Abnormal Signaling SIP Dialogs Detection based on Deep Learning

Diogo Pereira[†*], Rodolfo Oliveira[†*], Hyong S. Kim[‡]
[†]Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, FCT,
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[*]IT, Instituto de Telecomunicações, Portugal
[‡]Electrical & Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

***Abstract*—The detection of abnormal sequences of SIP messages in real-time is crucial to avoid SIP signaling-based attacks. In this paper, we propose a deep learning approach to detect signaling patterns of multimedia sessions established with the Session Initiation Protocol (SIP). The approach is based on a recurrent neural network (RNN). We study the performance of different Long Short-term Memory (LSTM) RNN architectures, which are trained using a SIP signaling dataset of trustworthy SIP dialogs captured by a SIP server. The trained RNNs are then used to detect the SIP dialogs in real-time. After characterizing the dataset adopted for the training, validation, and testing, we present the experimental results obtained for the different RNN architectures, showing that the classification probability of trustworthy SIP dialogs exceeds 93% in the test stage. Finally, we present two methodologies to detect abnormal SIP dialogs, i.e., not contained in the trustworthy training dataset. After a detailed analysis of the skewness and kurtosis computed with the numerical RNN outputs, we show that they can be used as classification features. The first method is based on a K-means unsupervised classifier, while the second one is based on a semi-supervised threshold-based classifier. Experimental results show that the threshold-based classifier achieves 99.45% of detection probability, showing the effective utility of the proposed methodology to detect abnormal SIP sequences in a short period of time.**

Keywords: Session Initiation Protocol, Deep Learning, Recurrent Neural Networks.

I. Introduction

The Session Initiation Protocol (SIP) has been adopted to support multimedia services, including Voice over Internet Protocol (VoIP) [1] or IP Multimedia Subsystem (IMS) services [2]. However, a significant number of SIP vulnerabilities is reported in the literature, being mainly related to the exploitation of SIP weaknesses found in the protocol implementation [3]. Through the combination of different signaling patterns, the attackers can cause denial-of-service, unauthorized access to a call, billing errors, and other type of attacks [4]. Consequently, it is important to identify potential malicious SIP signaling sequences received by the SIP servers, including new signaling sequences never observed before. While the already known potential malicious sequences can be detected in an automated way, the SIP sequences never observed before need to be analyzed by domain experts who can then assess their level of vulnerability.

The detection of anomalous SIP signaling sequences is challenging due to the high number of different signaling sequences, the order of the messages in the dialog, and the dialogs' variable length. In recent years, the adoption of machine learning and deep learning techniques has been observed in several areas of interest, due to the increase in processing power and the advances in data science. Multiple solutions have been presented to prevent SIP attacks caused by SIP message payload tampering and SIP message flooding. However, the attacks caused by SIP message flow tampering, a.k.a. attacks based on SIP signaling [4], have received limited attention and, as far as we know, the detection of abnormal 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 deep learning techniques to detect vulnerabilities caused by the SIP signaling patterns. Considering 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 classifying SIP signaling patterns, which can also be used to detect abnormal SIP messages. The main contributions of this work are summarized as follows:

- **C.1 -** Four classification models based on Long Short-term Memory (LSTM) Recurrent Neural Networks (RNNs) are proposed to classify SIP dialogs. The classification probability is evaluated based on experimental data, showing that it reaches at least 93% in all models;
- **C.2 -** To detect abnormal SIP dialogs we identify the classification features using the outputs of the LSTM RNNs in C.1. Specifically, we present experimental data demonstrating that the skewness and kurtosis of the the LSTM outputs can be used as classification features;
- **C.3 -** Using the classification features identified in C.2, two different classification schemes are proposed to detect abnormal SIP dialogs. An unsupervised scheme is proposed based on K-means clustering. A semi-supervised scheme is shown to reach higher performance, achieving a detection probability of 99.45%;

The rest of the paper is organized as follows. Literature review is presented in Section II. Section III describes the LSTM RNN models. Performance evaluation results are analyzed in Section IV. Finally, conclusions are drawn in Section V.

