

SIP-Classifier: Unsupervised Classification of SIP-IMS Signaling With Transformer and Clustering

Antonio Iacobelli[✉], Giorgio Franceschelli[✉], Lorenzo Rinieri[✉], *Member, IEEE*, Mirco Musolesi[✉], *Member, IEEE*, Marco Prandini[✉], *Member, IEEE*, and Franco Callegati[✉], *Senior Member, IEEE*

Abstract—Ensuring the reliability of voice services in 5G networks requires effective detection of anomalies in IMS signaling. However, this task remains challenging due to the architectural complexity of IMS and the large volume of signaling data. In this paper, we propose *SIP-Classifier*, an unsupervised methodology that combines Transformer-based representation learning with clustering to identify anomalous SIP sequences. The approach encodes SIP messages through protocol-aware tokenization, learns latent representations via an autoregressive Transformer, and clusters them to distinguish valid from anomalous flows. We evaluate the method on real-world IMS data collected from operational 5G networks. It achieves 98% accuracy, 98% precision, 95% recall, and a 96% F1-score, significantly outperforming state-of-the-art approaches.

Index Terms—SIP, IMS, Transformer, clustering, anomaly detection.

I. INTRODUCTION

SINCE the introduction of the first iPhone in 2007, the number of smartphones sold annually has grown dramatically, increasing from hundreds of millions to over 1.5 billion units within a decade [1]. This has enabled a pervasive always-online experience and fostered the widespread adoption of applications providing a variety of communication services. Despite this evolution, voice services remain a fundamental component of mobile networks, now provided through technologies such as Voice over LTE (VoLTE) [2] in 4G and Voice over New Radio (VoNR) [3] in 5G standalone deployments.

Therefore, ensuring the reliability of voice services remains a critical requirement in modern mobile networks. Voice signaling relies on the IP Multimedia Subsystem (IMS) [4], a complex and highly modular architecture where multiple

network functions interact through heterogeneous protocols. At the core of IMS signaling is the Session Initiation Protocol (SIP) [5], which governs session establishment, modification, and termination [6]. A single communication session involves multiple interacting protocols (e.g., SIP, Diameter, RTP) [7], [8], distributed network functions, and strict temporal dependencies across signaling messages. This complexity often leads to anomalous signaling behaviors caused by misconfigurations, software faults, or protocol inconsistencies. Detecting such anomalies is essential for maintaining service reliability and operational stability, yet it remains largely dependent on manual inspection or rule-based systems, which do not scale to the volume and diversity of modern network traffic.

Recent advances in machine learning have enabled data-driven approaches to SIP anomaly detection [9], [10]. However, existing solutions [11], [12] typically rely on supervised learning applied to synthetic datasets [13], limiting their applicability in real-world deployments. Moreover, most approaches treat signaling data either as flat feature vectors or as generic sequences, failing to capture the structured, protocol-driven nature of SIP procedures. As a result, they exhibit poor generalization when applied to real network environments.

In this paper, we argue that IMS signaling should be modeled as structured sequential data, where the semantics of a communication session emerge from the ordered interaction of network functions and protocol messages. Building on this insight, we propose *SIP-Classifier*, a novel unsupervised methodology that leverages Transformer-based representation learning to model SIP-IMS signaling sequences in a protocol-aware manner. Unlike traditional natural language processing applications, SIP signaling exhibits strict syntactic and semantic constraints, requiring domain-specific tokenization strategies to preserve protocol semantics. Our approach first transforms SIP-IMS signaling procedures into symbolic sequences through a protocol-aware tokenization scheme that captures the structural properties of signaling messages. We then train an autoregressive Transformer model [14], [15] to learn latent representations of signaling sequences by modeling their sequential dependencies. Rather than using the model for sequence generation, we exploit its internal representations as embeddings of signaling behavior. These embeddings are subsequently processed through dimensionality reduction [16] and clustered to identify groups of structurally similar signaling flows. This design is key to enabling fully unsupervised

Received 4 May 2026; revised 14 July 2026; accepted 15 July 2026. Date of publication 20 July 2026; date of current version 31 July 2026. This study was partially funded by the projects “Empowering awareness with a digital twin for mission critical communications (EDIT4MCC)”, CUP J53C24002800006, funded by “Ministero dell’Università e della Ricerca-program PRIN 2022”, and project D.A.I.MON, CUP B39J23002440005, funded by “Ministero dello Sviluppo e del Made in Italy”. This work was supported by Agenzia per la cybersicurezza nazionale under the programme for promotion of XL cycle PhD research in cybersecurity–J33C24002860006. The views expressed are those of the authors and do not represent the funding institution. The associate editor coordinating the review of this article and approving it for publication was L. Yang. (Corresponding author: Antonio Iacobelli.)

Antonio Iacobelli, Giorgio Franceschelli, Lorenzo Rinieri, Marco Prandini, and Franco Callegati are with the Department of Computer Science and Engineering (DISI), University of Bologna, 40126 Bologna, Italy (e-mail: antonio.iacobelli@unibo.it).

Mirco Musolesi is with DISI, University of Bologna, 40126 Bologna, Italy, and also with the Department of Computer Science, University College London, WC1E 6BT London, U.K.

Digital Object Identifier 10.1109/TNSM.2026.3715301

anomaly detection of SIP-IMS signaling patterns: labels are not used for representation learning, clustering, hyperparameter tuning, or model selection; they are only used post hoc to semantically interpret the discovered clusters and to report evaluation metrics.

