IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 6, 2021, accepted December 2, 2021, date of publication December 13, 2021, date of current version December 23, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3135195*

# Classification of Abnormal Signaling SIP Dialogs Through Deep Learning

**DIOGO PEREIRA[1,2], RODOLFO OLIVEIRA[1,2], (Senior Member, IEEE), AND HYONG S. KIM[3], (Senior Member, IEEE)**

[1]Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Ciências e Tecnologia (FCT), Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[2]Instituto de Telecomunicações, 1049-001 Lisbon, Portugal
[3]Department of Electrical & Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Corresponding author: Diogo Pereira (dfca.pereira@campus.fct.unl.pt)

This work was supported in part by the European Regional Development Fund (FEDER) through the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020 and Programa Operacional Regional LISBOA (LISBOA 2020); and in part by the National Funds through Fundação para a Ciência e Tecnologia under the Project InfoCent-IoT under Grant POCI-01-0145-FEDER-030433, Grant UIDB/50008/2020, and Grant PRT/BD/152200/2021.

**ABSTRACT** Due to the high utilization of the Session Initiation Protocol (SIP) in the signaling of cellular networks and voice over IP multimedia systems, the avoidance of security vulnerabilities in SIP systems is a major aspect to assure that the operators can reach satisfactory readiness levels of service. This work is focused on the detection and prediction of abnormal signaling SIP dialogs as they evolve. Abnormal dialogs include two classes: the ones observed so far and thus labeled as abnormal and already known, but also the unknown ones, i.e., specific sequences of SIP messages never observed before. Taking advantage of recent advances in deep learning, we use Long Short-Term Memory (LSTM) recurrent neural networks (RNNs) to detect and predict dialogs already observed. Additionally, and based on the outputs of the LSTM neural network, we propose two different classifiers capable of identifying unknown SIP dialogs, given the high level of vulnerability they may represent for the SIP operation. The proposed approaches achieve higher SIP dialogs detection scores in a shorter time when compared to a reference probabilistic-based approach. Moreover, the proposed detectors of unknown SIP dialogs achieve a detection probability above 94%, indicating its capability to detect a significant number of unknown SIP dialogs in a short amount of time.

**INDEX TERMS** Session initiation protocol, deep learning, vulnerability prediction, performance analysis.

## I. INTRODUCTION

Currently, the Session Initiation Protocol (SIP) plays a fundamental role as a signaling protocol of IP Multimedia Subsystem (IMS) services [1] and Voice over Internet (VoIP) services [2]. Apart from the classical vulnerabilities mainly associated with authentication schemes, the SIP protocol can also be explored by malicious users to take advantage of the request/response interaction integrated into the sequential behavior of the protocol. The exploration of different signaling patterns can effectively expose vulnerabilities of the multiple SIP servers in the SIP path, which can then be used to perpetrate novel types of attacks [3]–[5] known as SIP signaling attacks. Consequently, there are SIP signaling attacks already known, which can be detected or predicted through the sequence of SIP signaling messages already exchanged between the peers of the established session. Additionally, new types of attacks can be perceived by unseen sequences of SIP signaling sequences, highlighting the importance of detecting SIP sequences never observed before.

Motivated by the importance of predicting or detecting SIP signaling sequences established between SIP peers, in this work we study the performance of state-of-the-art deep learning techniques to detect known and unknown SIP sequences. More specifically, Long Short-Term Memory (LSTM) recurrent neural networks (RNNs) are used to classify a sequence of observed SIP messages into a known SIP dialog. Additionally, the outputs of the LSTM neural network are used to detect if the current observations are part of a known or unknown SIP dialog. When compared to the recent literature on the detection of abnormal SIP dialogs, the innovation of our work is mainly centered on the adoption of the LSTM neural networks and the way they are applied to the SIP specifics, as well as the performance gains reported in this

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Kashif Bashir.

work. As far as we know, our work is the first one characterizing the performance of LSTM deep learning structures for abnormal SIP detection and prediction. The innovative aspects of the paper are listed as follows:

- We propose a deep learning scheme capable of recognizing the SIP signaling patterns of observed SIP sequences already known. The learning structure is formed by either one or two recurrent neural network layers to model the relation of the temporal events, and an output dense layer to identify the SIP signaling pattern;
- The outputs of the recurrent neural network are used as a set of features to identify the signaling patterns incorrectly predicted. Two classifiers are proposed to distinguish between trained dialogs and unknown dialogs either through the skewness and kurtosis central moments or the maximum value of the neural network outputs;
- The proposed solutions to predict or detect the SIP dialogs are evaluated through different experiments and metrics to identify the appropriate neural network structure and its hyperparameters. The experiments assess the ability to predict SIP signaling sequences while the messages are received over time and the capacity of detecting SIP dialogs already observed in the past. Additional experiments are taken to identify the most accurate classifier. The experiments assess the probability of correctly identifying a trained and an unknown SIP dialog;
- The results obtained in the deep learning approach are compared with a machine learning approach proposed in [6], which computes the most probable SIP signaling patterns through a $n$-gram Hidden Markov Model (HMM). To the best of our knowledge, the HMM model proposed in [6] is the only work focused on detecting and predicting the SIP dialog identifier from a sequence of SIP messages, while considering the detected identifier to classify the dialog as a possible threat.
- The comparison presented in this paper includes the SIP dialog prediction and detection probabilities, the classification of unknown dialogs, and the computation time of the different approaches (LSTM and HMM). A comparison between the deep and the machine learning approaches evidence that deep learning halves the computation time. In addition, the prediction probability of the deep learning approach has an improvement of 17.04% over the HMM approach, presents a lower computation time, and achieves a detection probability of unknown SIP dialogs above 94%.

Regarding the structure of the paper, we start to introduce related works in Section II. Section III introduces the system model and describes the proposed LSTM models. Section IV presents the experimental results and the performance of the proposed solutions. Finally, Section V concludes the paper.

In what follows, vectors are represented in lower case, upright boldface type, e.g., $\mathbf{v} = \{v_1, v_2, \ldots, v_k\}$. A vector of $k$ consecutive (ordered) elements, also denominated a sequence, is denoted by $\mathbf{v} =< v^{(1)}, v^{(2)}, \ldots, v^{(k)} >$. Sets are represented in calligraphic font, e.g., $\mathcal{S}$.

## II. LITERATURE REVIEW

### A. SIP SECURITY

The security of the SIP Protocol [7] has been analyzed in several papers [5], [8], [9]. It is well known that the different types of attacks can lead to service interruption or destruction and the undue consumption of SIP resources previously allocated for other purposes. A significant amount of works on SIP anomaly detection is formed on malformed SIP messages and their possible consequences in terms of effective SIP attacks. Several techniques were already proposed to cope with malformed and malicious SIP messages, including the use of firewalls capable of detecting intrusion [10], specific learning schemes [11] and comparative approaches based on the statistics of different patterns [12]. Another source of SIP attacks is related to the SIP authentication schemes [13]. Several authentication schemes have been proposed for SIP including multiple-factor authentication methodologies [14]. Flooding attacks also constitute a representative number of service interruptions. Multiple solutions were already proposed to minimize the effects of the flooding attacks, including threshold-based schemes that identify the attacks by comparing the SIP traffic patterns with previous traffic patterns occurring during normal SIP operation [15]. SIP parser vulnerabilities are also a desirable target for attackers. In this case, the SIP messages can be modified to decrease the efficiency of the servers and/or even block the processing and memory resources, and the solutions to mitigate these types of attacks are usually based on prior classification of the receiver SIP messages before being parsed by the servers [16].