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

II. Related Work

The vulnerabilities of the SIP Protocol [5] have been targeted in different works [3], [4], [6], and can cause service interruption, service destruction, or unauthorized access to previously reserved computing resources or pools of SIP services. SIP service interruption can be caused by flooding attacks. 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 [7]. 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. An approach to mitigate parser-based flooding attacks was proposed in [8], where the SIP messages are classified before being parsed.

The majority of literature on SIP anomaly detection is particularly oriented to the problematic of malformed SIP messages. Malicious SIP messages are usually detected through intrusion detection systems (e.g. firewalls) [9], learning techniques [10] and/or identification of deviations from a priori statistics [11]. Apart from flooding and parser attacks, several vulnerabilities of the authentication protocols can also be explored. Different authentication protocols have been proposed for SIP, including one-factor authentication [12], and more secure schemes [13].

Additional SIP protocol vulnerabilities are related with the protocols' signaling logic and take advantage of defective implementations of the protocol. This type of attacks are based on SIP signaling, where possible protocol implementation errors can be explored by sending SIP messages to allow improper authentication mechanisms [4]. The SIP signaling vulnerabilities have motivated the authors in [14] to develop a SIP automatic debugger tool capable of analyzing the SIP messages flow and group them into dialogs to find protocols' compliance and interoperability faults. A rule-based approach was presented in [15] to mitigate SIP signaling vulnerabilities by capturing the contextual information in the SIP traffic. The mitigation of SIP signaling attacks was also addressed in [16], where the characterization of SIP sequences and their timings are used a posterior to detect deviations that may represent vulnerabilities.

Contrarily to works in [14]–[16], in our work 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 detect abnormal SIP dialogs that need to be categorized by domain experts. In this way, the vulnerability of the abnormal dialogs can be evaluated based on prior trustworthy SIP data.

III. Long Short-term Memory RNN Models

This section presents different LSTM RNN models to classify SIP dialogs according to a known dataset, i.e. in a supervised way. Four LSTM RNN models are proposed to classify a sequence of observed SIP messages as a known SIP dialog.

A. SIP Protocol

The SIP protocol supports multimedia sessions created by peers of user agents through the exchange of SIP messages. SIP messages can be either a request or a response. When a user agent requests a certain interaction a SIP message must be sent with a request containing one of the possible SIP methods (six in [5], although standardised 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 contains three 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. In what follows, we consider that all SIP servers and user agents in the SIP signaling path capture the SIP messages to detect the SIP dialogs. In this way, it is possible to detect new dialogs that may constitute new vulnerabilities.

B. Model Assumptions

Next we present the definitions that are used to describe how the supervised dataset of SIP dialogs is built. The dataset is used to train the LSTM RNNs. We start with the definition of a SIP message.

Definition 1. *A **SIP message** m_k, $k \in \mathcal{M} = \{1, 2, ..., M\}$, represents a specific type of SIP request or SIP response. The total number of SIP request 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 $\mathbf{d}_k =< m^{(1)}, m^{(2)}, ..., m^{(L_d)} >$, 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 a SIP server, the observations are the captured SIP messages and 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 $\mathbf{o}_k =< m^{(1)}, m^{(2)}, ..., m^{(L_o)} >$. Each SIP message is represented by $m^{(h)} = m_i$, $i \in \mathcal{M}$, $h \in \{1, 2, ..., L_o\}$. The*

symbol L_o represents the length of the observation. All SIP messages contained in a observation share the same SIP Call ID.

A requirement to meet when working with sequential neural networks is that the length of the input data must be described over consecutive discrete-time events. The set of events is represented by the observation. However, because the length of the observations can be variable, we transform each observation in a fixed-length stuffed sequence, denoted as a pad sequence.

Definition 4. *A **pad sequence** $\mathbf{n}_k$ associated to the observation $\mathbf{o}_k$, is a sequence of length $L_N = L_o + n$, where n represents the number of zeros added to the observation as follows, $\mathbf{n}_k =< \mathbf{o}_k, \underbrace{0, 0, ..., 0}_{(n)} >$. L_N represents a fixed length adopted in all pad sequences.*

The pad sequence is the data copied to the input of the LSTM RNNs during the training stage. Another aspect that must be evaluated is the type of data handled by the neural network, i.e., numerical or categorical data, since it influences how the input data is normalized. The SIP methods and responses are categorical data and their normalization process is presented in Definition 5.