We evaluate the proposed methodology on a real-world dataset collected from IMS cores of multiple operational 5G networks using high-capacity passive probes. Our results show that *SIP-Classifier* significantly outperforms state-of-the-art approaches when applied to real signaling data, highlighting the limitations of existing methods trained on synthetic datasets and demonstrating the effectiveness of Transformer-based representation learning in this domain.

The contributions of this paper are summarized as follows:

- We introduce a novel representation learning framework for SIP signaling based on the Transformer architecture, enabling protocol-aware modeling of IMS procedures;
- We design a novel, intuitive domain-specific tokenization strategy that captures the structural semantics of SIP-IMS signaling flows;
- We propose an unsupervised pipeline combining latent representation learning and clustering for anomaly detection without labeled data;
- We provide an extensive evaluation on a real-world IMS dataset from operational 5G networks, demonstrating strong performance on previously unseen signaling sequences within the evaluated multi-network corpus and superior performance compared to the considered baselines. We make the dataset publicly available.¹

The remainder of this paper is organized as follows. Section II discusses the main challenges of SIP signaling classification in IMS, highlights the limitations of existing approaches, and motivates the objectives of this work. Section III describes the dataset used in this work. Section IV presents *SIP-Classifier* and its main components. Section V reports the experimental evaluation and discusses the main results. Section VI discusses the limitations of the proposed approach and outlines directions for future work. Finally, Section VII concludes the paper.

II. BACKGROUND AND RELATED WORK

A. Motivations and Challenges

IMS outlines an extensive set of procedures for various signaling functions, all based on the SIP protocol [17], that can be divided into five different categories: (1) Registration and De-registration, (2) Session Initiation, (3) Session Termination, (4) Session Failure, and (5) Session Redirection. Our method focuses on SIP control-plane signaling dialogs across these procedures, and does not account for RTP media-plane traffic. The first category comprises procedures for the registration and de-registration of both new and existing users within the IMS network, while the others focus on the session establishment process, necessary to set up a call flow between two end users. Additionally, [18] emphasizes that each of the aforementioned IMS signaling categories can be

further subdivided into approximately 20 call-flow variants, and additional permutations arise when the caller and callee belong to different network operators, leading to at least nine cross-operator scenarios [19]. Further deployment-level heterogeneity can also arise from SIP/IMS interoperability mechanisms, such as IPv4–IPv6 coexistence and translation through SIP proxies in IMS environments [20].

Beyond the signaling diversity, IMS Core Networks host numerous Application Servers (AS) and network functions (NF) that can be dynamically inserted into a call dialog. For instance, a call may be diverted to the Media Resource Function (MRF) to play pre-recorded announcements or tones before the conversation begins [21]. Anomalous signaling flows may arise within each of the aforementioned scenarios, due to malfunctions or misconfigurations within IMS Core Network components, transient NF failures, and issues at lower layers of the TCP/IP stack. The intrinsic complexity of IMS signaling—arising from heterogeneous call scenarios, numerous SIP procedures, and many network functions—makes it hard even for experienced IMS experts to distinguish normal from abnormal SIP flows, and this variability often renders traditional approaches based on formal modeling [22], [23] and fixed-rule definitions of SIP interactions [24] ineffective.

B. State-of-the-Art Methods and Their Limitations

For the reasons outlined above, recent approaches have turned to data-driven techniques that aim to capture complex and context-dependent SIP signaling patterns and thereby detect deviations from normal operation [9], [11], [12], [25], [26]. However, despite their value, these methods still exhibit several limitations that hinder their applicability to real 5G SIP-IMS deployments. First, these techniques are trained and validated exclusively on synthetic SIP traces, primarily the dataset of Nassar et al. [13], since genuine SIP datasets are rarely released because of operator confidentiality and privacy constraints. Although invaluable for initial prototyping, such synthetic data do not cover all SIP request methods and response codes. Consequently, the robustness of these methods against the variability, noise, and heterogeneity of operational SIP-IMS traffic remains largely unverified. Second, they do not capture the actual structure of SIP-IMS signaling. In particular, they model a dialog essentially as an exchange between two endpoints (caller and callee), while real IMS sessions may traverse several intermediate network functions and application servers. Therefore, they ignore part of the signaling path and the additional complexity introduced by IMS service logic. Third, these methods are not designed to directly distinguish benign traffic from anomalous traffic. Instead, they mainly learn to tell apart signaling patterns that were already seen during training from those that were not. This is a strong limitation, because it assumes that benign traffic is sufficiently represented in the training set. In practice, this assumption is unrealistic, especially in SIP-IMS networks, where benign signaling can take many different forms, as discussed at the beginning of this section. Fourth, their evaluation is conducted under a train-test split with substantial overlap in signaling patterns. In fact, only a minority of the test dialogs corresponds to previously unseen patterns, while most test instances still