The kind of SIP vulnerabilities explored in this paper is related to the SIP signaling logic, where malicious users can explore the diversity of SIP systems to take advantage of defective implementations [5]. The SIP signaling vulnerabilities have been considered in [17], where the authors have proposed a debugger tool to analyze the flow of received SIP messages to be further categorized into groups of compliant dialogs and non-compliant ones. The scheme proposed in [3] to mitigate SIP signaling attacks is based on the contextual information of the SIP traffic, similar to the solution proposed in [4], where the interaction of the SIP peers and their specific timings are compared to prior data to identify significant deviations.

In our work, we are motivated by the latest advances in deep learning tools. Recently, the use of machine learning and deep learning techniques brought a plethora of unprecedented innovations. Learning was adopted in [18] to classify IP traffic based on the statistics of different flows. The work [19] has proposed an unsupervised detection scheme of spam over internet telephony. VoIP systems were also the main target

of the works presented in [20] and [21], where deep learning systems were proposed for the detection of possible steganography in VoIP streams. Deep learning was also used in [22] to detect VoIP traffic in tunneled and anonymous networks, in [23] to identify if voice calls were originated from VoIP systems or cellular/fixed voice networks, and in [24] and [25] to assess the quality of VoIP calls.

When compared to the works in [3], [4], and [17], our work is not considering a fixed probabilistic model of the SIP operation neither fixed rules that describe the SIP interaction. Our goal is to devise an automatic detection and prediction system based on deep learning, that is capable of detecting known and unknown SIP dialogs. While known SIP dialogs are already labeled and their level of vulnerability can be computed based on prior knowledge, the detection of unknown SIP dialogs is of high importance to detect novel attacks.

### B. SIP PROTOCOL

The SIP protocol was proposed for signaling multimedia sessions established by multiple peers. The signaling is implemented through the exchange of SIP messages. To initiate an interaction a peer sends a SIP request message containing the indication of its type through the SIP method field in the SIP message header. The peer receiving the SIP request answers with a SIP response message that includes a reply code in the message header. A SIP request exchanged between SIP peers forms a SIP transaction that includes the SIP request and any responses to it. A SIP dialog is formed by multiple SIP transactions and represents the sequence of SIP signaling operations exchanged between the SIP peers over time. Each SIP dialog is unequivocally identified through the SIP Call ID field in the message header. In this work, we assume that the peers and the SIP servers forming the path between the peers have access to the SIP messages exchanged by the peers and can read the headers of the SIP messages to identify the Call ID and the type of the SIP requests and responses capable of characterizing a specific dialog.

## III. DEEP LEARNING MODEL FOR SIP SIGNALLING PATTERNS CLASSIFICATION

This section describes in Subsection III-A the recurrent neural network models adopted to predict and detect SIP dialogs. The detection of unknown dialogs is based on statistical classification models and it is described in Subsection III-B.

In the proposed approach we consider that the exchanged SIP messages, denoted by $m_k$, are captured by a SIP server or SIP peer that runs the detector/estimator scheme, creating an observed sequence of SIP messages over time, denoted by $\mathbf{n}_k \in \mathcal{X}$, that is used as the input of the learning model. Using the Call ID information contained in the header of each SIP packet to identify the SIP dialog, the model uses the sequences in the observed sequence $\mathbf{n}_k$ to predict or detect the most probable SIP dialog identifier $\mathbf{y}_k \in \mathcal{Y}$, where $\mathcal{Y}$ denotes the output state space of the predictable SIP dialogs. The output $\mathbf{y}_k$ is compared with statistical information previously collected to validate the proposed model. A table of symbols is given in Table 1 to introduce the notation adopted in this section.

**TABLE 1. Table of symbols.**

| Symbols | Definitions |
|---|---|
| $m_k$ | SIP message $k$. |
| $\mathbf{m'}_k$ | Encoded SIP message $k$. |
| $\mathcal{M}$ | Set of all SIP messages. |
| $M$ | Number of all SIP methods and responses. |
| $\mathbf{d}_k$ | SIP dialog $k$. |
| $\mathbf{o}_k$ | Observation $k$. |
| $\mathbf{n}_k$ | Padded sequence of an observation $\mathbf{o}_k$. |
| $L_d$ | Length of a SIP dialog $\mathbf{d}_k$. |
| $L_o$ | Length of an observation $\mathbf{o}_k$. |
| $L_M$ | Length of the encoded SIP message $\mathbf{m'}_k$. |
| $n$ | Number of zeros added into the padded sequence. |
| $N$ | Number of unique SIP dialogs. |
| $\mathbf{y}_k$ | Identifier of dialog $k$. |
| $\mathcal{X}$ | Input state space. |
| $\mathcal{Y}$ | Output state space. |
| $Skew(.)$ | Skewness function. |
| $Kurt(.)$ | Kurtosis function. |
| $\lambda_M$ | Mean threshold (maximum output classifier). |
| $\lambda_S$ | Skewness threshold (skewness and kurtosis classifier). |
| $\lambda_K$ | Kurtosis threshold (skewness and kurtosis classifier). |
| $H_0$ | Hypothesis 0 (classifier detects a trained dialog). |
| $H_1$ | Hypothesis 1 (classifier detects an unknown dialog). |
| $\mu_S$ | Mean of the skewness of the trained dialogs. |
| $\mu_K$ | Mean of the kurtosis of the trained dialogs. |
| $\sigma_S^2$ | Variance of the skewness of the trained dialogs. |
| $\sigma_K^2$ | Variance of the kurtosis of the trained dialogs. |

### A. LSTM RNN MODELS

Two LSTM RNN models were identified in an iterative fashion to predict and detect the most likely SIP Dialog identifier. The first LSTM RNN model, illustrated in Figure 1(a), comprises one LSTM layer and a Dense layer. The LSTM model was chosen due to its ability to process temporal sequences. LSTM models are recurrent models, meaning that whenever a new element of the input sequence is processed the model always takes into account the previous elements of the sequence. After the LSTM layer, an output Dense layer is used to decode the LSTM output into the most probable SIP dialog. In the second model, represented in Figure 1(b), two LSTM layers are considered to increase the number of degrees of freedom to identify other existing relations between each sequence being trained. Furthermore, to prevent that each model becomes overfitted we have used a dropout probability block and an early stop training condition.