Definition 5. *An **encoded SIP message** $\mathbf{m}'_i$ is a boolean vector describing the SIP message m_i. The vector has length M and is obtained using a One Hot Encoder [17].*

Next we define the state space of the LSTM inputs.

Definition 6. *The **input state space** of the LSTM RNN, denoted by $\mathcal{X}$, is formed by the set of padded sequences of the permutations (with repetition) of the L_o SIP messages integrating each observation $\mathbf{o}_k$. Since permutations sum up M^{L_o}, $\mathcal{X} = \{\mathbf{n}_1, ..., \mathbf{n}_k\}$, $k = M^{L_o}$*

Since the goal of this work is to classify the input data (pad sequences) in multiple SIP dialogs contained in the trustworthy dataset, the outputs of the LSTM are the SIP dialogs to be classified. Consequently, each output of the LSTM represents a unique SIP dialog, i.e., a unique sequence of SIP messages.

Definition 7. *The **output state space** of the LSTM RNN is represented by $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$, where N stands for the total number of unique SIP dialogs contained in the trustworthy dataset, and $\mathbf{y}_i$ each unique SIP dialog.*

From the definitions presented so far, the classification problem to be solved by the LSTM can be seen as a regression problem $\mathcal{Y} = f(\mathcal{X}, \beta)$, where $f(.)$ is the estimate function obtained by regulating the LSTM weights β during the training stage of the LSTM RNN. Once trained, the LSTM RNN can be validated and tested using $f(.)$ in real time to compute the LSTM output values, so the input data can be classified as a unique SIP dialog. The notation adopted so far is represented in Table I.

TABLE I
TABLE OF SYMBOLS

Symbols	Definitions
$\mathbf{d}_k$	SIP dialog k.
L_d	Length of a SIP dialog $\mathbf{d}_i$.
L_o	Length of an observation.
L_N	Length of a pad sequence.
m_k	SIP message k.
$\mathcal{M}$	Set with all the possible SIP messages.
M	Number of possible SIP methods and responses.
$\mathbf{n}_k$	Pad sequence of an observation $\mathbf{o}_k$.
n	Number of zeros added in the pad sequence.
N	Number of unique SIP dialogs.
$\mathbf{o}_k$	Observation k.
$\mathcal{X}$	Input state space.
$\mathcal{Y}$	Output state space.

TABLE II
LSTM MODEL 1.X.

Step 1:	The LSTM input layer receives a $1 \times L_N \times M$ input sequence $\mathbf{n}_k$, produced by the Pad Sequence and the One Hot Encoder.
Step 2:	The LSTM layer processes one step at each L_N discrete time units of $\mathbf{n}_k$ and returns a $1 \times N$ sequence, $\mathbf{h}_0$, with real values in the interval $[-1, 1]$.
Step 3:	During the training stage if the model has a Dropout probability (Model 1.1) some LSTM outputs, $\mathbf{h}_0$ are excluded.
Step 4:	The Dense layer receives either the LSTM or Dropout outputs, depending on the model, and produces a $1 \times N$ output vector with real values in $[0, 1]$.

C. LSTM RNN Models

Four LSTM RNN models are next presented to classify the most likely SIP dialog given an observed input sequence of L_o SIP messages. The goal is to study the performance of each model. Figure 1(a), presents the initial model, formed by one LSTM layer and a Dense layer, i.e. a Multilayer Perceptron that receives the various inputs and connects all of them to each neuron responsible for the model's output. The LSTM layer is responsible for processing the encoded SIP messages. The Dense layer is responsible for the identification of the most probable dialog. The LSTM model is described in Table II. A second LSTM layer was added in the second model, illustrated in Figure 1(b) and described in Table III. The rationale of the two LSTM layers is to increase the degrees of freedom of the regression model by increasing the number of weights β.