¹<https://github.com/UniboSecurityResearch/Sip-Classifier/>

TABLE I
IMS CORE NETWORK FUNCTIONS INCLUDED IN OUR DATASET, WITH THEIR IDS AND DESCRIPTIONS

ID	IMS Network Function	Description
ASLI	Application Server for Lawful Interception	Enables lawful interception in compliance with regulatory requirements.
ATCF	Access Transfer Control Function	Facilitates seamless call transfer between access networks, ensuring session continuity during hand-overs.
BCCAS	Application Server for Broadcast Call Control	Manages group communication, broadcast, and conferencing services.
ECSCF	Emergency Call Session Control Function	Handles emergency calls.
GTW	Gateway	Converts signaling and media between heterogeneous networks, preserving session continuity.
IBCF	Interconnection Border Control Function	Secures the boundary between service-provider networks and manages signaling information.
ICSCF	Interrogating Call Session Control Function	Input node that performs dynamic routing to the appropriate serving node.
LBIMP	Load Balancer	Obtains load level reports from the S-CSCF and deregisters users from overloaded S-CSCF.
MSC	Mobile Switching Center	Connects calls by switching digital voice packets.
MTAS	Multimedia Telephony Application Server	Provides multimedia telephony services and executes logic for voice and video sessions.
OPBMRF	Media Resource Function	Performs media processing tasks such as mixing, transcoding, announcements, and tone generation.
PCSCF	Proxy Call Session Control Function	First contact point for the UE, handling authentication and SIP session management.
SCSCF	Serving Call Session Control Function	Main node for user registration, authentication, and SIP session control.
SLEVOOSP	SideLink EVOLution Open Settlement Protocol	Exchanges authorization, accounting, and usage information among service providers.
TSC	Tunneled Services Control Function	Responsible for the creation and management of service tunnels.
Undefined	Undefined Network Function	Role currently unknown to industry specialists, may ultimately map to one of the other network functions in this table.

belong to dialog types already represented in the training data. As a result, the reported performance does not fully reflect what happens in real SIP-IMS networks, where signaling is more complex and where the detector must handle many more unseen combinations of call flows, network functions, and traffic conditions. Overall, these methods were designed and validated for simplified SIP scenarios that no longer reflect the structure of modern carrier networks, where SIP signaling is typically part of a broader IMS architecture. Therefore, their effectiveness on real 5G SIP-IMS traffic cannot be taken for granted. We empirically demonstrate this aspect in Section V-F.

C. Objectives and Contributions

Clustering techniques are widely used to uncover hidden patterns in network traffic [27], [28]. By operating without explicit supervision, these methods can reveal the intrinsic structure of the data and identify similarities among heterogeneous signaling sequences. However, applying conventional clustering methods directly to SIP signaling flows is not straightforward. Traditional clustering approaches generally assume data described by well-defined numerical features, whereas SIP call flows are more naturally represented as structured textual sequences. This makes raw string-based SIP representations poorly suited for traditional clustering pipelines. To address this limitation, we employ a Transformer-based model to learn expressive representations of SIP sequences, capturing contextual information, sequential structure, and long-range dependencies. These representations are subsequently reduced through Principal Component Analysis and then clustered to explore the latent structure of the feature space.

Although our dataset includes class labels, they are not used during representation learning, dimensionality reduction,

clustering, or hyperparameter selection. Instead, the Transformer model, the PCA configuration, the clustering algorithm, and their corresponding hyperparameters are selected exclusively through unsupervised criteria, including model-specific objective functions and clustering-specific internal metrics, depending on the stage of the pipeline. This design ensures a fully unsupervised model selection process, while the available labels are used only a posteriori to validate and interpret the results. By contrast, supervised approaches are effective when reliable annotations are available. However, obtaining exhaustive and stable labels in the SIP signaling domain is difficult. In real operational environments, even datasets annotated by domain experts may contain a significant number of sequences labeled as “unknown”, reflecting both the ambiguity of manual annotation and the difficulty of assigning predefined classes to all signaling flows in advance. For these reasons, unsupervised methods are inherently more scalable, less dependent on annotation quality, and better suited to large and evolving signaling datasets. We therefore consider an unsupervised approach not merely as an alternative for settings where labels are unavailable, but as a methodologically appropriate choice for realistic SIP-IMS scenarios.

III. DATASET

The dataset employed in this study was provided by industry specialists who manage high-capacity passive probes embedded in the IMS cores of multiple Italian 5G networks.² The original data consisted of packet-level traffic captures, including both headers and payload. Since the payload was not relevant for this analysis, only packet headers were retained and processed. The source and destination addresses at both

²RESI: <https://www.resi.it/it/>

TABLE II
SIP REQUEST METHODS ALONG WITH A DESCRIPTION AND THEIR CORRESPONDING IDs USED IN OUR DATASET

ID	Message	Description
A	ACK	Confirms receipt of a final response to an INVITE request.
B	BYE	Terminates an existing call or session.
C	CANCEL	Cancels a pending request (e.g., an INVITE before the session is established).
E	SUBSCRIBE	Requests notification of events from the server, starting a subscription.
F	REFER	Indicates a request for the recipient to contact a third party using the provided URI.
H	INFO	Sends mid-session information unrelated to the session description or media transport.
I	INVITE	Initiates a call or a multimedia session, providing the session description for negotiation.
N	NOTIFY	Provides event notifications to a subscriber about changes in state.
P	PRACK	Acknowledges provisional responses to INVITE requests (reliable provisional responses).
Q	QUEUED	The call is in a queue and will be processed when resources become available.
R	RINGING	The call is being routed, and the destination device is ringing.
S	SESSION PROGRESS	Indicates that the session is in progress.
T	TRYING	The server has received the INVITE request and is attempting to process it.
U	UPDATE	Allows a client to update the parameters of a session.
W	CALL FORWARDED	The call has been redirected to another destination.

TABLE III
SIP RESPONSE CODES AND THEIR DESCRIPTION

Code	Response	Description
2xx	Success	The request has been received and accepted.
3xx	Redirection	The request should be delivered to a different place.
4xx	Client error	The request cannot be processed due to a request error.
5xx	Server error	The request cannot be processed due to a server failure.
6xx	Global failure	The request cannot be processed at any server.

