IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 28, 2021, accepted March 9, 2021, date of publication March 12, 2021, date of current version March 25, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3065660*

# A Machine Learning Approach for Prediction of Signaling SIP Dialogs

**DIOGO PEREIRA[1], 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 and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Corresponding author: Rodolfo Oliveira (rado@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, Programa Operacional Regional LISBOA (LISBOA2020), in part by the Fundação para a Ciência e Tecnologia (FCT), through projects InfoCent-IoT under Grant POCI-01-0145-FEDER-030433, and in part by the CoSHARE under Grant LISBOA-01-0145-FEDER-0307095, Grant PTDC/EEI-TEL/30709/2017, and Grant UIDB/EEA/50008/2020.

**ABSTRACT** In this paper, we propose a machine learning methodology for prediction of signaling sessions established with the Session Initiation Protocol (SIP). Given the increasing importance of predicting and detecting abnormal sequences of SIP messages to avoid SIP signaling-based attacks, we first propose a Bayesian inference method capable of representing the statistical relation between a SIP message, observed by a SIP user agent or a SIP server, and prior trustworthy SIP dialogs. The Bayesian inference method, a Hidden Markov Model (HMM) enriched with $n-$gram Markov observations, is updated over time, so the inference can be used in real-time. The HMM is then used for predicting and detecting SIP dialogs through a lightweight implementation of Viterbi algorithm for sparse state spaces. Experimental results are also reported, where a SIP dataset representing prior information collected by a SIP user agent and/or a SIP server is used to predict or detect if a received sequence of SIP messages is legitimate according to similar SIP dialogs already observed. Finally, we discuss the results obtained for a dataset of abnormal SIP sequences, not observed during the inference stage, showing the effective utility of the proposed methodology to detect abnormal SIP sequences in a short period of time.

**INDEX TERMS** Session initiation protocol, hidden Markov chains, Bayesian networks, machine learning.

## I. INTRODUCTION

Nowadays, the Session Initiation Protocol (SIP) has been massively adopted for establishing and controlling communication sessions that support multimedia services, including but not limited to Voice over Internet Protocol (VoIP) [1] or IP Multimedia Subsystem (IMS) services over cellular networks [2], [3]. The growing adoption of the SIP protocol has motivated the development of analytical tools capable of diagnosing and/or identifying potential problems caused by inconsistent SIP signaling. It is well reported in the literature that a significant number of SIP vulnerabilities is related to the exploitation of SIP weaknesses found in the protocol itself, or in the implementation [4], [5]. By exploring combinations of different signaling patterns, the attackers can cause denial-of-service, unauthorized access to a call, billing errors, and others [6]. Consequently, it is of high importance identifying potential malicious SIP signaling sequences at the SIP servers, or new signaling sequences never observed before that can represent new attacks. While the already known potential malicious sequences can be detected or predicted in an automated way, the SIP sequences never observed before also need to be detected to be analyzed by domain experts who can then assess their level of vulnerability. The detection and prediction of those SIP signaling sequences can then be used by the SIP servers to act according to their vulnerability severity, avoiding attacks.

The detection and/or prediction of specific SIP signaling sequences is a challenging topic due to the high number of different signaling sequences and because its length is not

The associate editor coordinating the review of this manuscript and approving it for publication was Yulei Wu.

constant. The high number of sequences increases the state space of sequences, increasing the detection or prediction computation time. The non-constant length of the sequences degrade the detection/prediction performance, because a long signaling sequence can contain a short sequence at the beginning, and both sequences can be legitimate until more information is known that can distinguish them.

In recent years, the adoption of machine learning and data mining techniques has been massively adopted in several areas of interest, due to the increase in processing power and to the scientific breakthroughs in data science. Machine learning has also being adopted in the telecommunications arena [7], [8], as it can bring many advantages in the processing of bulks of data that are generated by a plethora of different sources. Multiple tools have already been proposed to prevent SIP attacks caused by SIP message payload tampering [9], [10], and SIP message flooding [11], [12]. However, the attacks caused by SIP message flow tampering, a.k.a. attacks based on SIP signaling [6], have received limited attention and, as far as we know, the prediction or detection of SIP signaling patterns as a given SIP agent/server receives the sequential SIP messages has not been addressed before.

In this work, we are motivated by the advantages of adopting machine learning techniques to predict and avoid vulnerabilities caused by the SIP signaling patterns generated by the multiple agents that support multimedia communication sessions. Admitting that a SIP server, or a SIP agent, has access to all SIP messages as they occur over time, we propose a methodology that is capable of predicting or detecting the type of SIP signaling pattern when a sequential set of messages is already known. The main contributions of this work are summarized as follows:

- An inference model is proposed to associate the already known SIP signaling patterns to the observed SIP messages. The association is represented through a hidden Markov model, where the number of known SIP patterns can increase over time, so it can dynamically incorporate more knowledge as more signaling patterns are observed;
- Each sequence of SIP messages observed by a SIP server/agent is represented by a $n-$gram, a form of a $(n-1)$-order Markov model, so we are able to represent the likelihood of observing a $n-$th SIP message given that the previous $n-1$ SIP messages are already known. Additionally, due to the properties of the $n-$gram to distinguish shorter observations included in longer ones, the prediction achieves high accuracy even for short patterns of SIP messages that can be found in multiple and different SIP dialogs;
- A modified version of the Viterbi algorithm is used to compute the predicted SIP signaling pattern given the set of observed SIP messages. The algorithm is designed to address data sparsity efficiently;
- We present extensive experimental results to evaluate the performance of the proposed methodology using a SIP dataset available in the community [13]. Starting with a brief characterization of the dataset, we assess the detection probability when all messages of the SIP pattern are observed, and the prediction probability as a function of the SIP messages already observed so far. Several results are given to demonstrate the capacity of the proposed method to identify unknown/abnormal SIP patterns, showing that it can be effectively adopted in practical systems.

The rest of the paper is organized as follows. A brief overview of the SIP protocol and its vulnerabilities and solutions to avoid attacks is given in Section II, based on the works already published so far. Section III introduces the methodology to build the inference model. The prediction scheme is presented in IV. The dataset and the results used to evaluate the prediction methodology are presented in V. Finally, conclusions and future work are discussed in Section VI.

Regarding the notation adopted in this work, we use $P(X = x)$ to represent the probability of $X$, and $F_X(x)$ to represent the Cumulative Distribution Function (CDF). Vectors are represented in lower case, upright boldface type, e.g. $\mathbf{x}$. Matrices are represented in upper case upright boldface type, e.g., $\mathbf{X}$, being the element at row $i$ and column $j$ represented by $x_{i,j}$. A vector of $k$ elements is represented by $\mathbf{v} = \{v_1, v_2, \ldots, v_k\}$. A vector of $k$ consecutive (ordered) elements, also denominated a sequence, is denoted by $\mathbf{v} = <v^{(1)}, v^{(2)}, \ldots, v^{(k)}>$. The notation $\mathbf{v}[m]$ is used to denote the $m$-th element of $\mathbf{v}$. Sets and the state spaces are represented in calligraphic font, e.g., $\mathcal{S}$.

## II. RELATED WORK

This section presents a brief overview of the SIP protocol, its vulnerabilities, and the most representative works proposed to mitigate the different types of vulnerabilities.

### A. SESSION INITIATION PROTOCOL

The Session Initiation Protocol [14] has been extensively deployed to support multimedia sessions, including but not limited to signaling of IMS services and VoIP and video services over cellular and fixed networks. The vulnerabilities of SIP are well documented in the literature [4]–[6]. Despite the adoption of multiple security mechanisms, SIP services can suffer different types of attacks that can cause service interruption, service destruction, or unauthorized access to previously reserved computing resources or pools of SIP services.

SIP [14] is an application-layer protocol designed to initiate, terminate and change multimedia sessions created by peers of user agents. To perform these interactions a series of SIP messages must be exchanged between each user agent. SIP is similar to HTTP and SMTP protocols, where each message can be either a request or a response.

When a user agent requests a certain interaction to occur, e.g. start a session or register in a server, first, a SIP message must be sent with a request that can be identified by a specific method (six methods are defined in [14], although standardized SIP extensions can define more). In response to one of those methods, a response SIP message is sent with a reply code, i.e., a three-digit number categorized into six classes (Provisional, Success, Redirection, Client Error, Server Error, and Global Failure).

