin-off, Spring Off (University of Salerno), Fisciano, Italy. He is an NDE4.0 Ambassador for the Italian Association of Non-Destructive Evaluation and Test (AiPnD) in the EFNDT WG10. He has coordinated and participated in several national and international research projects. His current research interests include the NDT4.0, novel learning sensors and measurement systems for smart city, the Internet-of-Things (IoT), automotive, smart energy, and environment.

Gianfranco Miele (Senior Member, IEEE) is currently an Associate Professor with the Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, Italy. His current research interests include design and implementation of innovative methods for performance assessment of RF telecommunication systems and communication networks.

Giuseppe Tomasso (Member, IEEE) received the master's degree in electrical engineering and the Ph.D. degree in industrial engineering from the University of Cassino, Italy, in 1994 and 1999, respectively.

He is currently a Full Professor of power electronics and electric and hybrid vehicles with the University of Cassino and Southern Lazio, Cassino, Italy. Since 2009, he has been the Chief of the Industrial Automation Laboratory with the University of Cassino and Southern Lazio. He is the Founder of four start-up's and President of E-Lectra company, developing advanced technologies in the field of electric vehicles and energy storage systems. Since 2009, he has been also the Chairperson of the European Ph.D. School: Power Electronics, Electrical Machines, Energy Control and Power Systems. He is also promoter of a racing initiative related to electric go-kart for motorsport. He is coauthor of more than 130 publications in conference proceedings and international transactions. His main research interests include high performances power converters, advanced modulation techniques, industrial automation and electrical drives, and electric and hybrid vehicles powertrain.

Open Access funding provided by 'Università degli Studi di Cassino e del Lazio Meridionale' within the CRUI CARE Agreement