Before describing the structure of the LSTM RNN models, we introduce how the SIP protocol was modeled and the definitions required to describe the SIP dialog prediction and detection. Considering the characteristics of the SIP protocol and how the multimedia sessions are created, the proposed scheme for prediction/detection of the SIP dialogs can be

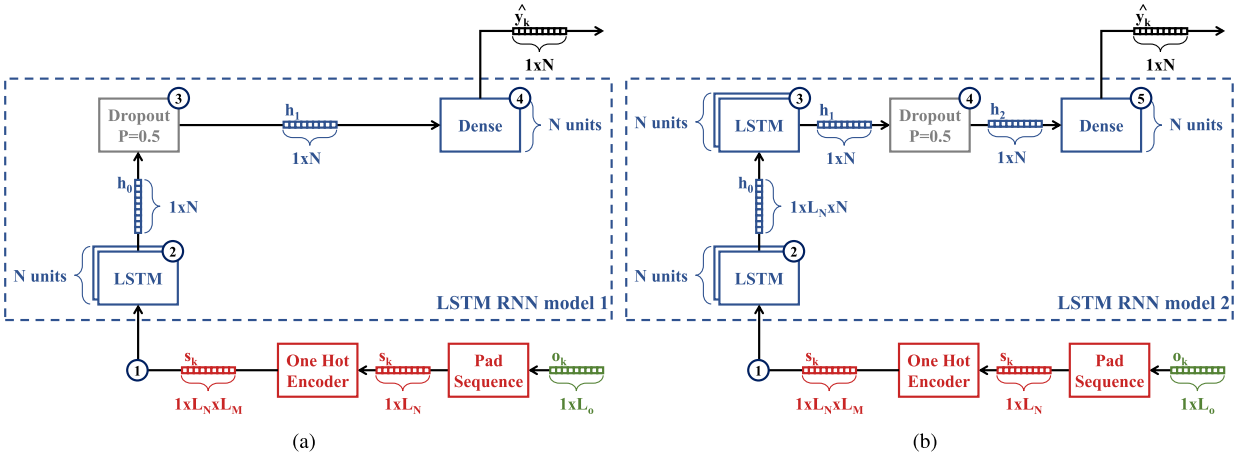


**FIGURE 1.** LSTM RNN models: (a) Model 1.0; (b) Model 2.0.

applied by the SIP user agents or in the SIP servers traversed by the SIP messages.

*Definition 1: A **SIP message** carried in a SIP packet and denoted by $m_k$, $k \in \mathcal{M} = \{1, 2, \ldots, M\}$, is a SIP request or SIP response of a specific type. We adopt the symbol $M$ to represent the total number of SIP request plus responses. Finally, $\mathcal{M}$ represents the set of the possible types of SIP requests and SIP responses.*

A SIP message can be formed either by a numerical code representing the type of the SIP response or a text field indicating the type of the SIP request. To use the type of the SIP message as an input of the learning process, we encode each SIP message $m_k$ using the One Hot Encoder algorithm into a unique Boolean vector univocally representing its type, thus making the representation of each type orthogonal to the others.

*Definition 2: An **encoded SIP message** $\mathbf{m}'_i$ is represented by a Boolean vector that univocally identifies the type of the SIP message $m_i$. The encoded message is obtained using a One Hot Encoder [26]. The Boolean vector has length $L_M$.*

The SIP messages are exchanged for a given purpose originating different transactions. A SIP dialog is completed when the multimedia session created by the peers of user agents is terminated.

*Definition 3: A sequence of consecutive SIP messages forms a **SIP dialog** denoted by $\mathbf{d}_k =< \mathbf{m}'^{(1)}, \mathbf{m}'^{(2)}, \ldots, \mathbf{m}'^{(L_d)} >$, where $\mathbf{m}'^{(j)}$ represents the j-th encoded message of the sequence. The length of the SIP dialog is represented by $L_d$. The SIP messages forming the SIP dialog contain the same Call ID string as well as the sender and receiver addresses in the packet's header.*

Although a SIP dialog is only defined when all SIP messages are exchanged, it is assumed that the model can estimate the dialog when only part of the SIP dialogs' messages have been exchanged. Therefore, instead of considering only sequences with length $L_d$, the model can process their subsequences, i.e., $1 \leq L_o \leq L_d$.

*Definition 4: An **observation** $k$ processed by a SIP user agent or server is a sequence of consecutive encoded SIP messages denoted as $\mathbf{o}_k =< \mathbf{m}'^{(1)}, \mathbf{m}'^{(2)}, \ldots, \mathbf{m}'^{(L_o)} >$. To describe the consecutive relation of the messages, each encoded SIP message is represented by $\mathbf{m}'^{(j)} = \mathbf{m}'_i$, $i \in \mathcal{M}$, $j \in \{1, 2, \ldots, L_o \leq L_d\}$, where $L_o$ represents the observation length. The SIP messages in the observation constitute a sub dialog or a complete dialog and, consequently, they share the same SIP Call ID.*

Because the observations can have different lengths ($1 \leq L_o \leq L_d$) and the LSTM model only processes sequences with the same length, the sequence describing the observation is transformed into a fixed-length stuffed sequence $\mathbf{n}_k$.

*Definition 5: A **padded sequence** $\mathbf{n}_k$ derived from each observation $\mathbf{o}_k$, is a sequence of length $L_N = L_o + n$, by adding $n$ zeros to the observation $\mathbf{o}_k$ as follows $\mathbf{n}_k =< \mathbf{o}_k, \underbrace{0, 0, \ldots, 0}_{(n)} >$. The length of the padded sequences is denoted by $L_N$.*

Until now, we have considered that an observed sequence was formed only by SIP messages. However, with the transformation proposed in Definition 5, a padding symbol is added to each $\mathbf{o}_k$. Thus, besides the encoding of each SIP message, the padding symbols are also encoded according to Definition 2. Therefore, the length of the encoded SIP message $\mathbf{m}'_k$ is $L_M = M + 1$ to account for every type of SIP message ($M$) and the zero-padding symbol.

Next, we describe the input and output state spaces used in the learning and prediction/detection of the SIP dialogs.

*Definition 6: The **input state space** of the supervised learning implemented through the LSTM RNN is the set $\mathcal{X}$ of padded sequences of the permutations with repetition represented by $\mathcal{X} = \{\mathbf{n}_1, \ldots, \mathbf{n}_k\}$, with $k = {L_M}^{L_N}$.*

*Definition 7: The set $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ represents the **output state space** of the neural network, where $N$ represents the total number of unique SIP dialogs in the training dataset*

*and each element* $\mathbf{y}_k, k \in \{1, \ldots, N\}$*, is the identifier of each unique SIP dialog* $\mathbf{d}_k$.

The computation of the predicted SIP dialog is equivalent to the regression problem $\hat{\mathbf{y}}_k = f(\mathbf{n}_k, \beta)$, where the estimate function $f(.)$ is defined by interactively computing the weights of the LSTM ($\beta$) during the training period. Once trained, the LSTM neural network identifies how close the observation $\mathbf{n}_k$ is to each dialog in the output space. Depending on the observation length $L_o$ the model is either **predicting** ($L_o < L_d$) or **detecting** ($L_o = L_d$) a SIP dialog.

Finally, the training steps and the topology of each LSTM RNN model are described in Tables 2 and 3.