Every SIP request exchanged between two or more user agents initiates a SIP transaction. A SIP transaction includes a single SIP request and any responses to it. Multiple SIP transactions exchanged between two peers form a SIP dialog, which represents the peer-to-peer relationship over time. Through the utilization of dialog IDs, a user agent can identify the different dialogs. The dialog is identified by three distinct elements: a Call ID, i.e., a unique identifier for every message on the actual dialog, a local tag, and a remote tag. The tags contain the Unique Resource Identifier (URI) from the sender and receiver user agent.

### B. SIP VULNERABILITIES

The types of reported SIP attacks caused by different vulnerabilities are usually categorized as [6]: flooding attacks; malformed-SIP message attacks; authentication attacks; and SIP signaling attacks. The majority of works published so far are mainly focused on flooding attacks, malformed-SIP message attacks, and authentication attacks. SIP signaling attacks have not deserved too much attention due to the complexity of the detection of the sequential signaling data in real-time and the large diversity of messages and SIP dialogs. The next paragraphs introduce the different types of SIP attacks and the exploited vulnerabilities that cause them.

SIP service interruption can be caused by **flooding attacks**, where a massive number of SIP requests are sent to SIP servers that are unable to process them. Different techniques have been proposed to avoid SIP flooding attacks, including threshold-based solutions that compare the traffic patterns occurring over time with the statistics of the network in normal operation [12]. The statistics can be described in a single dimension [12], or in multiple dimensions, the latter being more interesting to detect low rate flooding attacks [11]. SIP parser vulnerabilities can also be explored to deploy flooding attacks, where some fields of the SIP messages are changed to deplete the servers' processing power and unnecessarily long fields can be added to the headers to increase network utilization and service delay. An approach to mitigate parser-based flooding attacks is proposed in [15], where the authors recommend to not parse the SIP messages before passing them through a message classifier system that predicts the true class of each message and can simply drop them if they are suspicious.

The majority of literature on SIP anomaly detection is particularly oriented to **malformed-SIP message attacks** [16], which explore SIP parser vulnerabilities. The detection of malformed SIP messages has been studied in multiple works, since attacks may be attempted by simply changing the text-based SIP message headers. Malicious SIP messages are usually detected through intrusion detection systems (e.g. firewalls) [17], learning techniques [18], and/or identification of deviations from a priori statistics [9]. In [10] support-vector-machine classifiers were adopted to label incoming SIP messages as good or bad. A SIP message lexical analysis was developed to filter the messages that are not formed according to the standard and in a second stage a semantic filter was applied to the stream of the surviving messages to remove syntactic errors.

Apart from flooding and malformed-SIP messages attacks, **authentication attacks** are also of high importance [19]. In the authentication attacks several vulnerabilities of the authentication protocols are explored. Multiple schemes have been proposed to reduce the existing SIP authentication vulnerabilities, including password-based one-factor authentication [20], [21], two-factor authentication based on password and smart cards [22], [23], and three-factor authentication that includes biometric data [24], [25].

In addition to the types of attacks already mentioned, the attackers can also explore additional vulnerabilities related to the protocols' signaling logic and take advantage of defective implementations of the protocol in the servers and user agents. These types of attacks are known as **SIP signaling attacks** and explore possible protocol implementation errors by sending SIP messages to allow improper authentication mechanisms. These attacks include the ‘‘BYE-attack’’, the ‘‘CANCEL-attack’’, the ‘‘REFER-attack’’, and many others described in [6]. A common feature in these attacks is the possibility of capturing its ‘‘signature’’ through the pattern of SIP messages exchanged over the SIP servers. Several rule-based approaches, such as the one presented in [26], were proposed to mitigate SIP signaling vulnerabilities by capturing the contextual information in the SIP traffic, that is further used to construct event graphs to be matched with the protocol activities. These solutions explore known ‘‘signatures’’ that can be used further to policy the protocol. The main limitation of the ‘‘signature’’ based approach is the fact that the attacks can be perpetrated through new signatures, thus requiring dynamic algorithms that need to adapt to legitimate new signaling patterns. The solution proposed in this paper is a first-step to predict and detect such kinds of signatures, which can effectively be used to assess the vulnerability level as the SIP sessions are deployed in real-time. The SIP signaling vulnerabilities have also motivated the authors in [27] to propose a SIP automatic debugger tool to verify the compliance of SIP interoperability, being capable of analyzing the SIP messages flow and group them into dialogs to find protocols' compliance and interoperability faults. The mitigation of SIP signaling attacks was also addressed in [28], where the SIP operation is modeled as a discrete event system where the state transitions of the SIP dialog are described through a probabilistic counting deterministic timed automata. The description includes the characterization of the SIP sequences and their timings, which are used a posterior to detect deviations from the models.

We emphasize that the number of potential SIP dialogs is somehow constrained in normal operation. However, in SIP signaling attacks the malicious users explore the creation of new SIP dialogs that can take advantage of defective protocol implementations which are not observed during regular/normal operation. Although the number of different types of SIP messages is low, its combination in a SIP dialog can reach a high number of different patterns because the length of the SIP dialog is not limited. Moreover, the request/response is also not so deterministic, because for particular requests there are provisional responses that are sent back by servers according to the delays observed to process the requests, which can originate several responses to the same request.

As mentioned before the attacks can be perpetrated through new signatures, thus requiring dynamic algorithms that need to adapt to legitimate new signaling patterns. Contrarily to works in [26]–[28], we do not assume a fixed probabilistic model of the SIP operation or fixed rules that describe the SIP activity. Instead, we are interested in capturing the SIP messages to characterize the SIP dialogs during a dynamic probabilistic inference stage (presented in Section III), which can then be used as input to predict the SIP dialogs and/or indicate unknown dialogs that need to be categorized by domain experts. In this way, our goal is to learn from the captured data to assess the vulnerability of future SIP activities based on prior data and in an automated way. Consequently, the proposed method is designed assuming that new signatures can be identified, which can be further included in the knowledge base at any time. The dynamic detection and inclusion of new signatures improves the static schemes proposed so far in the literature, where the set of signatures is predefined and based on rules obtained from domain experts and not from information automatically detected without human intervention.

## III. INFERENCE MODEL FOR SIP SIGNALING PATTERNS

This section presents a model to infer the signaling patterns from SIP messages. We propose a Hidden Markov Model that also includes a Markov model associated with the observations, to represent the relationship between the signaling patterns and the observed SIP messages. The model is used to infer the specific probabilistic properties of the signaling patterns. The inference model is used by the estimation algorithm, presented in Section IV, to detect signaling patterns when all required information is known (a complete information problem), or to predict the signaling pattern when only incomplete information is available.

### A. PROBABILISTIC MODEL

Henceforth, we consider that all SIP servers and user agents in the SIP signaling path capture the SIP messages to infer the SIP statistics and also predict or detect the SIP dialogs. In this way, it is possible to predict and detect new dialogs that may constitute new vulnerabilities. The proposed inference model captures the relation between the observed SIP messages and the signaling patterns. The model is updated whenever a new SIP dialog is captured by a SIP server or an user agent, depending on where the model is used.

The modeling approach followed in this work is based on a Hidden Markov Model. The HMM is capable of modeling the probability of occurring an unobservable hidden state, $\mathbf{d}_k \in \mathcal{S}$, and the conditional probabilities of occurring observable events (observations), $\mathbf{n}_k \in \mathcal{O}$, for each hidden state. Consequently, the observable events can be used to learn about the hidden states. Next, we introduce multiple concepts required to derive the model. A table of the symbols is given in Table 1 to introduce the notation.

TABLE 1. Table of symbols.

| Symbols | Definitions |
|---|---|
| $\mathbf{A}$ | HMM Transition Matrix. |
| $\mathbf{B}$ | HMM Emission Matrix. |
| $\mathbf{d}_k$ | SIP dialog $k$. |
| $\gamma$ | Length of the state space of an $n$-gram $\mathbf{n}_k$. |
| $\Gamma$ | Zero padded sequence to define an $n$-gram. |
| $\Lambda$ | Subsequence of length $n$ obtained from $\Gamma$. |
| $\mathbf{i}$ | Observed input sequence. |
| $L_d$ | Length of a SIP dialog $\mathbf{d}_i$. |
| $L_o$ | Length of an observation. |
| $m_k$ | SIP message $k$. |
| $\mathcal{M}$ | Set with all the possible SIP messages. |
| $M$ | Number of possible SIP methods and responses. |
| $n$ | Size of the $n$-gram sliding window. |
| $\mathbf{n}_k$ | $n$-gram state space of an observation $\mathbf{o}_k$. |
| $N$ | Number of HMM Hidden States (unique SIP dialogs). |
| $\mathbf{o}_k$ | Observation $k$ (sequence of SIP messages observed in a dialog). |
| $\mathcal{O}$ | Observable state space. |
| $\Pi_k$ | Distribution of the hidden Markov chain at time $k$. |
| $\mathcal{S}$ | HMM's Hidden state space. |
| $\Theta$ | Inference dataset. |
| $\mathbf{x}$ | Viterbi Path. |

We start with the definition of a SIP message.