A dropout probability layer was added to the models in Figures 1(c) and 1(d). The dropout probability layer is used during the training stage to prevent the LSTM model from overfitting, where input and recurrent connections to LSTM units are randomly excluded from activation and weight updates.

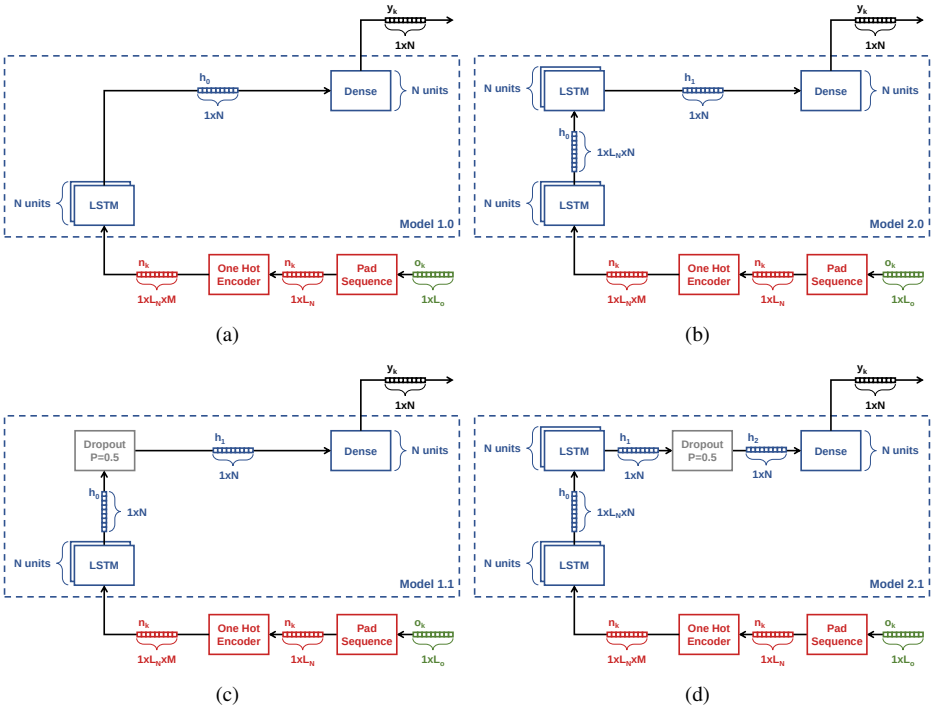

Fig. 1. LSTM RNN models: (a) Model 1.0; (b) Model 2.0; (c) Model 1.1; (d) Model 2.1.

TABLE III
LSTM MODEL 2.X.

Step 1:	Equal to Model 1.X in Table II.
Step 2:	The LSTM layer process one step at the time the L_N elements of $\mathbf{n}_k$ and returns the outputs of each timestep (return sequences). The return sequence is a $1 \times L_N \times N$ sequence, $\mathbf{h}_0$, with real values in $[-1, 1]$.
Step 3:	The second LSTM layer receives the output $\mathbf{h}_0$ and returns a $1 \times N$ sequence, $\mathbf{h}_1$, with real values in $[-1, 1]$.
Step 4:	Equal to Step 3 of Model 1.X in Table II.
Step 5:	Equal to Step 4 of Model 1.X in Table II.

TABLE IV
LSTM RNN PARAMETERS

Model Parameters	
M	18
L_N	56
LSTM layer (N)	Units = 1492
Dense layer	Units = 1492
Dense layer activation function	Softmax
Dropout probability	P = 0.5
Epochs	500
Batch size	933
Loss Function	Categorical cross entropy
Optimizer	Adam (learning rate = 0.001)

IV. PERFORMANCE EVALUATION

A. Dataset Characterization

The performance evaluation adopts the SIP dataset described in [18]. 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 an integer value. The dataset contains a total of 18782 SIP dialogs established between a group of 249 user agents, which corresponds to 1492 unique SIP dialogs. The dataset was randomly divided into three different datasets, the train, the validation, and the test datasets, by only considering 50%, 20%, and 30% of the information contained in the original dataset, respectively. The majority of the SIP dialogs in the dataset, 66.23%, only occur once, which may result in lower detection performance, especially when the unique SIP dialogs are only included in the validation or test datasets and not considered during the train stage. Finally, the number of unique SIP messages and the SIP dialogs' length were identified in the dataset, resulting in $M = 18$ SIP messages and the longest SIP dialog is $\max\{L_d\} = L_N = 56$.