**TABLE 2. LSTM RNN model 1.**

| | |
|---|---|
| Step 1: | An input sequence $\mathbf{n}_k$ of length $1 \times L_N \times M$ is generated by the One Hot Encoder and the Padded Sequence. |
| Step 2: | The LSTM layer processes the encoded SIP message $\mathbf{m'}_k$ of the padded sequence $\mathbf{n}_k$ and returns a $1 \times N$ sequence, $\mathbf{h}_0$, of real numbers in $[-1, 1]$. |
| Step 3: | The model discards the LSTM outputs with probability $P$. |
| Step 4: | The Dense layer receives the outputs from the Dropout block and generates an output vector of length $1 \times N$ of real numbers in $[0, 1]$. |

**TABLE 3. LSTM RNN model 2.**

| | |
|---|---|
| Step 1: | Similar to Step 1 in Table 2. |
| Step 2: | The LSTM layer processes each encoded SIP message $\mathbf{m}_k$ of the padded sequence $\mathbf{n}_k$ and returns a $1 \times L_N \times N$ sequence $\mathbf{h}_0$ of real numbers in $[-1, 1]$. |
| Step 3: | A second LSTM layer processes $\mathbf{h}_0$ and returns the sequence $\mathbf{h}_1$ of $1 \times N$ real numbers in $[-1, 1]$. |
| Step 4: | Similar to Step 3 in Table 2. |
| Step 5: | Similar to Step 4 Table 2. |

### B. UNKNOWN SIP DIALOGS DETECTOR

During the training of the LSTM RNN models, they acquire the ability to differentiate each SIP dialog, denominated **trained dialog**. However, when a SIP dialog that was never seen during the training stage, denominated as **unknown dialog**, is copied to the inputs of the LSTM RNN model, the neural network also generates output values. The detection methodology to identify unknown dialogs is based on the statistical properties of the LSTM RNN output values and the rationale behind the detection is based on the statistical dissimilarity of the outputs when the input is a known or an unknown SIP dialog.

The first classifier detects possible anomalies by looking into the maximum value of the LSTM RNN model outputs. Whenever a sequence is predicted the maximum output value is compared with the average of the maximum values obtained for the trained/known dialogs. Depending on the maximum value of the outputs the classifier decides if it is a **trained/known dialog** or an **unknown dialog**. In terms of detection, we define the hypotheses $H_0$ and $H_1$. Considering that the hypothesis $H_0$ represents the detection of a known SIP dialog (previously trained) and the hypothesis $H_1$ represents the detection of an unknown SIP dialog, the classification of a predicted output sequence is stated as

$$H_0 : \max(\hat{\mathbf{y}}_k) \geq \lambda_M,$$
$$H_1 : \max(\hat{\mathbf{y}}_k) < \lambda_M,$$

where $\lambda_M$ is the mean of the maximum value of the $N$ LSTM outputs obtained for each dialog during the training stage and $\max(\hat{\mathbf{y}}_k)$ represents the highest LSTM output value of the dialog to be classified.

In the second classifier, all outputs are used as a source of information for the classification of unknown dialogs. The second classifier is based on statistical metrics computed from the outputs of the LSTM RNN neural network, particularly the skewness and kurtosis standardized central moments. Therefore, whenever a SIP dialog is detected the skewness and the kurtosis of the LSTM RNN outputs are computed and compared with the thresholds given by $\lambda_S = \mu_S - \sigma_S^2$ and $\lambda_K = \mu_K - \sigma_K^2$, respectively. The variables $\mu_S$ and $\mu_K$ represent the mean of the skewness and kurtosis of the LSTM RNN outputs obtained for the trained dataset and $\sigma_S^2$ and $\sigma_K^2$ denote their variance, respectively. Thus, as in the previous classifier, whenever a sequence is classified two hypotheses are tested, representing a **trained dialog** (hypothesis $H_o$) or an **unknown dialog** (hypothesis $H_1$). The hypotheses are written as

$$H_0 : Skew(\hat{\mathbf{y}}_k) \geq \lambda_S, Kurt(\hat{\mathbf{y}}_k) \geq \lambda_K,$$
$$H_1 : Skew(\hat{\mathbf{y}}_k) < \lambda_S, Kurt(\hat{\mathbf{y}}_k) < \lambda_K,$$

where $Skew(\hat{\mathbf{y}}_k)$ and $Kurt(\hat{\mathbf{y}}_k)$ represent the skewness and kurtosis of the LSTM RNN outputs of the SIP dialog to be classified.

## IV. PERFORMANCE EVALUATION

In the following subsections, we evaluate the performance of the proposed LSTM RNN models to predict or detect a SIP dialog formed by a sequence of observed SIP messages - objective (a). Furthermore, considering that unknown dialogs might be observed, we evaluate the performance of the detectors proposed in Subsection IV-B to classify them - objective (b). Both objectives are important to detect signaling SIP attacks. In objective (a) the model is capable of classifying the dialogs it already knows and labeled as safe, anomalous, or according to different vulnerabilities ranks. Consequently, the importance of a model with a higher detection and prediction performance can be leveraged to obtain more accurate results with regards to the classification of safe or harmful dialogs previously known. Additionally, with objective (b) the model gains the ability to recognize if the observed SIP dialog was considered during the training stage or if it is unknown, the latter representing the case when it

should be analyzed by an expert domain to assess its vulnerability level.

Regarding the organization of this section, Subsections IV-B and IV-C evaluate the objective (a) by characterizing the classification performance of the SIP dialogs already trained so far. Subsection IV-D addresses the objective (b) by evaluating the capability of detecting unknown SIP dialogs. The experimental methodology is presented in Subsection IV-A.

### A. EXPERIMENTAL METHODOLOGY AND DATASETS

To evaluate the performance of each model in the following experiments we adopted the SIP dataset created by Nassar *et al.* [27]. The dataset was selected to enable the comparison between the proposed LSTM RNN models with the performance obtained using a *n*-gram Hidden Markov Model described in [6].

The SIP dataset is composed of two datasets: one for the non-anomalous dialogs, and another one for the anomalous dialogs. The non-anomalous dataset contains 18782 SIP dialogs created by 249 user agents. The 18782 dialogs correspond to a total of 1492 unique SIP dialogs in which 66.23% only occur once. Furthermore, each dialog is formed by a combination of a maximum of 17 types of unique SIP messages and the length of the combination varies between 3 and 56. As in [6], we have considered the non-anomalous dataset for training and testing the LSTM RNN model prediction and detection performance. The non-anomalous dataset was divided into a training and test datasets in a proportion of 80/20. The test dataset contains the last 20% of the dialogs exchanged by each user and the training dataset contains the remaining 80% of the dialogs. Regarding the anomalous dataset, it contains 152 unique SIP dialogs representing possible attacks.

Some of the LSTM RNN topological parameters are based on the distribution of the dialogs of the training dataset. The number of unique SIP dialogs ($N$) used in the training stage is 1043 (not the 1492 in the entire non-anomalous dataset due to the 80/20 proportion). Therefore, some dialogs are only contained in the test dataset (more precisely 449 dialogs) and are not used during the training stage. Besides the value of unique SIP dialogs $N$, the remaining parameters adopted in the LSTM RNN model are described in Table 4. The LSTM RNN models were implemented in TensorFlow 2.0 running in a 64bit Ubuntu 20.04 OS system over an Intel Core(TM) i5-5200U CPU @ 2.20GHz with 8 GB of RAM and a GeForce 840M GPU.