*Definition 1: A* ***SIP message*** $m_k$*,* $k \in \mathcal{M} = \{1, 2, \ldots, M\}$*, represents a specific type of SIP request or SIP response. The total number of SIP requests plus responses is denoted by* $M$*, and* $\mathcal{M}$ *represents the set of all types of SIP messages.*

A SIP dialog is composed by SIP messages and is defined as follows.

*Definition 2: A* ***SIP dialog*** *is a sequence of consecutive SIP messages represented by* $\mathbf{d}_k = <m^{(1)}, m^{(2)}, \ldots, m^{(L_d)}>$*, where* $m^{(j)}$ *represents the j-th message of the SIP dialog.* $L_d$ *represents the SIP dialog length. All SIP messages contained in a SIP dialog share the same SIP Call ID, and sender and receiver URIs.*

Given the number of possible SIP methods in a request and possible reply codes in a response, the number of different dialogs that can be created between the user agents is high. Besides that, the dialogs can be legitimate or anomalous.

In a SIP user agent or SIP server, the observations (inputs) are the captured SIP messages, while the unobservable data is related with the signaling patterns, i.e. the SIP dialogs (output) to detect or predict. Regarding the observations, they are defined as follows.

*Definition 3: An* ***observation*** *k taken by an user agent or a SIP server is a sequence of consecutive SIP messages represented by* $\mathbf{o}_k = <m^{(1)}, m^{(2)}, \ldots, m^{(L_o)}>$. *Each SIP message is represented by* $m^{(h)} = m_i$, $i \in \mathcal{M}$, $h \in \{1, 2, \ldots, L_o\}$. *The symbol* $L_o$ *represents the length of the observation. All SIP messages contained in an observation share the same SIP Call ID.*

The observations of length $L_o = 1$ represent the special case when each observation is exactly the last SIP message captured during the progress of a SIP dialog. However, for $L_o > 1$, we consider that all $L_o$ SIP messages are represented by an observation Markov model of $L_o - 1$ order with probability $P(X_{L_0} = m^{(L_o)} | X_1 = m^{(1)}, \ldots, X_{L_o-1} = m^{(L_o-1)})$, where the random variable $X_j$ represents the $j$-th SIP message in $\mathbf{o}_k$, $1 \leq j \leq L_o$. In this work, we adopt the $n$-gram probabilistic model [29] as the observation Markov model.

*Definition 4: A* ***n-gram*** $\mathbf{n}_k$ *associated to the observation* $\mathbf{o}_k$, *is an ordered vector of sequences of length n, obtained through a sliding window of length n over the zero padded sequence* $\Gamma = <\underbrace{0, 0, \ldots, 0}_{(n-1)}, \mathbf{o}_k, \underbrace{0, 0, \ldots, 0}_{(n-1)}>$ *of length* $2(n-1) + L_o$. *Each sequence of length n is denoted by* $\Lambda$. *The state space of the n-gram* $\mathbf{n}_k$ *is formed by all sequences* $\Lambda$ *when the sliding window is displaced 1 unit over the sequence* $\Gamma$, *resulting in a space of* $\gamma = (n-1)+L_o$ *sequences. Finally, the n-gram is written as* $\mathbf{n}_k = <\Lambda^{(1)}, \Lambda^{(2)}, \ldots, \Lambda^{(\gamma)}>$.

The representation of an observation $\mathbf{o}_k$ as an ordered vector of $\gamma$ sequences $\Lambda$, i.e., the adoption of the n-gram model allows the association of a probability for each of the $\gamma$ sequences observed in a consecutive way. Consequently, each displacement of the sliding window over $\Gamma$ is conditioned on prior $n-1$ elements. Additionally, because $n-1$ consecutive zero elements are considered in the head and tail of $\Gamma$, an observation $\mathbf{o}_a$ contained in a longer sequence $\mathbf{o}_b$ can be univocally represented by a $n$-gram $\mathbf{n}_a$, allowing to distinguish shorter observations included in longer ones.

*Definition 5: The* ***observation space****, denoted by* $\mathcal{O}$, *is formed by the set of n-grams of the permutations (with repetition) of the* $L_o$ *SIP messages integrating each observation* $\mathbf{o}_k$. *Since the permutations sum up* $M^{L_o}$, *i.e.,* $\mathcal{O} = \{\mathbf{n}_1, \ldots, \mathbf{n}_k\}$, $k = M^{L_o}$, *and each* $\mathbf{n_k}$ *n-gram contains* $\gamma$ *sequences* $\Lambda$, *the maximum number of sequences* $\Lambda$ *in* $\mathcal{O}$ *is* $\gamma M^{Lo}$, *i.e.* $\mathcal{O} = \{<\Lambda_1^{(1)}, \Lambda_1^{(2)}, \ldots, \Lambda_1^{(\gamma)}>, \ldots, <\Lambda_k^{(1)}, \Lambda_k^{(2)}, \ldots, \Lambda_k^{(\gamma)}>,$ $k = M^{L_o}$.

Regarding the dimension of the observation space, it is noteworthy to mention the importance of the $n$-gram length ($n$). As mentioned before, the state space of the $n$-gram $\mathbf{n}_k$ is formed by all sequences $\Lambda$, resulting in the space formed by $\gamma = (n-1) + L_o$ sequences. Consequently, the dimension of the observation space, given by $\gamma M^{Lo}$, increases with $n$, showing the influence of the $n$-gram length on the computational costs associated with operations over the observation space.

The definitions of the hidden states, the HMM, as well as the HMM transition matrix, emission matrix, and distribution are next introduced.

*Definition 6: The* ***hidden states space*** *is represented by* $\mathcal{S} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$, *where N stands for the total number of unique SIP dialogs (hidden states).*

*Definition 7: The pair of discrete-time stochastic processes represented by the random variables* $(S_k, O_k)$, $S_k \in \mathcal{S}$, $O_k \in \mathcal{O}$, *where* $k \geq 1$ *represents the discrete-time index, is a* ***hidden Markov model****, if* $S_k$ *represents the non-observable (hidden) Markov process, and the conditional probability* $P(O_k \in \mathcal{O} | S_1, \ldots, S_k) = P(O_k \in \mathcal{O} | S_k = \mathbf{d}_k)$ *is defined over the observable state space* $\mathcal{O}$.

*Definition 8: The* ***transition matrix*** *of the hidden Markov chain is denoted by* $\mathbf{A}$, *defined by the probabilities* $a_{i,j} = P(S_{k+1} = \mathbf{d}_j | S_k = \mathbf{d}_i)$, *where* $S_k$ *is a random variable describing the hidden state at discrete time k, and* $1 \leq i \leq N$, $1 \leq j \leq N$.

*Definition 9: The* ***emission matrix*** *of the hidden Markov chain is denoted by* $\mathbf{B}$, *defined by the probabilities* $b_{i,j} = P(O_k = \Lambda_j | S_k = \mathbf{d}_i)$, *with* $1 \leq i \leq N$, $1 \leq j \leq \sum_{k=1}^{L_{d_{max}}} [(n-1) + k] M^k$, *and* $L_{d_{max}}$ *represents the maximum length of a SIP dialog.*

Next, we define the distribution of the hidden Markov chain at a given discrete time.

*Definition 10: The* ***distribution of the hidden Markov chain*** *at the discrete time instant k is represented by* $\Pi_k = \{\pi_1, \ldots, \pi_N\}$, *where* $\{\pi_1, \ldots, \pi_N\}$ *represents the probabilities that* $\{P(S_k = \mathbf{d}_1), \ldots, P(S_k = \mathbf{d}_N)\}$, *with* $\sum_{l=1}^{N} P(S_k = \mathbf{d}_l) = 1$.