B. SIP Dialogs' Classification

The LSTM RNN models presented in Subsection III-C were trained during 500 epochs, and the LSTM weights were updated 101 times per epoch. To consider the randomness of the weights assignation, each model was trained 10 times (Multiple Run) considering the parameters described in Table IV. Regarding the performance metrics, the detection probability, P_D, represents the probability of successful classifying the inputs of the LSTM in the appropriate SIP dialog. The appropriate SIP dialog is identified by the LSTM output achieving the maximum value during the multiple run.

Table V presents the detection probability achieved by each LSTM model during the train, validation, and test stages, respectively. The results indicate that all models can fully learn the train dataset ($P_D = 1$). However, similar results are obtained for the four proposed models and, consequently, there is no advantage of choosing a model with a dropout layer or with two LSTM layers when compared to the simplest model (model 1.0). As indicated in the table, the detection probability of the validation and test datasets is 92.80% and 93.54%, respectively. For these cases the probability never reaches 100%, because the sets contain unique SIP dialogs that were not considered in the training stage (they are only included in the test or validation datasets). Knowing that 66.23% of the SIP dialogs in the dataset only occur once, the total number of dialogs included in the validation dataset but not included in the train set are 277 (7.20%) and for the test dataset are 362 (6.46%). These percentages represent the miss-detection probability, that is, the detection of the SIP dialogs fails for the ones not included in the train/trustworthy dataset or, in other words, the proposed models are capable of correctly classifying all SIP dialogs included in the training set.

Due to the lower computational complexity of model 1.0 and the performance results in Table V, in what follows we always refer to the model 1.0 when mentioning the LSTM RNN.

TABLE V
P_D ACHIEVED IN THE TRAIN, VALIDATION, AND TEST STAGES.

Model	1.0	1.1	2.0	2.1
P_D (train)	100.00%	100.00%	100,00%	100,00%
P_D (validation)	92.80%	92.80%	92.80%	92.80%
P_D (test)	93.54%	93.54%	93.54%	93.54%

C. Detection of Abnormal SIP Dialogs

This subsection proposes two different classification schemes to detect abnormal SIP dialogs. We consider that the abnormal SIP dialogs are the ones not included in the train dataset that only contains trustworthy dialogs. The goal behind the detection of abnormal SIP dialogs is to identify SIP dialogs not included in the train dataset.

The output of the LSTM RNN is used to extract the classification features. More specifically, the central moments of the LSTM RNN output were computed to evaluate if they can be used to distinguish new SIP dialogs not contained in the train dataset. Figure 2 presents the joint probability density function (PDF) of the normalized Skewness and Kurtosis values computed for the new SIP dialogs and for the ones already trained. In what follows all results are obtained with the test dataset. The figure shows that the data can be effectively divided in two possible labels. The samples labeled as "Already Trained Dialog" have a higher skewness and kurtosis because the LSTM RNN can predict those dialogs, so the distance represented by those moments is close to 1. However, for some samples labeled as "New Dialogs" the values are identical to the dialogs already trained, due to the similarity between the new/abnormal dialogs and the already trained ones.

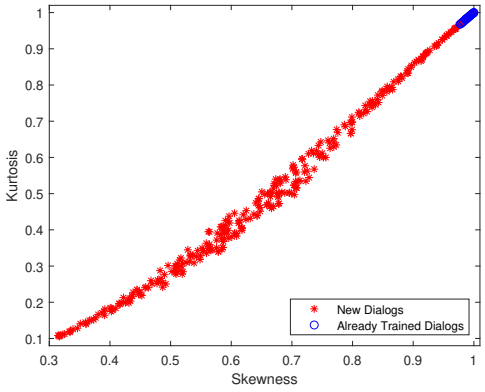

Fig. 2. Joint PDF of the normalized skewness and kurtosis of the LSTM RNN output values.