### B. DETECTION PERFORMANCE

This subsection evaluates the detection performance of each LSTM RNN model after the training phase. The number of training epochs for each model was 438 (model 1) and 289 (model 2) as a result of the use of the early stop condition. To assess the detection performance we have computed the detection probability ($P_D$) for each LSTM RNN model in the training and test datasets. The detection probability

**TABLE 4. LSTM RNN parameters.**

| Model Parameters | |
|---|---|
| $M$ | 17 |
| $L_M$ | 18 |
| $L_N$ | 56 |
| $N$ | 1043 |
| LSTM layer units | 1043 |
| Dense layer units | 1043 |
| Dense layer activation function | Softmax |
| Dropout probability | P = 0.5 |
| Early Stopping condition | Minimum of the test loss |
| Batch size | 64 |
| Loss Function | Categorical cross entropy |
| Optimizer | Adam (learning rate = 0.001) |

**TABLE 5. $P_D$ achieved in the train and test stages.**

| Model | train dataset | test dataset | joint dataset |
|---|---|---|---|
| HMM EdC [6] | 0.9927 | 0.8623 | 0.9651 |
| HMM MFdC [6] | 1.0000 | 0.8636 | 0.9712 |
| LSTM RNN m1 | 1.0000 | 0.8636 | 0.9712 |
| LSTM RNN m2 | 1.0000 | 0.8636 | 0.9712 |

expresses the probability of the LSTM RNN model output ($\hat{\mathbf{y}}_k$) indicate the correct SIP dialog ($\mathbf{y}_k$). Table 5 presents the achieved detection probability. The results indicate that the LSTM RNN models achieve a similar detection probability. Although the proposed models 1 and 2 are capable of detecting all SIP dialogs of the training dataset, some of the SIP dialogs of the test dataset are not detected because they were not included in the training dataset. Regarding the detection probabilities, we observe that they are identical to the results achieved with the HMM approach with the MFdC criteria proposed in [6].

The computation time required to classify the SIP dialogs is of high importance since it represents the time required to run the LSTM RNN model during the detection of SIP dialogs. To show the potential of the proposed solution we measured the amount of time each LSTM RNN model needs to compute the output for a given observed sequence. The Cumulative Distribution Probability (CDF) of the computation time required to detect each SIP dialog in the non-anomalous dataset is represented in Figure 2 for models 1 and 2, which are compared with the times obtained with the detection scheme proposed in [6] (HMM). The results indicate that the HMM model achieves lower computation times for approximately 50% of the dialogs in the dataset. However, the times for the remaining dialogs are much higher than the times achieved with the LSTM RNN models. For the HMM approach, the detection of the SIP dialogs is based on the Viterbi algorithm that computes the most probable SIP dialog according to the number of observed SIP messages. Afterward, a backward search is used to obtain the output sequence of the model. Thus, the complexity of the algorithm is $\mathcal{O}\left(\gamma M^{L_o} N^2\right)$, where $\gamma$ denotes

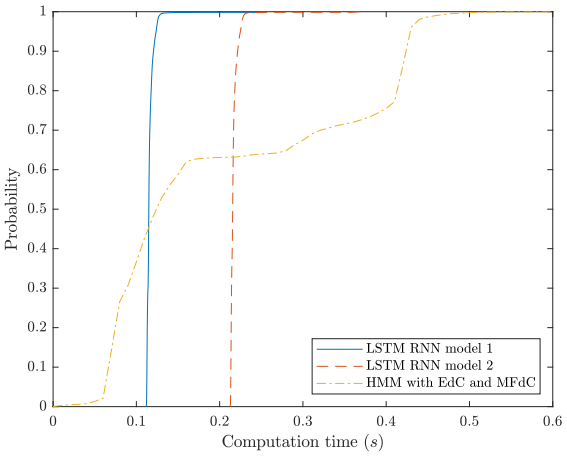


**FIGURE 2.** CDFs of the detection computation times.

the length of $n$-gram sequence presented to the HMM model. As described in the computational complexity expression, the computation time of the HMM is a function of the observation length, and lower computation times are achieved for shorter observations and vice-versa. Regarding the LSTM RNN models, the decision is made through a mapping function that does not depend on the observation length, i.e., $\mathcal{O}(1)$. Finally, the average computation time is 240 ms, 116 ms, and 217 ms for the HMM model, LSTM RNN model 1, and LSTM RNN model 2, respectively. The difference between the two LSTM RNN models is related to the number of parameters they have. As the LSTM RNN model becomes more complex, the time needed to compute the output increases, as observed for the LSTM RNN model 2. Therefore, the model selected to perform the detection of SIP dialogs is the LSTM RNN model 1.

### C. PREDICTION PERFORMANCE

This subsection evaluates the models' ability to estimate the most probable SIP dialogs when the observed sequence is still being transmitted, i.e., as the SIP dialog evolves. To this end, the LSTM RNN models were retrained, using a prediction dataset (during 120 epochs for model 1 and 57 epochs for model 2). The prediction dataset is based on the dataset used during the detection. However, instead of considering that there is only one observed sequence per SIP dialog we have considered $L_d$ SIP sequences that represent the subsequences from the instant the first SIP message is observed until the growth of the observed sequence reaches the $L_d$ SIP messages that form the SIP dialog. Therefore, each SIP dialog is decomposed in the following observed sequences $\mathbf{o}_1 =< m^{(1)} >$, $\mathbf{o}_2 =< m^{(1)}, m^{(2)} >, \ldots, \mathbf{o}_{L_d} =< m^{(1)}, m^{(2)}, \ldots, m^{(L_d)} >$. As a consequence, the 18782 dialogs of the non-anomalous dataset are decomposed into sequences that create the prediction dataset. Thus, the train and test datasets are formed by 132855 and 43503 sequences, respectively.

As in the previous subsection, the performance of the LSTM RNN models is evaluated through the probability of the models' output $\hat{\mathbf{y}}_k$ be identical to the correct SIP dialog identifier $\mathbf{y}_k$. However, unlike the previous subsection, the prediction probability ($P_E$) is computed for the observed sequences with $L_o < L_d$, while the detection probability considers only the observed sequences with $L_o = L_d$. Table 6 presents the prediction probability $P_E$ of each LSTM RNN model in the training and test datasets. The results indicate that the prediction performance of the LSTM RNN models is higher than the one obtained for the HMM approach. The different prediction probabilities between the LSTM RNN and the HMM models are related to how the observed sequences are transformed in these two approaches. In the case of the LSTM RNN model, each observed sequence $\mathbf{o}_k$ is stuffed with zeros at the end to form a fixed-length padded sequence $\mathbf{n}_k$, where all their SIP messages are orthogonalized through the One Hot Encoder. In the HMM approach, the observed sequence is also stuffed with zeros. But in this approach, the zeros are placed at the beginning and at the end of the observation, and no guaranty of orthogonality with all the different sequences is assured, thus it is expected to achieve a lower performance.

**TABLE 6.** $P_E$ achieved in the train and test stages.

| Model | train dataset | test dataset | joint dataset |
|---|---|---|---|
| HMM EdC [6] | 0.3801 | 0.3152 | 0.3638 |
| HMM MFdC [6] | 0.3802 | 0.3152 | 0.3639 |
| LSTM RNN m1 | 0.5497 | 0.4861 | 0.5338 |
| LSTM RNN m2 | 0.5500 | 0.4871 | 0.5342 |
| Theor. Upper Bound | 0.5933 | 0.6538 | 0.6085 |

Regarding the prediction performance of each LSTM RNN model, we conclude that as the complexity of the model increases so does the prediction probability. The theoretical upper bound of the prediction probability is also indicated in Table 6. The theoretical value of $P_E$ is the result of the summation of the number of occurrences of the most frequent dialog for each observed sequence $\mathbf{o}_k$ divided by the size of the dataset. Therefore, we conclude that the LSTM RNN results are closer to the theoretical upper bound than the ones obtained with the HMM approach.