### B. IMPLEMENTATION OF THE INFERENCE MODEL

The inference model is computed for each SIP dialog contained in a dataset $\Theta = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\}$, $m \geq N$, of legitimate SIP dialogs. The following steps are required to compute the inference model:

1) If a SIP dialog $\mathbf{d}_k \in \Theta$ has never been inferred before it is included in the hidden states space $\mathcal{S}$;
2) Each SIP dialog $\mathbf{d}_k \in \Theta$ is seen as an observation and the corresponding $n$-gram $\mathbf{n}_k$ is computed according to Definition 4. The $n$-gram identifies the sequence $\Lambda^{(1)}, \ldots, \Lambda^{(\gamma)}$ and, for each $\Lambda$ a corresponding counter is incremented. The counters are then used to update the probability values in the emission matrix $\mathbf{B}$, to describe how likely is to observe the $n$-gram $\mathbf{n}_k$ when a SIP dialog $\mathbf{d}_k$ occurs;

3) The dataset is run from $\mathbf{d}_1$ to $\mathbf{d}_m$ to compute the probabilities $a_{i,j}$ of the transition matrix $\mathbf{A}$.

The dataset is used to identify the unique dialogs, which also represent the number of HMM hidden states, i.e., $N$. The SIP standard (RFC) [14] is used to identify the number of observable SIP messages, i.e. $M$. The different types of SIP request and SIP responses are univocally translated to integer values. To compute the transition matrix $\mathbf{A}$ we have considered that a transition occurs whenever a SIP message is transmitted, so there will be $L_d - 1$ transitions per SIP dialog. Regarding the emission matrix, it is computed by counting the number of occurrences of each sequence $\Lambda$ computed for each SIP dialog.

We highlight that $\mathbf{B}$ is a sparse matrix. Admitting that the longest SIP dialog in the dataset has length $L_{d_{max}}$, the maximum dimension of matrix $\mathbf{B}$ would be $N \times \sum_{k=1}^{L_{d_{max}}} [(n-1)+k]M^k$, as indicated in Definition 9. This means that the maximum number of the matrix columns must take into account the longest SIP dialogs, i.e., the ones containing the highest number of SIP messages. However, the great majority of SIP dialogs are shorter, i.e., they are formed by less than $L_{d_{max}}$ SIP messages. For a specific dialog $\mathbf{d}_k$ of length $L_d < L_{d_{max}}$, the number of non-null probabilities to update in $\mathbf{B}$ is only $(n-1)+L_d$, because the specific dialog is formed by less SIP messages than the longest one and contains a single combination of $L_d$ SIP messages. Consequently, while the maximum number of columns of $\mathbf{B}$ is $\sum_{k=1}^{L_{d_{max}}} [(n-1)+k]M^k$, for a specific SIP dialog of length $L_d$ only $(n-1)+L_d$ columns need to be updated. This fact shows that the implementation of $\mathbf{B}$ can make use of efficient data structures where only $(n-1)+L_d$ elements need to be stored for each of the $N$ hidden states.

## IV. PREDICTION OF SIP DIALOGS

In this section we describe how to predict or detect the most likely SIP dialog. The computation of the prediction and detection algorithm is based on an input sequence of SIP messages, as described in Definition 11.

*Definition 11: The* ***observed input sequence*** *is represented by the observation* $\mathbf{i} = <m^{(1)}, \ldots, m^{(L_o)}>$, $L_o \leq L_d$.

The prediction algorithm is based on the well known Viterbi algorithm [30] adapted to the $n$-gram observation model. The algorithm contains two phases: the forward and the backward stages, which are presented in Algorithm 1. The algorithm uses the transition matrix, $\mathbf{A}$, the emission matrix, $\mathbf{B}$, and the space of hidden states, $\mathcal{S}$, computed in the inference stage. The distribution of the hidden Markov chain, $\Pi_0$, is the inferred HMM distribution when the algorithm is run.

In line 1 the $\gamma$ sequences $\Lambda$ are built considering the input vector $\mathbf{i}$, so the $n$-gram model is applied to the observed input sequence. The "find" function in lines 2 and 9 finds the index of a particular sequence $\Lambda$ contained in $\mathcal{O}$. In the forward stage (lines 2 to 11) the probability of each hidden state, $\mathbf{T}_1$, is computed for each sequence $\Lambda_{(j)}$, $1 \leq j \leq \gamma$.

**Algorithm 1** Prediction Algorithm

```
    Input: $\mathbf{i}$, $\Pi_0$, $\mathcal{S}$, $\mathbf{A}$, $\mathbf{B}$
    Output: $\mathbf{x} =< \mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(\gamma)} >$          ▷ Viterbi Path
 1: $\mathbf{n}_k$ = build_$n$-gram_sequences ($\mathbf{i}$)          ▷ build $\mathbf{n}_k$ for $\mathbf{i}$
 2: $k$ = find($\mathbf{n}_k[1]$, $\mathcal{O}$)          ▷ find $\Lambda^{(1)}$ index in $\mathcal{O}$
 3:          ▷ Forward Stage:
 4: for each state $i = 1, 2, \ldots, N$ do
 5:     $\mathbf{T}_1[i, 1] \leftarrow \Pi_0(i) \cdot b_{i,k}$
 6:     $\mathbf{T}_2[i, 1] \leftarrow 0$
 7: end for
 8: for each $\Lambda^{(j)}$ sequence in $\mathbf{n}_k$, $j = 2, \ldots, \gamma$ do
 9:     $k$ = find ($\mathbf{n}_k[j]$, $\mathcal{O}$)
10:     for each state $i = 1, 2, \ldots, N$ do
11:         $\mathbf{T}_1[i, j] \leftarrow \max_{1 \leq s \leq N} (\mathbf{T}_1[s, j-1] \cdot a_{s,i} \cdot b_{i,k})$
12:         $\mathbf{T}_2[i, j] \leftarrow \arg\max_{1 \leq s \leq N} (\mathbf{T}_1[s, j-1] \cdot a_{s,i} \cdot b_{i,k})$
13:     end for
14: end for
15: if $\mathbf{T}_1[:, \gamma] == 0$ then return $\mathbf{x} =<>$
16: end if
17:          ▷ Backward Stage:
18: $s_\gamma \leftarrow \arg\max_{1 \leq s \leq N} (\mathbf{T}_1[s, \gamma])$
19: $\mathbf{d}^{(\gamma)} \leftarrow \mathcal{S}[s_\gamma]$
20: for $j = \gamma, \ldots, 2$ do
21:     $s_{j-1} \leftarrow \arg\max_{1 \leq s \leq N} (\mathbf{T}_2[s_j, j])$
22:     $\mathbf{d}^{(j-1)} \leftarrow \mathcal{S}[s_{j-1}]$
23: end for
```

The computations form a Viterbi Trellis represented in Figure 1, where each observable input is represented by a square and each hidden state by a circle. Having computed the Viterbi Trellis, the algorithm computes in the backward stage the optimal sequence of hidden states $\mathbf{x} = <\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(\gamma)}>$, defined as the Viterbi path as follows.

*Definition 12: The* ***Viterbi path****,* $\mathbf{x} = <\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(\gamma)}>$, *is the most likely sequence of predicted hidden states for the observed input sequence* $\mathbf{i}$.

The two stages of the algorithm are equivalent to compute the shortest path for the observed input sequence based on the inferred transition, emission, and initial distribution matrices. In some cases, the Viterbi path might be empty (condition in line 15 of the algorithm). In this case the algorithm is unable to compute the complete path because the observed input messages and/or the hidden states transition have null probabilities.

Given that the output of the estimation algorithm has dimension $\gamma$, i.e., it can contain multiple SIP dialogs, we propose two different methods to compute the estimated SIP dialog. A SIP dialog might be predicted according to the criteria presented in definitions 13 or 14.