TABLE IV
COMPARATIVE ANALYSIS BETWEEN THE DATASET EMPLOYED IN THIS WORK AND [13], WHICH HAS BEEN EXTENSIVELY EMPLOYED IN THE LITERATURE FOR THE DETECTION OF SIP ATTACKS

Dataset	Number of different SIP Request Methods	Number of different SIP Response Codes	Number of different Network Functions	Number of unique signaling flows	Min signaling flow length	Max signaling flow length
[13]	4	13	0	1644	2	56
Ours	15	26	16	11329	3	867

the IP and transport layers were used to identify the originating and receiving network functions. Because each NF is associated with operator-specific IP addresses, all IPs were replaced with anonymized NF identifiers. This transformation serves two purposes: (i) preserving confidentiality and privacy; (ii) ensuring a more consistent semantic representation, as different IP addresses may correspond to the same NF across operators, while signaling semantics depend on the function rather than on specific IPs.

At the application level, headers provide information about the role of each packet within the SIP-IMS signaling flow. In general, signaling flows are represented as sequences of elements in the following format:

network_function_1, network_function_2,
message_ID[-response_code]

where the first and second fields identify the sender and receiver network functions, the third field specifies the SIP request method, and an optional response code is appended when relevant. The complete mapping of NF IDs appears in Table I, while the admissible message_IDs and response_codes are listed in Tables II and III, respectively.

The SIP signaling flows of the dataset were then rigorously annotated by the aforementioned industry specialists, yielding a corpus that accurately reflects production conditions across distinct carrier networks. Each record comprises two attributes: (i) *signaling_flow*, as explained above, and (ii) *class*, the expert-assigned label, which is 0 for *valid*, 1 for *anomalous*, and 2 for *unknown*. The dataset contains 10,000 *valid* (2,823 without considering duplicates) dialogs that fully comply with the IMS specification [17], 8,344 *anomalous* (1,305 without considering duplicates) dialogs exhibiting deviations caused by IMS core malfunctions, NF misconfigurations, or lower layer issues, and 10,000 *unknown* (7,201 without considering duplicates) dialogs whose status remains ambiguous even after expert inspection. It is important to note that, in real-world scenarios, obtaining large-scale signaling data with reliable valid/anomalous labels is generally not feasible. The annotated dataset used here was specifically curated for validation purposes and does not reflect typical operational conditions. For this reason, as introduced earlier and further discussed in the next section, this work focuses on a fully unsupervised methodology. The available labels are

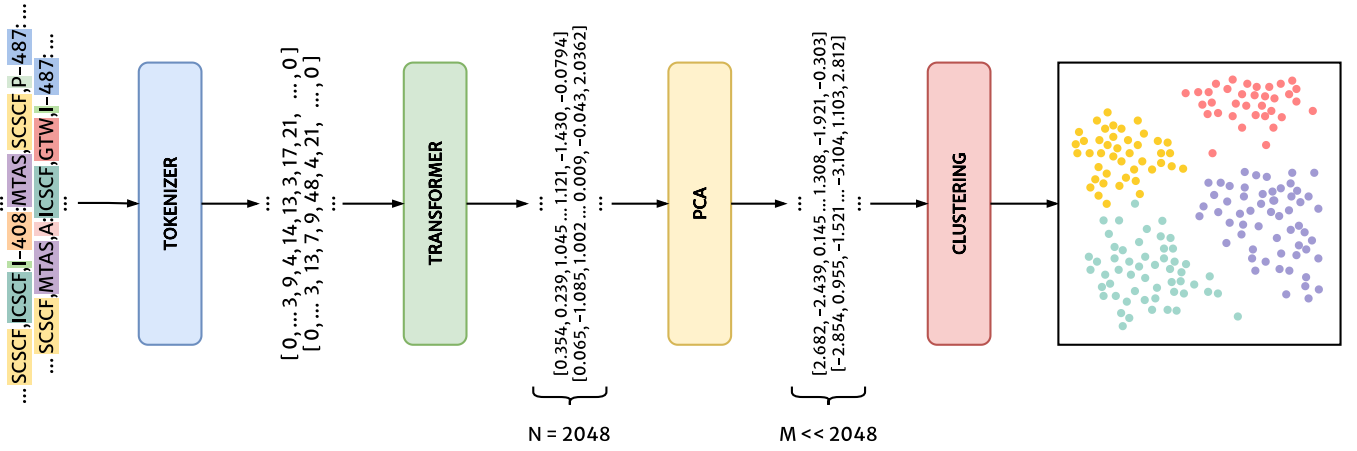


Fig. 1. The full pipeline of our methodology. First, the input sequences are tokenized based on their core elements. Then, the tokenized sequences are processed by our trained transformer, and the 2048-long latent representations are collected. Finally, the features are reduced with PCA and organized into clusters via the K-Means++ algorithm.



Fig. 2. On the left, two examples of our Tokenizer in action. On the right, a snapshot of the whole vocabulary used by our Tokenizer given the training data.

used only *a posteriori* to evaluate the proposed approach, and not for model design or training.

Finally, Table IV highlights the differences between this dataset and the widely used one by Nassar et al. [13]. Our dataset exposes three times as many SIP request methods, twice the variety of response codes, and, crucially, 16 distinct network functions, a dimension entirely absent from the earlier benchmark. It also offers an order-of-magnitude increase in unique signaling flows, with dialog lengths stretching from 3 to 867 hops. Additionally, the dataset of Nassar et al. [13] also exhibits two important limitations. First, the dataset does not provide a clean dialog-level distinction between benign and anomalous SIP flows. It is organized into benign scenarios and attack scenarios, however, traces generated under attack scenarios still contain SIP messages and dialogs that also appear in benign scenarios. Therefore, the boundary between benign and anomalous flows is inherently ambiguous, and a reliable dialog-level distinction would require additional annotations that are not provided in the original dataset. Second, the dataset is markedly imbalanced. When duplicated SIP signaling sequences are retained, it contains 18,782 non-anomalous dialogs versus only 4,325 anomalous attack dialogs. The imbalance becomes even more pronounced when only unique signaling patterns are considered, as the non-anomalous portion still comprises 1,492 unique SIP dialogs, whereas the anomalous portion contains only 152 unique

attack dialogs. For these reasons, and because it does not include complete IMS signaling paths with intermediate network functions, we do not use the dataset of Nassar et al. [13] in this work. Instead, we rely on a dataset captured under real multi-operator conditions and enriched with expert ground truth, thus providing a realistic benchmark unavailable in previous SIP studies.