Given that the prediction probability is computed considering that each SIP dialog is formed by $L_d$ subsequences $(\mathbf{o}_1, \ldots, \mathbf{o}_{L_d})$, in Figure 3 we plot the prediction probability conditioned by the number of received SIP messages ($L_o < L_d$). The results show that the LSTM RNN model 2 outperforms the HMM with MFdC criteria when the number of observed SIP messages is between 1 and 11. Afterward, their probabilities are similar. Therefore, we conclude that the LSTM RNN model predicts more SIP dialogs for a lower number of received messages. Regarding the behavior of the prediction probability for the LSTM RNN model, its performance can be divided into two regions: $L_o < 15$ and $L_o \geq 15$. In the first region, the prediction probability gradually increases as the length of the observed sequences increases and, consequently, as the number of likely SIP

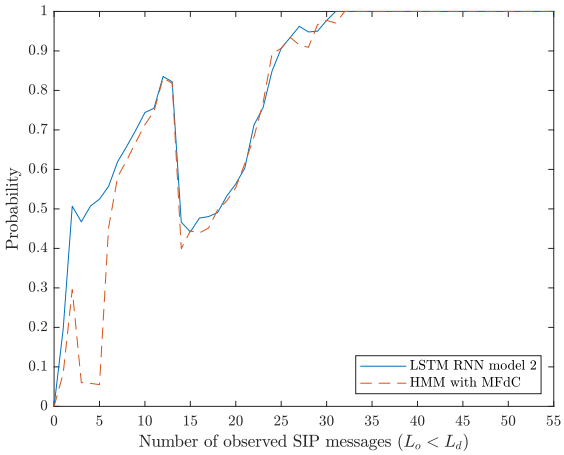


**FIGURE 3.** Prediction probability of SIP dialogs for different lengths of the observed input sequence.

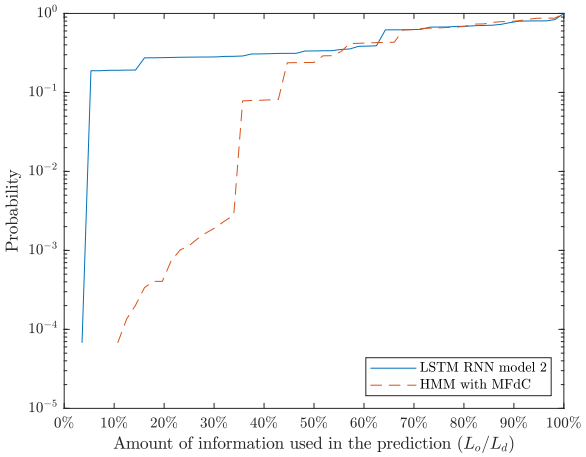


**FIGURE 4.** SIP dialogs prediction probability over the amount of available information (SIP messages).

dialogs in the output space decreases. However, when the number of received SIP messages is 15 the prediction performance decreases. The justification for the lower probability is related to the higher occurrence of the dialogs with $L_d \leq 15$, in comparison with the dialogs with $L_d > 15$. Besides that, there is a higher number of unique dialogs of length above 15. Nevertheless, as the length of the observation increases so does the prediction probability. Furthermore, when $L_o = 31$ the LSTM RNN model 2 can predict all the SIP dialogs with $L_d \geq 32$.

Figure 4 presents the prediction probability conditioned to the amount of information received so far ($L_o/L_d$), i.e., the length of each observation is normalized in respect to its dialog length. The results show the amount of information needed to predict each dialog. The curves were obtained by predicting every observed sequence from each SIP dialog $\mathbf{d}_k$. The results show that the LSTM RNN model can predict each observation sequence with less available information than the HMM approach. The conclusion is supported considering that the prediction probability is much higher than the one obtained for the HMM model when $L_o/L_d \leq 42.86\%$. After that value, an identical performance is observed for both models. Additionally, the minimum amount of information needed to predict a SIP dialog is 3.571% and 10.71% for the LSTM RNN model 2 and HMM with MFdC, respectively. Finally, to predict 50% of all the SIP dialogs the LSTM RNN model 2 and the HMM with MFdC need approximately 64.29% and 67.86% of the available information, respectively.

### D. DETECTION OF UNKNOWN SIP DIALOGS

Despite the advantages of the LSTM RNN models demonstrated so far in terms of the computation time and prediction of SIP dialogs the same cannot be concluded for the detection of unknown dialogs. An indication of the inability to detect unknown dialogs is represented in Table 5, where both LSTM RNN models assigned an incorrect SIP dialog identifier for 13.64% of the sequences from the test dataset. The reason for the SIP dialogs misdetection is due to the nonexistence of those dialogs in the input and output state space leaving the model to assign them the identifier of the most similar SIP dialog contained in $\mathcal{Y}$. However, for the HMM approaches whenever an observed sequence different from the ones in the input state space is detected no output is returned from the prediction algorithm.

Regarding the proposed classifiers, presented in Subsection III-B, we seek to characterize its ability to detect the sequences from the anomalous dataset and the unknown dialogs included in the test dataset. The classification of each SIP dialog into a **trained/known dialog** or an **unknown dialog** is based on the statistical features collected from the output of the LSTM RNN model. In the first classifier, the statistical information collected is related to the maximum value of the LSTM RNN model 1 output, which is depicted in Figure 5. In the figure, the 1 dimensional distribution is replicated in both axis and its data is differentiated according to its characteristics: **anomalous dialog** (anomalous dataset), **unknown dialog** (13.64% of the test dataset), and **trained dialog** (training and 86.36% of the test dataset).

The results from the figure indicate that the trained dialogs have lower uncertainty and a higher maximum value in comparison with the other classes because the LSTM RNN model was trained to detect those dialogs. Figure 6 illustrates the computed threshold ($\lambda_M = 0.99985$), and the classifier performance.

Regarding the classification performance there are four possible outcomes: the dialog was correctly classified as a trained dialog (**true positive**), incorrectly classified as a trained dialog (**false positive**), and classified as unknown dialog (**true negative** and **false negative**). According to the results illustrated in Figure 6, the classifier cannot completely separate the two classes. A reason to support the existence of false positives despite the higher threshold value can be related to the similarity between the unknown and the trained dialogs, which lead to higher LSTM RNN model output values.