*Definition 13 (**Equal-Dialog Criterion (EdC)**): Given the Viterbi path* $\mathbf{x} = <\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(\gamma)}>$, *the SIP dialog is only predicted when all dialogs* $\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(\gamma)}$ *in* $\mathbf{x}$ *are equal. Otherwise, no SIP dialog is predicted.*

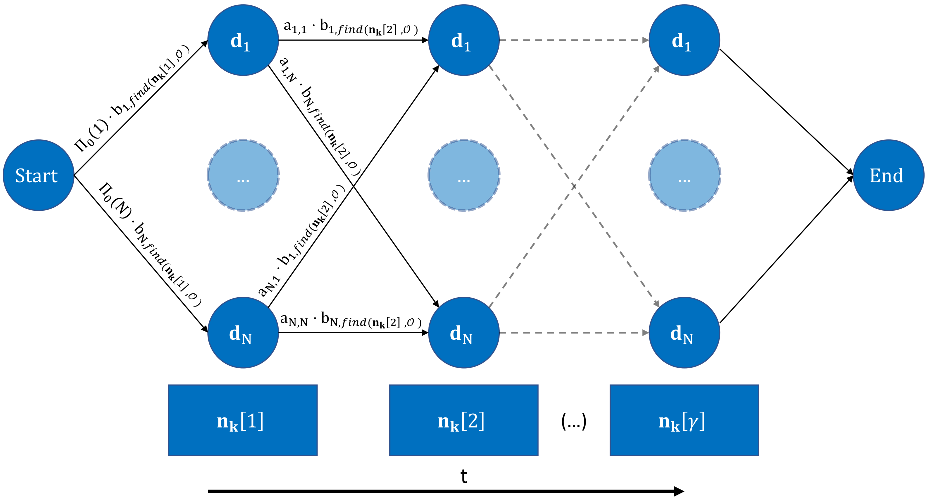

Start
$\Pi_0(1) \cdot b_{1,find(\mathbf{n_k}[1], \mathcal{O})}$
$\Pi_0(N) \cdot b_{N,find(\mathbf{n_k}[1], \mathcal{O})}$
$\mathbf{d}_1$
...
$\mathbf{d}_N$
$a_{1,1} \cdot b_{1,find(\mathbf{n_k}[2], \mathcal{O})}$
$a_{1,N} \cdot b_{N,find(\mathbf{n_k}[2], \mathcal{O})}$
$a_{N,1} \cdot b_{1,find(\mathbf{n_k}[2], \mathcal{O})}$
$a_{N,N} \cdot b_{N,find(\mathbf{n_k}[2], \mathcal{O})}$
$\mathbf{d}_1$
...
$\mathbf{d}_N$
$\mathbf{d}_1$
...
$\mathbf{d}_N$
End
$\mathbf{n_k}[1]$
$\mathbf{n_k}[2]$
(...)
$\mathbf{n_k}[\gamma]$
t

**FIGURE 1.** Viterbi trellis.

*Definition 14* ***Most Frequent Dialog Criterion (MFdC):*** *Given the Viterbi path* $\mathbf{x} = <\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(\gamma)}>$*, the predicted SIP dialog is the most frequent dialog in* $\mathbf{x}$*. In case of a tie no SIP dialog is predicted.*

When the observed input sequence $\mathbf{i}$ has length $L_o < L_d$, only a part of the SIP dialog is observed and, consequently, the algorithm predicts the SIP dialog. In the special case when $L_o = L_d$, the algorithm acts as a detector of the SIP dialog.

One disadvantage of the Viterbi algorithm is its computational complexity, a topic explored in [31], [32]. In our case the Viterbi algorithm runs in time $O(\gamma M^{L_o} N^2)$. Although $M$, the number of different SIP messages, is low, and the same can hold true for $L_o$ and $\gamma$, the number of different dialogs, $N$, is usually high. However, if the emission or transition matrices are sparse, i.e., less than 50% of its elements are non-zero values, the computation time can be reduced. Algorithm 1 is computed for the $N$ unique SIP dialogs observed during the inference stage. For a specific dialog $\mathbf{d}_k$, only $\gamma = (n-1)+L_o$ computations are required to update $\mathbf{T_1}$ and $\mathbf{T_2}$, instead of $\gamma M^{L_o}$ in the Viterbi algorithm. This is because no computations are required for the entire observation state space $\mathcal{O}$, but only for the elements of the space found in the observed input sequence $\mathbf{i}$. Consequently, the computational complexity of the proposed algorithm is $O(\gamma N^2)$, representing a significant computational gain.

## V. PERFORMANCE EVALUATION

This section characterizes the performance of the detection and prediction algorithms. The capability of detecting unknown SIP dialogs is also validated.

### A. EXPERIMENTAL METHODOLOGY AND DATASETS

The purpose of the experimental methodology followed in the next subsections is to assess the capacity of classifying an observed and already inferred dialog in the correct SIP dialog (**objective (a)**), but also assess the capacity of detecting SIP dialogs not inferred so far (**objective (b)**), denominated as **unknown SIP dialogs**.

Both objectives are important from the point of view of hypothetical attacks. Objective (a) can be used **to classify dialogs that have been inferred so far** and are already labeled as safe, abnormal, or according to different vulnerability rank labels. In this case, it is important to classify the observed messages in the correct SIP dialog, so that harmful SIP dialogs can be effectively recognized. Consequently, in the experimental methodology, the probability of detection and prediction is only computed for the SIP dialogs inferred so far. We highlight that this goal is mainly related to the fact that the SIP dialogs inferred so far can be mis-detected due to the different lengths of the SIP dialogs because a longer dialog can contain a shorter one and without using the $n$-gram technique the Viterbi algorithm was only capable of detecting the shorter one or the longer one (not both). Regarding objective (b) it is particularly important **to detect unknown SIP dialogs not inferred so far** because it can represent a new type of attack. While in objective (a) the outcome of the detection is always a **non-null Viterbi path**, indicating a correct or incorrect SIP dialog, in objective (b) the goal is to validate the detection of unknown SIP dialogs not inferred so far through the **null Viterbi path** outcome. To this end, the experimental methodology is conducted in the following order:

- In Subsection V-B we analyze the $n$-gram performance to distinguish shorter SIP dialogs that can be contained in longer ones. Particularly, we characterize the influence of the parameter $n$ to classify the SIP dialogs inferred so far and on in its computation time;
- Subsection V-C addresses objective (a) by characterizing the classification performance of the SIP dialogs already inferred so far;
- Finally, Subsection V-D addresses objective (b) by showing the capability of detecting unknown SIP dialogs.

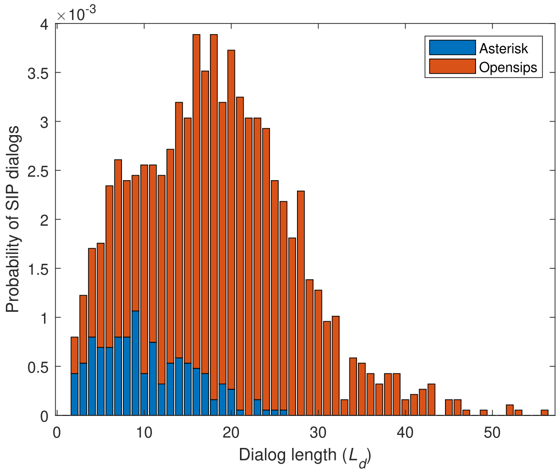


**FIGURE 2.** Histogram of the length of the unique SIP dialogs.

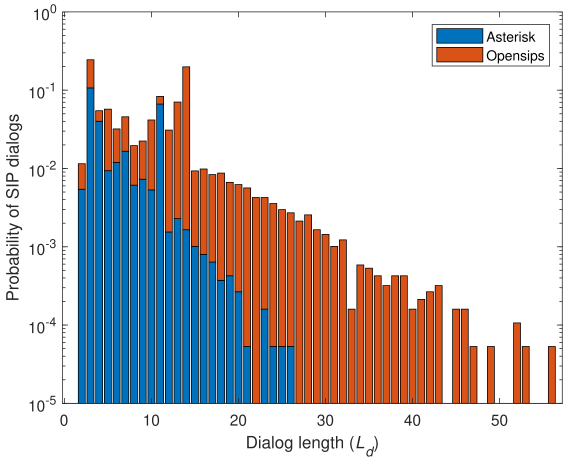


**FIGURE 3.** Histogram of the length of all SIP dialogs in the dataset.

Next we characterize the datasets used in the experiments. The performance evaluation is based on the SIP dataset created by Nassar *et al.* [13]. The dataset uses two types of open source SIP servers: Asterisk [33] and Opensips [34]. They include information about the URI of both sender and receiver user agent, the messages sent, the number of packets, the timestamp of each SIP dialog, and the final SIP session state, i.e., if the call was successful, rejected, or canceled. To identify each SIP dialog in the dataset we have used information about the URI user agent, the SIP messages sent, and the timestamp of each SIP dialog. The different types of SIP messages were encoded as integer values. In what follows we denominate this dataset as the **non-anomalous dataset**. The non-anomalous dataset contains a total of 18782 SIP dialogs established between a group of 249 user agents, which corresponds to 1492 unique SIP dialogs, i.e. the number of hidden states $N$. The majority of the 1492 unique SIP dialogs, 66.23%, only occur once, which may result in lower prediction performance due to its low occurrence probability.