IV. METHODOLOGY

To address the challenges outlined in Section II, we introduce SIP-Classifer, an unsupervised methodology specifically designed for classifying signaling sequences of SIP in IMS. Figure 1 illustrates the architecture of the SIP-Classifer, which comprises four main components: Tokenizer ①, Transformer ②, Feature Reduction ③, and Clustering Analysis ④. The initial phase of the methodology involves tokenizing (i.e., dividing and converting) the SIP sequence into a symbolic representation that can be effectively processed by the Transformer. While different tokenizations can be used, we focus here on an intuitive but powerful one based on the core elements of the SIP language ①. Subsequently, a Transformer model is trained on the tokenized training sequences using a self-supervised learning approach, which does not require labeled sequence classes. The model learns to capture internal structure and dependencies within the data, from which representative features are subsequently extracted ②.

Then, to enhance computational efficiency and reduce feature dimensionality, PCA is employed to compress the extracted features while preserving their information ③. Finally, upon the reduction of features to the most significant ones, the K-Means++ algorithm is applied to identify clusters that contain different scenarios of SIP dialogs in IMS ④. In the following, we present each of these components in detail. However, as reported in Section V, we experimentally observed that changing the tokenization scheme, transformer architecture, or clustering algorithm does not significantly hinder performance, making our pipeline robust to specific implementations and technological advancements.

A. Tokenizer

The first component of our methodology converts the signaling sequences into symbolic representations that a Transformer can digest. Tokenization is the process of splitting a sequence into individual units. All distinct individual units in the training set are used to create a vocabulary of tokens that maps each token (plus special tokens for padding, beginning- and end-of-sequence, and unknown units) to a numeric index. A component called Tokenizer is then used to transform a given sequence into its corresponding list of token indexes.

The choice of the tokenization strategy, i.e., how we define the individual units of a sequence, is not straightforward, and each possible approach has its pros and cons in terms of vocabulary dimension, handling of uncommon situations, and intuitiveness. A signaling SIP sequence comprises a series of colon-separated triplets, each composed of a source network, a destination network, and a message with an optional numeric code. Each triplet represents a single logical timestep, similar to a word in a sentence, and thus might seem the most intuitive choice for individual units. However, working at the triplet level would make the vocabulary huge, causing the Transformer to be larger and the computation slower, and the chance of encountering an unknown (i.e., not present in the training set) triplet at inference time higher. On the other hand, decomposing each triplet into its three or four building elements would make the vocabulary smaller and the probability of unknown tokens lower (since an unknown triplet might be made of known elements), while maintaining its intuitiveness and simplicity, as it leverages known properties and patterns without introducing any additional complexity. Because of this, our token vocabulary $\mathcal{V}$ contains all network functions, messages, and codes available in the training set, and our element-wise Tokenizer simply divides the sequence based on commas, colons, and hyphens and converts each of them into its corresponding token index from $\mathcal{V}$ (see Figure 2).

B. Transformer

The second component of our methodology is a decoder-only Transformer [29]. While our approach does not explicitly target the generation of new signaling sequences, we decided to adopt a generative, decoder-based, autoregressive model to better capture the sequential nature of our domain. In this way, we can learn a model of the SIP dialogs distribution without the need for labeled data (necessary for supervised

learning) and use the extracted, latent features for unsupervised or supervised (but more data-efficient) tasks. Since our goal is to learn how to effectively represent possible SIP dialogs, we use as many sequences as possible (including unlabeled ones) during training, in order to map a broader spectrum of potential, syntactically correct SIP sequences.

The architecture of our model follows that of the open-source LLaMa family [15] and is summarized in Figure 3. First, the input tokenized sequence (i.e., a list of token indexes) is converted into a sequence of token embeddings of size H (through a learnable lookup table), and a position encoding is added to provide information on the temporal location of the token. Then, a series of L Transformer block layers is applied, where each block returns hidden states of size H and contains layer normalization, multi-head self-attention, a feed-forward layer, and residual connections. Finally, the output of the last layer is linearly transformed into a vector of length $|\mathcal{V}|$, and a softmax is applied to get the probability distribution of the next token. To learn the correct next-token distribution, the θ -parameterized Transformer is trained via stochastic gradient descent to maximize the log-probability of training data Y :

$$\mathcal{L}(\theta) = \mathbb{E}_{y \sim Y} \left[\sum_{t=1}^T -\log p_{\theta}(y_t | y_{t-1} \dots y_1) \right].$$

In other words, the Transformer learns to predict the next token given previous ones. However, the concatenation of all hidden layers' outputs can be seen as a compact, latent representation of the input sequence, ideally encoding all important information. To get a feature vector for each sequence, we simply extract the output of all Transformer layers while the model is predicting the $\langle \text{endoftext} \rangle$ token, which should include information about all the input tokens before it and therefore about the entire sequence, obtaining a latent vector of size $L \cdot H$ for each sequence.