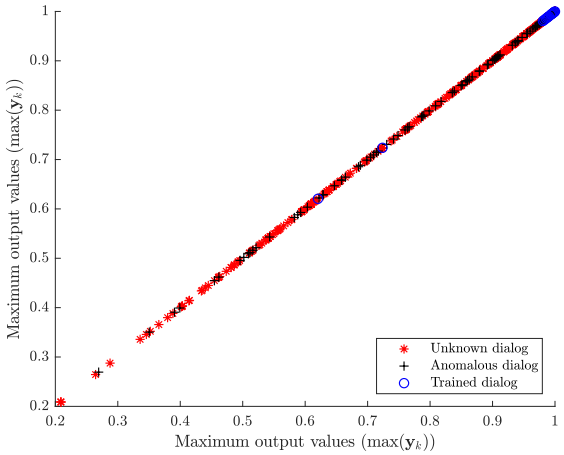


**FIGURE 5.** LSTM RNN model 1 maximum output value.

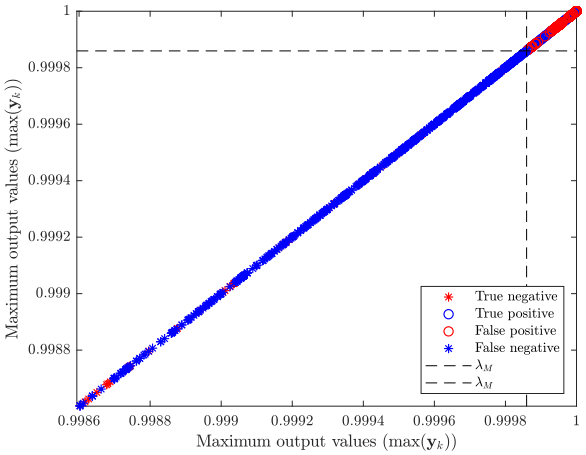


**FIGURE 6.** Maximum output value threshold classifier.

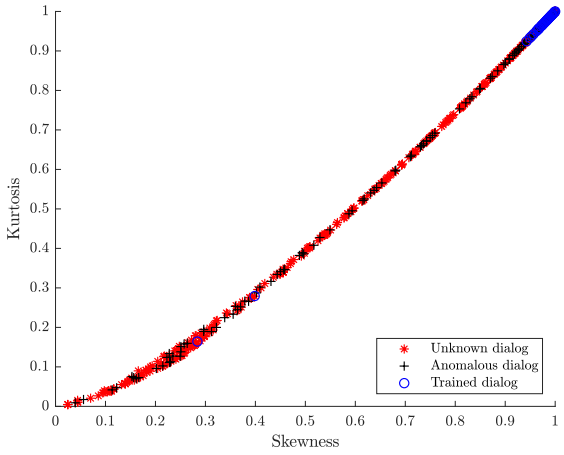


**FIGURE 7.** Normalized skewness and kurtosis of the LSTM RNN model 1 output values.

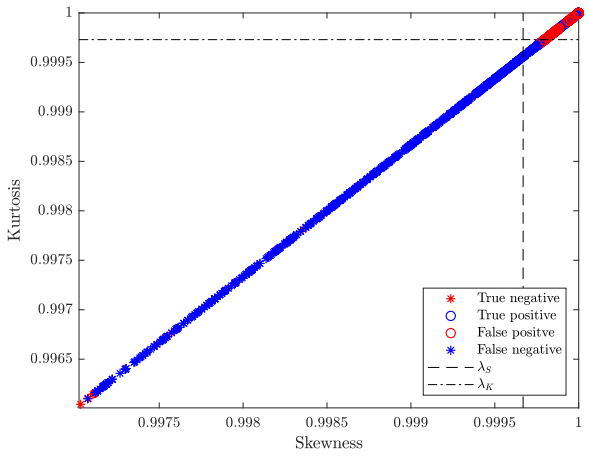


**FIGURE 8.** Skewness and kurtosis thresholds classifier.

In the second classifier, the classification of the SIP dialogs is based on the skewness and kurtosis standardized central moments of the LSTM RNN model 1. The statistical information collected for the proposed classifier is represented in Figure 7 through the normalized skewness and kurtosis. As in the previous classifier, the selection of the threshold value took into consideration the distribution of the trained dialogs, especially their average and variance. Figure 8 illustrates the classifier performance and its threshold values computed with the proposed detection model ($\lambda_K = 0.99967$ and $\lambda_S = 0.99973$). Similar to the previous classifier, the skewness and kurtosis threshold-based classifier cannot differentiate the two classes. The justification for missing the detection of some dialogs is identical to the one presented for the maximum value threshold-based classifier.

Finally, Table 7 presents the performance metrics to compare the proposed classifiers and the performance of HMM model from [6]. The metrics are based on the four possible outcomes already presented (confusion matrix): true positive, true negative, false positive, and false negative. Considering the results from the confusion matrix only, we observe that the maximum value threshold-based classifier is the one that correctly classifies more trained dialogs, while the skewness and kurtosis threshold-based classifier distinguishes more unknown dialogs. The results previously stated are also validated through the sensitivity and specificity metrics, since the former quantifies the probability of correctly classifying

**TABLE 7.** Performance evaluation of the unknown SIP dialogs classifiers.

| Model | HMM [6] | Classifier 1 | Classifier 2 |
|---|---|---|---|
| True negative ($tn$) | 1.000 | 0.9430 | 0.9451 |
| True positive ($tp$) | 1.000 | 0.9365 | 0.9189 |
| False positive ($fp$) | 0.000 | 0.0635 | 0.0811 |
| False negative ($fn$) | 0.000 | 0.0570 | 0.0549 |
| Specificity $\left(\frac{tn}{tn+fp}\right)$ | 1.000 | 0.9430 | 0.9451 |
| Sensitivity $\left(\frac{tp}{tp+fn}\right)$ | 1.000 | 0.9365 | 0.9189 |
| Precision $\left(\frac{tp}{tp+fp}\right)$ | 1.000 | 0.9718 | 0.9723 |
| Accuracy $\left(\frac{tp+tn}{tp+tn+fp+fn}\right)$ | 1.000 | 0.9386 | 0.9274 |
| F1-Score | 1.000 | 0.9538 | 0.9449 |

a trained dialog considering all the trained dialogs, while the latter represents the probability of correctly classifying an unknown dialog considering all the unknown dialogs. Regarding the precision and accuracy of the classifiers, the second classifier achieves higher precision, while the first one has higher accuracy. The f1-score metric, used when the results are obtained from unbalanced data and the classifier's outcome is binary, exhibits a higher score for Classifier 1. Thus, showing that the performance achieved by the two classifiers are too close to each other and the superiority of each one effectively depends on the considered performance metric. Comparing the performance of both classifiers with the HMM model proposed in [6], we observed that the HMM model completely distinguishes both classes but exhibits higher computational time.

## V. CONCLUSION

A deep learning approach is proposed in this work to detect and predict known and unknown SIP dialogs. The proposed solution is based on a LSTM neural network, which can predict and detect SIP dialogs already observed so far. Two detectors are also proposed to detect SIP dialogs never observed before. Adopting a publicly available SIP dataset, we have assessed the performance of the proposed classifier and detectors. Several performance metrics were evaluated, including the detection and prediction probabilities and computation time. Moreover, the experimental results were compared to a probabilistic-based solution, showing that the proposed methods achieve higher SIP dialogs detection scores in a shorter time. Finally, the detection probability of unknown SIP dialogs is above 94%, indicating a significant capability to detect a high number of unknown SIP dialogs in a short amount of time.

## REFERENCES

[1] F. Belqasmi, C. Fu, M. Alrubaye, and R. Glitho, "Design and implementation of advanced multimedia conferencing applications in the 3GPP IP multimedia subsystem," *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 156–163, Nov. 2009.

[2] A. Uzelac and Y. Lee, *Voice Over IP (VOIP) Sip Peering Use Cases*, document RFC 6405, Internet Requests for Comments RFC Editor, Nov. 2011.

[3] A. Lahmadi and O. Festor, "A framework for automated exploit prevention from known vulnerabilities in voice over IP services," *IEEE Trans. Netw. Service Manage.*, vol. 9, no. 2, pp. 114–127, Jun. 2012.

[4] D. Golait and N. Hubballi, "Detecting anomalous behavior in VoIP systems: A discrete event system modeling," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 730–745, Mar. 2017.