A histogram of the SIP dialog lengths is plotted in Figure 2 for the unique 1492 SIP dialogs. As can be seen the longest SIP dialog is formed by 56 SIP messages. There is a higher concentration of SIP dialogs with length between 3 and 30, and longer SIP dialogs are less likely to occur. A higher number of dialogs with length 15, 17, and 19, are observed.

While the results in Figure 2 report the histogram obtained for the 1492 different types of dialogs, in Figure 3 we characterize the histogram of the SIP dialog lengths considering the 18782 SIP dialogs included in the non-anomalous dataset, which contain multiple occurrences of the unique SIP dialogs. The results in Figure 3 show two peaks for the dialogs with length 3 and 14 and indicate that the distribution of the dialogs in the entire dataset is not so concentrated for shorter lengths as it is when only unique SIP dialogs are considered. However, both have a huge concentration of dialogs with length 3.

Nassar *et al.* [13] also made available another dataset containing 152 unique SIP dialogs representing attacks, which we denominate as the **anomalous dataset**. We highlight that none of the SIP dialogs contained in the anomalous dataset is contained in the non-anomalous dataset.

The non-anomalous dataset was divided into two different datasets, **the inference dataset** and **the test dataset**. The test dataset was formed by the last 20% of each user agent's SIP dialogs while the inference dataset contained the remaining 80%. The computation of **A**, **B**, and $\Pi$, was based on the dialogs' distribution of the inference dataset. In Subsection V-D the results reported with the test dataset have followed a cross-validation k-fold methodology, by creating 5 different inference and test datasets over the non-anomalous dataset.

Finally, the inference model and the prediction algorithm was implemented in Matlab running in a 64bit Windows 10 OS system over an Intel Core(TM) i5-5200U CPU @ 2.20GHz with 8 GB of RAM and a GeForce 840M GPU.

### B. IMPACT OF THE n-GRAM LENGTH

In this subsection we evaluate the impact of the $n-$gram sequences length ($n$) in the detection performance and detection computation times.

Regarding the experimental methodology, in the evaluation process we vary $n$ from 1 to 56, which corresponds to the range of the SIP dialogs length contained in the dataset. The HMM model was inferred for each sequence length value using the inference dataset. Then, the SIP dialogs contained in both inference and test datasets were detected by considering each SIP dialog in the dataset as an observed input sequence, **i**, in Algorithm 1. In this way, we evaluate the capacity of detecting the different SIP dialogs of both datasets for different $n-$gram sequences length.

The detection results are plotted in Figure 4 for the different EdC/MFdC detection criteria. The detection of the SIP dialogs increases with $n$ because the Markov order of the observations increases and the likelihood of distinguishing shorter dialogs included in longer dialogs also increases. The results show that the detection rate of the SIP dialogs in the

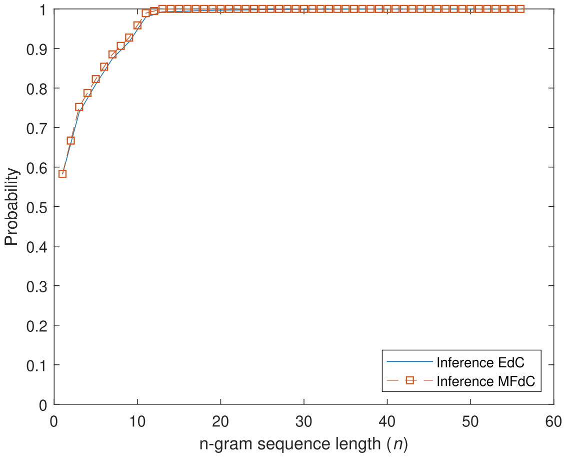


**FIGURE 4.** **Detection performance for different $n$-gram sequence lengths ($n$).**

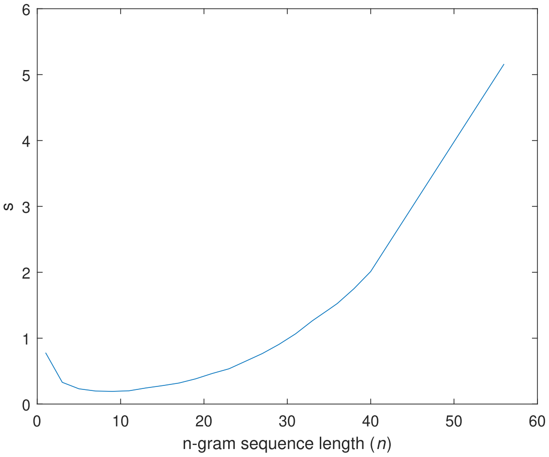


**FIGURE 5.** **Average detection computation time per SIP dialog for different $n$-gram emission matrices.**

inference dataset is 100% for $n = 13$, which is an important value because it indicates that the proposed algorithm is able to detect all types of SIP dialogs using the inference dataset, even the shortest ones included in longer SIP dialogs. The detection results are slightly higher for the MFdC (the EdC achieves 0.9927, and the MFdC achieves 1.0000).

The choice of the $n$-gram sequence length, $n$, cannot be made only taking into consideration the detection performance, because the $n$-gram sequence length influences the computational complexity of the detection algorithm. Figure 5 presents the average computation time to detect each SIP dialog for the different values of $n$. Longer $n$-gram sequences increase the observation space, resulting in higher average computation times. However, for smaller $n$ values the computation time also increases due to the fact that more observation states are returned by the function “find” in the algorithm, resulting in more non-null probabilities of the emission and transition matrices that need to be considered in the computation process. From the results in Figures 4 and 5, we have parameterized the length of the $n$-gram sequences to 13, since it represents the best trade-off between the detection performance and computation time (approximately 243 ms per SIP dialog). All the experimental results presented next assumed $n = 13$.

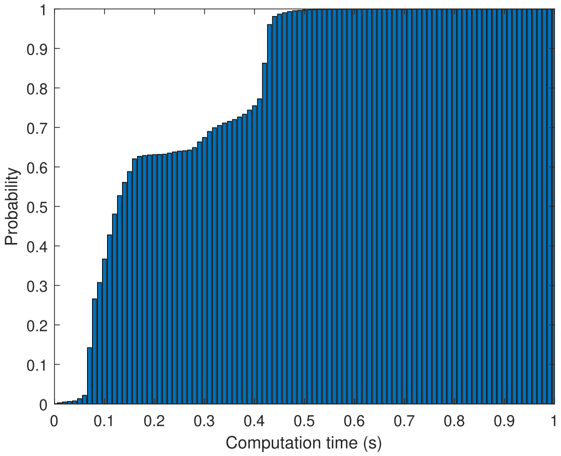


**FIGURE 6.** **CDF of the detection times of the non-anomalous dataset ($n = 13$).**

### C. DETECTION AND PREDICTION PERFORMANCE

The inference model is based on counters that register the number of transitions between hidden states and the number of different types of SIP messages forming each SIP dialog. Consequently, the inference procedure is updated as the information flows throughout each server, being computed in real-time, which explains why the inference *per se* is not critical for real-time prediction. We emphasize that the most critical aspect to run the proposed methodology in real-time is Algorithm 1. To show the potential of the proposed solution in terms of its real-time computation, we characterized the computational time required to run the detection algorithm. To this end we have run the detection algorithm for all SIP dialogs included in the non-anomalous inference dataset. The cumulative distribution function of the computation time is presented in Fig. 6 for the non-anomalous dataset (similar results were obtained for the anomalous dataset). As plotted in Figure 6, approximately 65% of the SIP dialogs are computed in less than 200 ms, showing the feasibility of the detection algorithm in real-time.