C. Feature Reduction

After obtaining the representative features of all the sequences in the dataset using the Transformer, these are reduced through the Principal Component Analysis technique. The application of PCA is intended to prevent the "curse of dimensionality" [30] and reduce the redundancy of the data, thus improving their representation and enhancing the model performance [31]. PCA aims to identify the directions of maximum variance within the high-dimensional data and subsequently project them into a new feature space with a smaller number of dimensions. The axes of the new feature space, which are perpendicular to each other (orthogonal axes), can be interpreted as the directions of maximum variance. In order to perform PCA [16], it is necessary to standardize the data set, i.e., to ensure that all variables have the same weight. The data is standardized by subtracting the mean and dividing by the standard deviation. Once the data has been standardized, PCA can be applied, which involves constructing a covariance matrix that describes how the variables in the dataset are related to each other. The subsequent step involves the spectral decomposition of the matrix into its eigenvalues and eigenvectors, denoted by λ and v , respectively. The

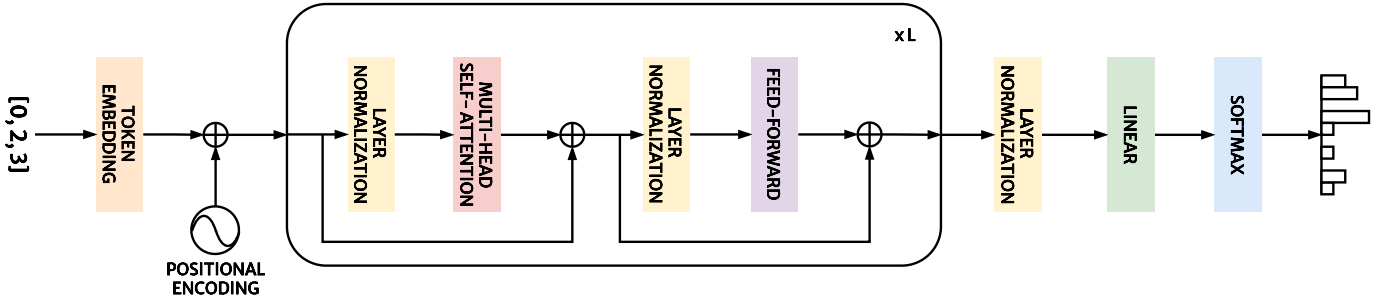


Fig. 3. A schematic representation of the full decoder-only transformer architecture to autoregressively model signaling sequences.

eigenvalues represent the amount of variance explained by each principal component, while the eigenvectors define the directions of the principal components. The eigenvectors are then sorted according to the corresponding eigenvalues, which are ordered in a decreasing manner ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$). The new variables are obtained by projecting the data onto the directions of the selected eigenvectors. Once the principal components have been selected and ordered according to their eigenvalues, the cumulative explained variance for each principal component is calculated. The cumulative variance is the sum of the variances explained by all the principal components up to the i -th component [32]. The cumulative variance for component i can be expressed as:

$$\sum_{j=1}^i \frac{\lambda_j}{\sum_{j=1}^n \lambda_j}.$$

D. Clustering

The final component of our methodology aims to group the refined latent representations into cohesive, semantically meaningful categories. To automatically discover the underlying behavioral patterns of the network traffic without relying on ground-truth labels, we employ unsupervised clustering. By partitioning the latent space based on geometric proximity, we can isolate distinct SIP scenarios, naturally separating legitimate calling behaviors from anomalous or malicious deviations. Specifically, we apply the K-Means++ algorithm [33] to identify these different SIP scenarios. This algorithm is a variant of the K-means algorithm designed to improve the initial choice of centroids, thus leading to faster convergence and better quality clusters. In contrast to the original K-means algorithm, which starts with a random choice of centroids, K-Means++ uses an initialization method that places initial centroids well apart. This approach is intended to mitigate the likelihood of suboptimal outcomes arising from an inadequate selection of initial centroids, which can result in the algorithm converging on a suboptimal local minimum. To avoid an erroneous selection of centroids, this algorithm, following the random selection of the initial centroid from the dataset $D = \{X_i\}_{i=1}^n$, calculates for each data point X_i in the dataset the probability $P(X_i)$ of being selected as the subsequent centroid, according to the following formula:

$$P(X_i) = \frac{D(X_i)^2}{\sum_{j=1}^n D(X_j)^2}$$

where $D(X_i)$ represents the minimum distance between point X_i and the already selected centroids:

$$D(X_i) = \min_{1 \leq k \leq m} \|X_i - c_k\|^2.$$

In particular, c_k denotes one of the pre-selected centroids, and m denotes the total number of centroids selected up to this point. The process is repeated until the desired number of K initial centroids, $c_1, c_2, \dots, c_k$, is obtained.

Subsequently, following the selection of the initial centroids, the standard K-means algorithm [34] is employed following two steps:

- 1) Cluster assignment: each X_i point is to be assigned to the nearest cluster s_j :

$$s_j = \{X_i : \|X_i - c_j\|^2 = \min_{1 \leq k \leq K} \|X_i - c_k\|^2\}.$$

- 2) Centroid update: the centroids c_j are to be recalculated as the average of the points belonging to cluster s_j :

$$c_j = \frac{1}{|s_j|} \sum_{X_i \in s_j} X_i.$$

These two steps are repeated until the cluster assignment ceases to undergo alteration, or until the stipulated tolerance or maximum number of iterations is reached.

V. EVALUATION

This section evaluates SIP-Classifier's overall efficacy to assess whether it can be applied to real-world SIP scenarios for detecting anomalous calls. In particular, we focus on empirically answering the following questions:

- Q1 How effective is SIP-Classifier at distinguishing valid SIP signaling sequences from anomalous ones?
- Q2 How relevant is the choice of the Transformer architecture?
- Q3 What is the impact of the clustering algorithm choice on the quality and class consistency of the obtained clusters?
- Q4 Is there a better tokenization method than our custom, element-wise scheme?
- Q5 How does SIP-Classifier perform compared with state-of-the-art methods?