Using the data represented in Figure 2, the first method to classify the new SIP dialogs is based on the K-means unsupervised learning algorithm. The K-means algorithm was computed considering 2 data clusters, representing the new/abnormal SIP dialogs and the ones already included in the train dataset. The results in Figure 3 represent the output of the K-mean algorithm when the skewness and the kurtosis of the test dataset are used as classification features.

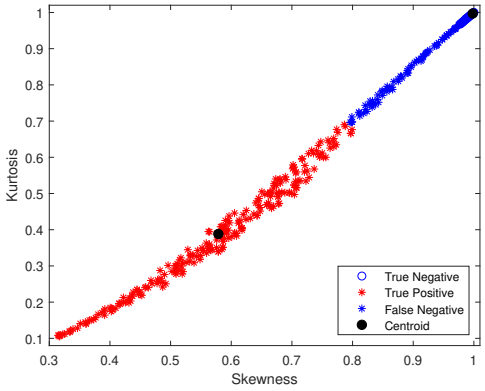

Fig. 3. K-means Classification to detect the New Dialogs

The second method is a semi-supervised threshold-based classifier also supported by the PDF in Figure 2. In this method the kurtosis and skewness are computed using the training dataset. The kurtosis and sknewness thresholds are given by $\lambda_k = \mu_k - \sigma_k^2$ and $\lambda_s = \mu_s - \sigma_s^2$, respectively, where μ_k and μ_s represent the mean of the kurtosis and skewness values computed from the LSTM RNN outputs for all SIP dialogs in the train set, and σ_k^2 and σ_s^2 represent the variance of the kurtosis and skewness values, respectively. A SIP dialog is classified as a known dialog, hypothesis H_0, or new/abnormal dialog, hypothesis H_1, according to the conditions

$$H_0 : \mu^3 \geq \lambda_s, \mu^4 \geq \lambda_k,$$
$$H_1 : \mu^3 < \lambda_s, \mu^4 < \lambda_k,$$

where μ^3 and μ^4 represent the skewness and kurtosis of the LSTM RNN output obtained with the SIP dialog (from the test dataset) to classify.

Figure 4 illustrates the results of the threshold-based classifier applied to the test dataset, identifying true and false positives, and true and false negatives, close to the decision thresholds λ_k and λ_s. The thresholds were computed from the outputs of the LSTM RNN (model 1.0) obtained with the training dataset, obtaining $\lambda_k = 0.999759$ and $\lambda_s = 0.999828$. As confirmed by the threshold values, both thresholds are close to 1, meaning that for trained SIP dialogs the skewness and kurtosis of the LSTM RNN outputs are close to 1 and low uncertainty is observed in the magnitude value of the outputs. Contrarily, for non-trained SIP dialogs the outputs of the RNN exhibit higher uncertainty and, consequently, both skewness and kurtosis values are below the thresholds.

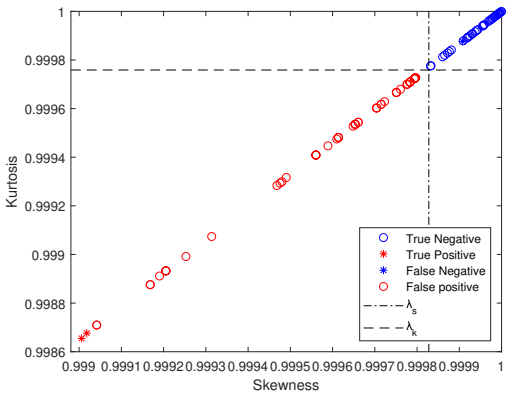

Fig. 4. Threshold Classification to detect the New Dialogs.

Finally, Table VI compares the probabilities of detection (P_D) and false alarm (P_{FA}) of abnormal SIP dialogs for the K-means and the threshold-based classifiers, indicating the higher detection performance of the threshold-based classifier, although exhibiting higher probability of false alarm.