The detection performance of EdC and MFdC was evaluated for different sets of input sequences. Three new datasets were defined with SIP dialogs copied from the inference set: the dataset $K_{most}$ includes the 150 most likely unique dialogs; the dataset $K_{less}$ includes the 150 less probable unique dialogs; and, the set $K_{random}$ includes 150 SIP dialogs randomly chosen (in this case 100 random trials were considered). The detection performance obtained for the different datasets is presented in Figure 7, showing the comparative performance of both criteria for the less and the most probable SIP dialogs of the inference dataset. The results show that higher performance can be achieved with the MFdC. The performance gap is bigger for less probable SIP dialogs ($K_{less}$ set), where the EdC achieved 96% of detection probability

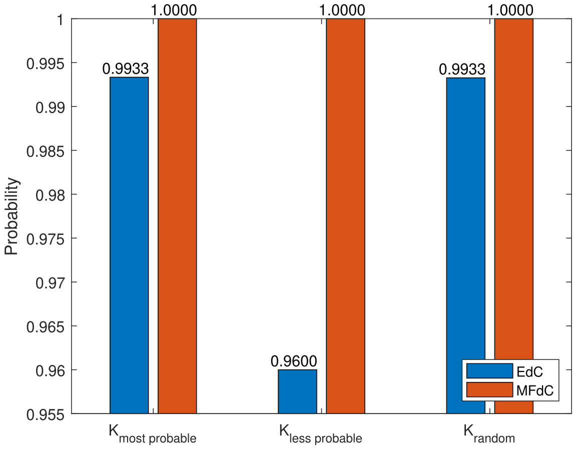


**FIGURE 7.** Detection probability for the $K_{most}$, $K_{less}$, and $K_{random}$ observed input sequence sets.

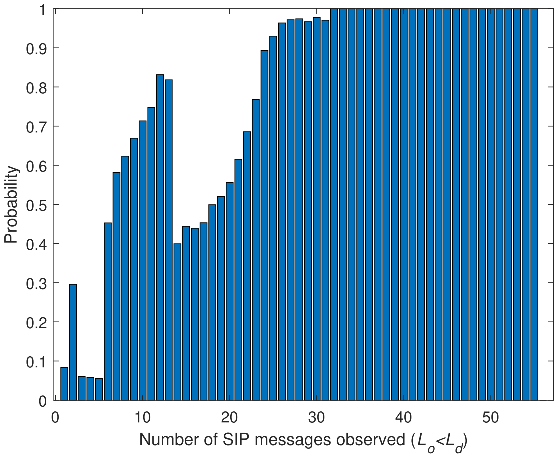


**FIGURE 8.** Prediction probability of SIP dialogs for different lengths of the observed input sequence (MFdC).

because the dialogs are less represented in the inference dataset. Finally, the results obtained for the $K_{random}$ set are similar to the ones in Figure 4.

Next, we characterize the prediction of all SIP messages contained in the inference dataset. In the experimental methodology, we have considered that $L_d$ observed input sequences, $\mathbf{i}$, are created from each SIP dialog in the inference dataset. This means that for each dialog of length $L_d$ we obtain $L_d$ observed input sequences as follows, $\mathbf{i}_1 = <m^{(1)}>$, $\mathbf{i}_2 = <m^{(1)}, m^{(2)}>, \ldots, \mathbf{i}_{L_d} = <m^{(1)}, m^{(2)}, \ldots, m^{(L_d)}>$, which represents a real-time interaction between two users as the SIP messages are sent over time. To evaluate the number of SIP messages needed to correctly predict the right SIP dialog the Viterbi Path, $\mathbf{x}$, was computed for the 132855 $\mathbf{i}$ input sequences obtained from the 18782 dialogs of the inference dataset.

Figure 8 presents the prediction probability per length of the observed input sequence ($L_o$). The prediction probability was computed taking into consideration the number of observed input sequences for which the SIP dialogs were successfully predicted over the number of all input sequences of equal length (the results were obtained with the MFdC, but similar results were observed for the EdC). The results show that the prediction probability increases with the length of the observed input sequence, because the similarity between the input sequence and known dialogs increases for longer observed SIP sequences. However, this justification is not entirely observed in the figure because the distribution of the SIP dialogs' length presented in Figure 3 must also be taken into consideration. Based on the results in the figure, the prediction performance can be analyzed for three different regions: $L_o \leq 5$, $5 < L_o < 14$, and $L_o \geq 14$. In the first region the prediction performance is low but for $L_o = 2$ there is an increase of dialogs correctly predicted. The result indicates that most of the dialogs with $L_d = 3$ only need 2 SIP messages to be correctly estimated. One reason that supports this conclusion is the high occurrence of these dialogs (see Figure 3) and the low number of unique dialogs with that length (see Figure 2). In the second region the prediction probability increases with $L_o$, and for $L_o = 13$ the prediction probability reaches 0.8182. However, in the beginning of the third region the probability decreases due to the higher occurrence of the dialogs with length $6 \leq L_d \leq 14$, in comparison with the dialogs with length above 14 (see Figure 3). Simultaneously, the number of unique dialogs reaches the highest number when its length is above 14 (see Figure 2). Finally, during the third region the probability of correctly predicting a dialog increases again with the length of the observed input sequence, achieving 1 when the input sequence is longer than 32.

While the previous results considered the prediction rate according to the length of the observed input sequence, next we characterize the prediction rate according to the amount of information available to predict each dialog. Specifically, we consider the ratio between the observed input sequence length used in the prediction probability and the SIP dialog's length, i.e., $L_o/L_d$. The experiment has adopted the same dataset used to obtain the results in Figure 8, but the prediction algorithm was computed considering a single observation given by the first SIP message in the SIP dialog, two observations given by the first two SIP messages in the SIP dialog, and so on, until reaching the $L_d$ SIP messages in the SIP dialog.

The prediction results are presented in Figure 9 for the MFdC and similar results are obtained for the EdC. The ratio between the observed input sequence length over the SIP dialog's length is represented in percentage in the x-axis. For $L_o/L_d < 8.929\%$ the prediction algorithm misdetects all the dialogs by classifying them in a wrong type. In this case the amount of observed information is not enough to predict any SIP dialog. For $L_o/L_d \geq 8.929\%$ the prediction probability of SIP dialogs increases with $L_o/L_d$, because more information is observed and used in the prediction algorithm. A maximum prediction percentage of 87.16% is achieved for $L_o/L_d$= 98.21% and the detection probability, i.e., for $L_o/L_d = 100\%$, achieves 100%.

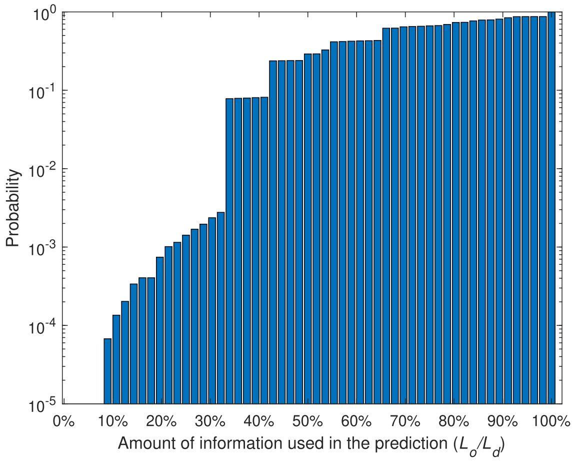


**FIGURE 9.** SIP dialogs prediction probability over the amount of available information (MFdC).

### *D. DETECTION OF UNKNOWN SIP DIALOGS*

In this subsection, we evaluate the performance of the inference and detection scheme to detect unknown SIP dialogs, which were inferred before. We highlight that unknown SIP dialogs are detected whenever the computed Viterbi path is null.

In the first experience, we have used the test dataset using the k-fold cross-validation methodology described in Subsection V-A. After running the inference model to learn the inference dataset we have run the detection algorithm for each unique SIP dialog of the test dataset using the MFdC criterion. The detection algorithm has successfully classified 99.942% of the inferred SIP dialogs and misclassified 0.058%. The detection probability is in line with the detection probability reported in Figure 9 for the inference dataset only. Moreover, the computed Viterbi path was null for all SIP dialogs that have not be inferred, i.e., for the unknown SIP dialogs contained in the test datasets. This result confirms the possibility of detecting all SIP dialogs not contained in the inference datasets, which is a crucial feature to detect new types of attacks.

Next, we highlight the advantage of online learning. We consider the data contained in the anomalous dataset, which contains 152 unique SIP dialogs not included in the inference dataset. Initially, the emission and transition matrices are updated according to the data contained in the non-anomalous dataset. To evaluate the online learning we randomly select SIP dialogs contained in the anomalous dataset. Each dialog is then used in the inference model described in Section III.B and the emission and transition matrices are updated accordingly. Then the dialog is detected using the algorithm proposed in Section IV. The goal is to determine how many times ($\chi$) each unknown SIP dialog needs to be updated in the learning algorithm in Subsection III.B before it can be successfully detected. In such a way we characterize the benefit of adopting online learning and its efficiency according to the required amount of prior learning updates. The results in Table 2 characterize the online detection performance.