TABLE V
TRANSFORMER PARAMETERS

Parameter	Value
Vocabulary size	59
Embedding and hidden state size H	512
Feed-forward layers size	2048
Number of hidden layers L	4
Number of attention heads per layer	8
Model size (number of parameters)	16842240
Dropout rate	0.1
Batch size	16
Epochs	13
Warmup steps	10
Optimizer	AdamW
Learning rate	$5e-4$
Weight decay	0.1
Gradient clipping value	1.0
Learning rate scheduler type	Cosine

A. Experimental Setup

1) *Dataset and Tokenization*: We validated our method over the dataset detailed in Section III. First, we defined the Tokenizer by building a vocabulary with one token for each distinct element as described in Section IV-A, plus two special tokens (one for test-time unknown tokens and one for delimiting and padding sequences), yielding a total of 59 tokens. Then, we divided the signaling sequences into three subsets: training, validation, and test set. To preserve a sufficient amount of unseen, labeled sequences, we retained 1,000 distinct sequences (700 valid and 300 anomalous) as our test set. Then, we extracted another 1,000 distinct sequences (preserving the proportion of labels with respect to the training set) for our validation set, and we used all remaining 22,385 sequences (9,329 without duplicates) as our training set. This split strategy prevents any exact-sequence leakage across subsets, ensuring a fair validation- and test-time evaluation.

2) *Transformer Setup*: We used the training set to train our Transformer, and the validation set to select the most appropriate architectural and learning parameters. We conducted extensive experiments varying the embedding and state size (between 512 and 768), the number of hidden layers (4 or 8), the number of attention heads per layer (8 or 12), the batch size (among 16, 32, and 64), the learning rate ($5e-4$ or $1e-4$), and the dropout rate (either 0.1 or no dropout) while implementing early stopping to select the best-performing epoch; the other hyperparameters have been set according to common best practices for Transformers. We selected the setting with the best performance (reported in Table V), as indicated by the validation loss, and used it in the subsequent stages of our pipeline. However, it is important to highlight that several different configurations achieved very similar validation loss, suggesting that the choice of hyperparameters does not play a significant role here, and our approach can be adopted without the need to run extensive and expensive hyperparameter-tuning experiments.

3) *Dimensionality Reduction*: To obtain a more compact and less sparse latent representation, we first standardized the latent features to have zero mean and unit variance, and then applied PCA. Rather than fixing the dimensionality a priori, we explored different PCA truncation levels by retaining the leading principal components from the minimum number required to explain at least 80% of the cumulative variance up to the number of components explaining 95% of the variance. We introduced the lower bound to prevent overly compressed representations from missing the variability of unseen sequences. Conversely, we adopted the upper bound to preserve most information while substantially reducing dimensionality.

4) *Clustering Configuration for Anomaly Detection*: On top of these reduced representations, we performed a grid search using K-Means++ to identify the optimal combination of retained principal components and number of clusters K . In our anomaly-detection setting, we restricted clustering configuration and evaluation to sequences associated with a reliable binary ground truth, namely *valid* and *anomalous* dialogs. We excluded sequences labeled as *unknown* from both the model-selection stage and the final test evaluation, since the *unknown* label is not directly compatible with the adopted binary formulation of the task. Including such samples would have introduced an ambiguous target definition, as they cannot be consistently mapped to either the valid or anomalous class required by binary detection metrics.

Specifically, we explored a two-dimensional search space. The first dimension varied the number of retained PCA components, spanning the 80%–95% cumulative explained-variance range defined above. To reduce computational cost, when this interval contained more than 20 candidate dimensionalities, we evaluated every second PCA dimension; otherwise, all integer values in the interval were considered. The second dimension varied the number of clusters. To preserve the unsupervised nature of the experiment, we selected the optimal model under a “blind” setting, without access to ground-truth labels. For the cluster dimension, we explored an even number of clusters ranging from 2 up to the theoretical upper bound of $K_{\max} \approx \sqrt{N/2} \approx 84$ (where $N = 14,253$ is the number of training sequences considered for clustering after excluding *unknown* dialogs), following a widely adopted rule of thumb for cluster analysis [35]. Rather than fixing the number of clusters *a priori*, we selected it by maximizing internal validation criteria. More specifically, we evaluated each candidate configuration using the Silhouette score and the Calinski-Harabasz index. We then aggregated their rankings into a consensus score and selected the highest-ranked configuration. According to this procedure, the optimal configuration retained 11 principal components and partitioned the data into 84 clusters. We revealed ground-truth labels only after this purely geometric selection step. We then evaluated the selected model on the anomaly-detection task using Accuracy, Precision, Recall, and F1-score.

B. Detection Performance (Q1)

This section assesses the SIP-Classifier’s ability to discriminate between valid and anomalous SIP signaling sequences.

TABLE VI

DETECTION PERFORMANCE OF SIP-CLASSIFIER IN TERMS OF ACCURACY, PRECISION, RECALL, AND F1-SCORE. RESULTS ARE REPORTED FOR BOTH THE TRAINING AND TEST SETS

Dataset	Accuracy	Precision	Recall	F1-score
Training	0.99	1.00	0.99	0.99
Test	0.98	0.98	0.95	0.96

We evaluate its detection performance using four standard metrics:

- **Accuracy:** measures the overall proportion of correctly classified instances across the entire dataset.
- **Precision:** evaluates the proportion of correctly identified positive instances among all positive predictions.
- **Recall:** assesses the proportion of actual positive instances that are correctly recognized by the model.
- **F1-Score:** computes the harmonic mean of precision and recall, providing a balanced assessment of the detection quality.