TABLE VI
ABNORMAL SIP DIALOGS' DETECTION AND FALSE ALARM PROBABILITIES.

	K-means	Threshold-based
P_D	69.61%	99.45%
P_{FA}	0.00%	3.95%

V. CONCLUSIONS

This work proposed four classification models based on LSTM RNNs to classify SIP dialogs. The detection probability was evaluated based on experimental data. To detect abnormal SIP dialogs, we have adopted classification features computed from the output of the LSTM RNN model and two different classification schemes were proposed. A semi-supervised scheme is shown to reach higher performance, achieving a detection probability of 99.45%, thus confirming the effective utility of the proposed methodology to detect abnormal SIP sequences in a short period of time.

ACKNOWLEDGEMENTS

This work was funded by Fundação para a Ciência e Tecnologia, under the projects InfoCent-IoT (PTDC/EEI-TEL/30433/2017), CoSHARE (PTDC/EEI-TEL/30709/2017), and RFSense (UIDB/50008/2020).

REFERENCES

[1] A. Uzelac and Y. Lee. Voice over ip (voip) sip peering use cases. RFC 6405, RFC Editor, November 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 Communications Magazine*, 47(11):156–163, 2009.

[3] D. Sisalem, J. Kuthan, and S. Ehlert. Denial of service attacks targeting a sip voip infrastructure: attack scenarios and prevention mechanisms. *IEEE Network*, 20(5):26–31, 2006.

[4] 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 Communications Surveys Tutorials*, 8(3):68–81, 2006.

[5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Sip: Session initiation protocol. RFC 3261, RFC Editor, June 2002.

[6] Sven Ehlert, Dimitris Geneiatakis, and Thomas Magedanz. Survey of network security systems to counter sip-based denial-of-service attacks. *Computers & Security*, 29(2):225 – 243, 2010.

[7] 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*, 8:112574–112584, 2020.

[8] 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 Transactions on Network and Service Management*, 12(3):467–482, 2015.

[9] H. Li, H. Lin, H. Hou, and X. Yang. An efficient intrusion detection and prevention system against sip malformed messages attacks. In *2010 International Conference on Computational Aspects of Social Networks*, pages 69–73, 2010.

[10] Mohamed Nassar, Radu State, and Olivier Festor. Monitoring sip traffic using support vector machines. In Richard Lippmann, Engin Kirda, and Ari Trachtenberg, editors, *Recent Advances in Intrusion Detection*, pages 311–330, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[11] 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 *2011 4th IFIP International Conference on New Technologies, Mobility and Security*, pages 1–5, 2011.

[12] Hamed Arshad and Morteza Nikooghadam. An efficient and secure authentication and key agreement scheme for session initiation protocol using ecc. *Multimedia Tools Appl.*, 75(1):181–197, January 2016.

[13] Yuanyuan Zhang, Kunming Xie, and Ou Ruan. An improved and efficient mutual authentication scheme for session initiation protocol. *PLOS ONE*, 14(3):1–15, 03 2019.

[14] D. Bao, D. L. Carni, L. De Vito, and L. Tomaciello. Session initiation protocol automatic debugger. *IEEE Transactions on Instrumentation and Measurement*, 58(6):1869–1877, 2009.

[15] A. Lahmadi and O. Festor. A framework for automated exploit prevention from known vulnerabilities in voice over ip services. *IEEE Transactions on Network and Service Management*, 9(2):114–127, 2012.

[16] D. Golait and N. Hubballi. Detecting anomalous behavior in voip systems: A discrete event system modeling. *IEEE Transactions on Information Forensics and Security*, 12(3):730–745, 2017.

[17] David Harris and Sarah Harris. *Digital Design and Computer Architecture, Second Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2012.

[18] Mohamed Nassar, Olivier Festor, et al. Labeled voip data-set for intrusion detection evaluation. In *Meeting of the European Network of Universities and Companies in Information and Communication Engineering*, pages 97–106. Springer, 2010.