The first row of Table 2 indicates that approximately 96.6592% of the unknown SIP dialogs are successfully detected after having been updated once when the EdC detection criterion is adopted. As can be seen for the EdC criterion there are a few SIP dialogs that are not successfully detected after being updated once, but they represent less than 4% of all unknown dialogs. However, we show that the MFdC criterion achieves higher performance, as it successfully detects all unknown SIP dialogs after they have been updated once. The results in Table 2 show that the proposed learning and detection algorithms are capable of achieving high successful detection rates even when the transmission and emission matrices are only updated once, meaning that a SIP dialog can be successfully detected after its first inference when the MFdC criterion is adopted.

**TABLE 2.** Detection performance of the unknown SIP dialogs after being inferred $\chi$ times before a successful detection.

| $\chi$ | EdC | MFdC |
|---|---|---|
| 1 | 96.6592% | 100.0000% |
| 2 | 1.7817% | - |
| 3 | 0.6682% | - |
| 4 | 0.2227% | - |
| 9 | 0.2227% | - |
| 12 | 0.2227% | - |
| 14 | 0.2227% | - |

Departing from the fact that the inference dataset can be updated over time, and so the transition and emission matrices, the detection of an unknown SIP dialog has to do with the correct rejection of that dialog, since it was never inferred before. Since none of the unknown SIP dialogs is contained in the inference dataset, the correct rejection performance of Algorithm 1 is enough to assess its capability of identifying SIP dialogs never observed before. In this way, the SIP dialogs can be classified as possible vulnerabilities or eventual dialogs that should be confirmed by domain experts before being included in the inference dataset.

To evaluate the capacity of correct rejection, we have used the inference dataset to obtain the transition and emission matrices. Then the Algorithm 1 was run for all SIP dialogs in the anomalous dataset. Each unknown SIP dialog was used as the observed input sequence adopted in the algorithm to evaluate if it was correctly rejected. In this experiment, we have observed that the returned Viterbi path was always empty, meaning that all anomalous SIP dialogs were correctly rejected. This indicates the capacity to correctly reject all anomalous SIP messages contained in the anomalous dataset, indicating a correct rejection rate of 100%. Consequently, the proposed methodology can be effectively used to detect anomalous or unknown (non-inferred) SIP dialogs, attesting the capability of the proposed algorithms to identify possible vulnerable SIP dialogs never inferred before.

## VI. CONCLUSION

This work has proposed a machine learning technique to predict SIP dialogs as SIP signaling messages are observed over time. An inference model was proposed to associate the already known SIP signaling patterns to the observed SIP messages. The inference is used by a prediction algorithm to compute the SIP signaling pattern given the set of observed SIP messages. The algorithm is applied to a subset of the state space, so we can address data sparsity efficiently. Experimental results evaluate the detection and prediction performance of the proposed methodology. We also present results obtained for a dataset of anomalous SIP dialogs, not included in the inference dataset, showing that the proposed methodology can accurately detect all the abnormal SIP sequences in a short period of time. The experimental results demonstrate the performance of the proposed solution, showing its effectiveness in classifying anomalous SIP dialogs.

The proposed prediction methodology can be enriched through the use of additional information contained in the SIP packets of a specific dialog. The influence of the adoption of more information in the performance of the prediction performance and in the computational time will be further investigated. As future work, we also plan to assess the prediction performance and computation complexity when adopting other prediction techniques, such as deep learning.

## REFERENCES

[1] A. Uzelac and Y. Lee, *Voice Over IP (VoIP) SIP Peering Use Cases*, RFC, document 6405, RFC Editor, Nov. 2011.

[2] 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.

[3] Y. Rebahi, M. Sher, and T. Magedanz, "Detecting flooding attacks against IP multimedia subsystem (IMS) networks," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl.*, Mar. 2008, pp. 848–851.

[4] 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.

[5] 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. 2, pp. 225–243, Mar. 2010.

[6] D. Geneiatakis, T. Dagiuklas, G. Kambourakis, C. Lambrinoudakis, S. Gritzalis, K. 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.

[7] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019.

[8] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[9] N. Hentehzadeh, A. Mehta, V. K. Gurbani, L. Gupta, T. Kam 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.

[10] R. Ferdous, R. L. Cigno, and A. Zorat, "On the use of SVMs to detect anomalies in a stream of SIP messages," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, Dec. 2012, pp. 592–597.

[11] J. Tang, Y. Cheng, Y. Hao, and W. Song, "SIP flooding attack detection with a multi-dimensional sketch design," *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 6, pp. 582–595, Nov. 2014.

[12] 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.

[13] M. Nassar, R. State, and O. Festor, "Labeled VoIP data-set for intrusion detection evaluation," in *Proc. 16th EUNICE/IFIP WG 6.6 Conf. Netw. Services Appl., Eng., Control Manage. (EUNICE)*. Berlin, Germany: Springer-Verlag, 2010, pp. 97–106.

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

[15] 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.

[16] D. Seo, H. Lee, and E. Nuwere, "SIPAD: SIP–VoIP anomaly detection using a stateful rule tree," *Comput. Commun.*, vol. 36, no. 5, pp. 562–574, Mar. 2013.

[17] 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.

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

[19] B. B. Gupta and V. Prajapati, "Secure and efficient session initiation protocol authentication scheme for VoIP communications," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Jul. 2019, pp. 866–871.

[20] 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.

[21] A. Irshad, M. Sher, M. S. Faisal, A. Ghani, M. Ul Hassan, and S. Ashraf Ch, "A secure authentication scheme for session initiation protocol by using ECC on the basis of the tang and Liu scheme," *Secur. Commun. Netw.*, vol. 7, no. 8, pp. 1210–1218, Aug. 2014.

[22] Y. Zhang, K. Xie, and O. Ruan, "An improved and efficient mutual authentication scheme for session initiation protocol," *PLoS ONE*, vol. 14, no. 3, Mar. 2019, Art. no. e0213688.

[23] D. He, J. Chen, and Y. Chen, "A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography," *Secur. Commun. Netw.*, vol. 5, no. 12, pp. 1423–1429, Dec. 2012.

[24] S. H. Islam, P. Vijayakumar, M. Z. A. Bhuiyan, R. Amin, M. V. Rajeev, and B. Balusamy, "A provably secure three-factor session initiation protocol for multimedia big data communications," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3408–3418, Oct. 2018.

[25] X.-Y. Guo, D.-Z. Sun, and Y. Yang, "An improved three-factor session initiation protocol using chebyshev chaotic map," *IEEE Access*, vol. 8, pp. 111265–111277, 2020.

[26] 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.

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

[28] 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.

[29] Y. Ren and D. Li, "Fast and robust wrapper method for $N$-gram feature template induction in structured prediction," *IEEE Access*, vol. 5, pp. 19897–19908, 2017.

[30] Q. Wang, L. Wei, and R. A. Kennedy, "Iterative Viterbi decoding, trellis shaping, and multilevel structure for high-rate parity-concatenated TCM," *IEEE Trans. Commun.*, vol. 50, no. 1, pp. 48–55, 2002.

[31] W. Liu and W. Han, "Improved Viterbi algorithm in continuous speech recognition," in *Proc. Int. Conf. Comput. Appl. Syst. Model. (ICCASM )*, vol. 7, Oct. 2010, pp. V7–207.

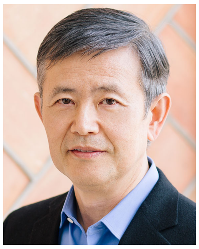
[32] H. Zhiheng, Y. Chang, B. Long, J. F. Crespo, A. Dong, S. Keerthi, and S. L. Wu, "Iterative Viterbi A* algorithm for k-best sequential decoding," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2012, vol. 1, no. 3, pp. 611–619.

[33] Sangoma Technologies. (2020). *Asterisk Project*. [Online]. Available: https://wiki.asterisk.org/wiki/display/ast/home

[34] (2021). *OpenSIPS (Open SIP Server) Project*. [Online]. Available: https://www.opensips.org/Documentation/Manuals

**DIOGO PEREIRA** received the M.Sc. degree in electrical and computer engineering from Nova University of Lisbon, in 2020, where he is currently pursuing the Ph.D. degree. He is also affiliated as a Researcher with the Instituto de Telecomunicações. His research interests include machine learning and performance evaluation of computer network protocols.

**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. He 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 of *Ad Hoc Networks* (Elsevier), the IEEE Open Journal of the Communications Society, and the 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.

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