[5] D. Geneiatakis, T. Dagiuklas, G. Kambourakis, C. Lambrinoudakis, S. Gritzalis, K. S. Ehlert, and D. Sisalem, "Survey of security vulnerabilities in session initiation protocol," *IEEE Commun. Surveys Tuts.*, vol. 8, no. 3, pp. 68–81, 3rd Quart., 2006.

[6] D. Pereira, R. Oliveira, and H. S. Kim, "A machine learning approach for prediction of signaling SIP dialogs," *IEEE Access*, vol. 9, pp. 44094–44106, 2021.

[7] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "*SIP: Session Initiation Protocol*, document RFC 3261, Internet Requests for Comments, RFC Editor, Jun. 2002.

[8] S. Ehlert, D. Geneiatakis, and T. Magedanz, "Survey of network security systems to counter SIP-based denial-of-service attacks," *Comput. Secur.*, vol. 29, no. 1, pp. 225–243, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404809001060

[9] D. Sisalem, J. Kuthan, and S. Ehlert, "Denial of service attacks targeting a SIP VoIP infrastructure: Attack scenarios and prevention mechanisms," *IEEE Netw.*, vol. 20, no. 5, pp. 26–31, Sep. 2006.

[10] H. Li, H. Lin, H. Hou, and X. Yang, "An efficient intrusion detection and prevention system against SIP malformed messages attacks," in *Proc. Int. Conf. Comput. Aspects Social Netw.*, Sep. 2010, pp. 69–73.

[11] M. Nassar, R. State, and O. Festor, "Monitoring sip traffic using support vector machines," in *Recent Advances in Intrusion Detection*, R. Lippmann, E. Kirda, and A. Trachtenberg, Eds. Berlin, Heidelberg: Springer, 2008, pp. 311–330.

[12] N. Hentehzadeh, A. Mehta, V. K. Gurbani, L. Gupta, T. K. Ho, and G. Wilathgamuwa, "Statistical analysis of self-similar session initiation protocol (SIP) messages for anomaly detection," in *Proc. 4th IFIP Int. Conf. New Technol., Mobility Secur.*, Feb. 2011, pp. 1–5.

[13] H. Arshad and M. Nikooghadam, "An efficient and secure authentication and key agreement scheme for session initiation Protocol using ECC," *Multimedia Tools Appl.*, vol. 75, no. 1, pp. 181–197, Jan. 2016, doi: 10.1007/s11042-014-2282-x.

[14] Y. Zhang, K. Xie, and O. Ruan, "An improved and efficient mutual authentication scheme for session initiation protocol," *PLoS ONE*, vol. 14, no. 3, pp. 1–15, Mar. 2019, doi: 10.1371/journal.pone.0213688.

[15] I. M. Tas, B. G. Unsalver, and S. Baktir, "A novel SIP based distributed reflection denial-of-service attack and an effective defense mechanism," *IEEE Access*, vol. 8, pp. 112574–112584, 2020.

[16] S. Marchal, A. Mehta, V. K. Gurbani, R. State, T. Kam-Ho, and F. Sancier-Barbosa, "Mitigating mimicry attacks against the session initiation protocol," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 3, pp. 467–482, Sep. 2015.

[17] D. Bao, D. L. Carni, L. D. Vito, and L. Tomaciello, "Session initiation protocol automatic debugger," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 6, pp. 1869–1877, Jun. 2009.

[18] T. T. Nguyen, G. Armitage, P. Branch, and S. Zander, "Timely and continuous machine-learning-based classification for interactive IP traffic," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1880–1894, Dec. 2012.

[19] K. Toyoda, M. Park, N. Okazaki, and T. Ohtsuki, "Novel unsupervised SPITters detection scheme by automatically solving unbalanced situation," *IEEE Access*, vol. 5, pp. 6746–6756, 2017.

[20] H. Yang, Z. Yang, Y. Bao, S. Liu, and Y. Huang, "Fast steganalysis method for VoIP streams," *IEEE Signal Process. Lett.*, vol. 27, pp. 286–290, 2020.

[21] Z. Lin, Y. Huang, and J. Wang, "RNN-SM: Fast steganalysis of VoIP streams using recurrent neural network," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1854–1868, Jul. 2018.

[22] F. U. Islam, G. Liu, J. Zhai, and W. Liu, "VoIP traffic detection in tunneled and anonymous networks using deep learning," *IEEE Access*, vol. 9, pp. 59783–59799, 2021.

[23] Y. Huang, B. Li, M. Barni, and J. Huang, "Identification of VoIP speech with multiple domain deep features," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2253–2267, 2020.

[24] E. Cipressi and M. L. Merani, "An effective machine learning (ML) approach to quality assessment of voice over IP (VoIP) calls," *IEEE Netw. Lett.*, vol. 2, no. 2, pp. 90–94, Jun. 2020.

[25] P. Charonyktakis, M. Plakia, I. Tsamardinos, and M. Papadopouli, "On user-centric modular QoE prediction for VoIP based on machine-learning algorithms," *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1443–1456, Jun. 2016.

[26] D. Harris and S. Harris, *Digital Design and Computer Architecture*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2012.

[27] M. Nassar and O. Festor, "Labeled VoIP data-set for intrusion detection evaluation," in *Meeting of the European Network of Universities and Companies in Information and Communication Engineering*. Berlin, Germany: Springer, 2010, pp. 97–106.

**DIOGO PEREIRA** received the B.Sc. and M.Sc. degrees in electrical and computer engineering from the NOVA School of Science and Technology (FCT NOVA), where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests include the areas of stochastic processes applied to computer and data science, wireless mobile networks, and network modeling.

**RODOLFO OLIVEIRA** (Senior Member, IEEE) received the Licenciatura degree in electrical engineering from the Faculdade de Ciências e Tecnologia (FCT), Universidade Nova de Lisboa (UNL), Lisbon, Portugal, in 2000, the M.Sc. degree in electrical and computer engineering from the Instituto Superior Técnico, Technical University of Lisbon, in 2003, and the Ph.D. degree in electrical engineering from UNL, in 2009. From 2007 to 2008, he was a Visiting Researcher with the University of Thessaly. From 2011 to 2012, he was a Visiting Scholar with Carnegie Mellon University. He is currently with the Department of Electrical and Computer Engineering, UNL, and is also affiliated as a Senior Researcher with the Instituto de Telecomunicações, where he researches in the areas of wireless communications, computer networks, and computer science. He serves in the Editorial Board for *Ad Hoc Networks*, (Elsevier), IEEE Open Journal of the Communications Society, and IEEE Communications Letters.

**HYONG S. KIM** (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electrical engineering from McGill University, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto.

He has been with Carnegie Mellon University, since 1990, where he is currently a Drew D. Perkins Chaired Professor in electrical and computer engineering. His Tera ATM switch architecture developed at CMU has been licensed for commercialization to AMD and Samsung. In 1995, he founded Scalable Networks, a Gigabit-Ethernet switching startup. Scalable Networks was later acquired by FORE Systems, in 1996. In 2000, he founded AcceLight Networks, an optical networking startup, and was CEO of AcceLight Networks, until 2002. He founded and directed the CyLab Korea, International Cooperative Research Center, Carnegie Mellon University, from 2004 to 2008. He is an author of over 130 published papers and holds over ten patents in networking and computing technologies. His research interests include advanced switching architectures, fault-tolerant, reliable, secure networks and computer system architectures, distributed computing, and network management systems.