Although we selected the clustering model without exploiting label information, we used the original labels provided with the dataset (see Section III) as ground truth for the *ex-post* computation of accuracy, precision, recall, and F1-score, considering anomalous sequences as the positive class. To compute these metrics, each cluster was mapped to a binary label through majority voting over the labeled training samples assigned to it.

Table VI reports the detection performance of SIP-Classifier on both the training and test sets. Overall, the proposed combination of Transformer-based feature extraction and clustering achieves very high classification performance, with all the considered metrics above 0.95 on the test set. In particular, the model reaches 0.98 accuracy and 0.98 precision, showing that the resulting clusters are highly consistent with the ground-truth distinction between valid and anomalous SIP signaling sequences. Comparing training and test results also suggests that the method generalizes well. Although the performance on the training set is slightly higher, the slight performance drop on the test set indicates that the learned latent representation preserves the relevant structural characteristics of the sequences also on unseen data. Recall scores the lowest value on the test set, equal to 0.95, meaning that a small fraction of anomalous sequences is still classified as valid. Nevertheless, the corresponding F1-score of 0.96 confirms that the overall balance between precision and recall remains very strong.

C. Comparative Evaluation of Transformer Architectures (Q2)

In addition to our LLaMa architecture, we also experimented with two alternative Transformer architectures. First, we tested a GPT-like model [36], [37], which deviates from LLaMa for small implementation choices such as activation function and positional embeddings, and we found no significant differences in terms of validation loss. Moreover, once embedded in our pipeline, it yields worse scores under the same tokenization scheme (and almost identical scores under

different schemes, as reported in Section V-E) despite its best configuration having many more parameters (and a smaller batch size, which requires longer training time), suggesting that our method is resilient to small architectural changes and merely scaling up the model does not provide benefits. Then, we tried with a RoBERTa-like model [38], which instead deviates more radically from LLaMa, as it is an encoder-based Transformer trained with a masked language modeling objective where it learns to predict randomly-masked tokens given the context, i.e., the tokens on the left and right sides of it (while GPT and LLaMa are trained to predict the next token given only what precedes it). Although this makes it impossible to compare the models based on the validation loss, we found that once embedded in our full pipeline, RoBERTa yields competitive, but not better, results, despite a remarkably larger model. Table VII reports the classification performance of our pipeline with all three architectures and their best hyperparameter configurations in terms of validation loss.

D. Comparative Evaluation of Clustering Methods (Q3)

To validate the choice of K-Means++ for the proposed SIP-Classifier pipeline, we compared it against DBSCAN [39] on the Transformer-based latent representations. We evaluate the clustering quality using the following metrics:

- **Silhouette Score:** measures intra-cluster cohesion versus inter-cluster separation.
- **Calinski-Harabasz index:** evaluates the ratio of inter-cluster variance to intra-cluster variance, suitable for centroid-based methods.
- **Density-Based Clustering Validation (DBCV):** assesses density connectedness and separation, designed for density-based algorithms.
- **Homogeneity:** a validation metric measuring cluster purity with respect to ground-truth labels.

For DBSCAN, we adopted an unsupervised model-selection strategy analogous to K-Means++. We performed a grid search varying the retained PCA components, *MinPts*, i.e., the minimum number of points required for a region to be considered dense, and ϵ , i.e., the radius defining the local neighborhood around each point. For *MinPts*, we considered a range in [5, 70]. The small lower bound accommodates minor legitimate format variations [40] and rare anomalous sequences, preventing them from being dismissed as noise. The upper bound prevents the over-smoothing of macroscopic dense regions representing the bulk of the SIP traffic. For each *MinPts*, we dynamically estimated candidate ϵ values using k-nearest-neighbor distances [41].

The optimal DBSCAN configuration maximized a consensus score aggregating the Silhouette Score and the DBCV index, penalized for excessive noise ratios. Since anomalous sequences constitute a large portion of our dataset, they are expected to form distinct topological structures rather than scattered outliers. Therefore, penalizing excessive noise ensures the algorithm effectively captures these anomalous patterns. As with K-Means++, we strictly reserved homogeneity for *ex-post* evaluation.

TABLE VII

CLASSIFICATION PERFORMANCE OF OUR PIPELINE WITH DIFFERENT TRANSFORMER ARCHITECTURES. EACH ARCHITECTURE EMPLOYS ITS BEST HYPERPARAMETER SETTING IDENTIFIED THROUGH EXTENSIVE ABLATION STUDIES. FOR EACH ARCHITECTURE, THE OVERALL MODEL SIZE (AS THE TOTAL NUMBER OF PARAMETERS), THE TRAINING TIME (AS THE SECONDS REQUIRED TO COMPLETE THE FULL TRAINING), AND THE CLASSIFICATION SCORES ON THE TEST SET ONCE EMBEDDED IN OUR PIPELINE ARE REPORTED

Model	Model size	Train time	Accuracy	Precision	Recall	F1-Score	Homogeneity
LLaMa	16.8M	736.9 sec	0.98	0.98	0.95	0.96	0.89
GPT	25.7M	1479.6 sec	0.93	0.90	0.87	0.88	0.74
RoBERTa	58.0M	2065.5 sec	0.97	0.95	0.95	0.95	0.88

TABLE VIII

CLUSTERING PERFORMANCE COMPARISON. WE REPORT THE SELECTED PCA DIMENSIONALITY, CLUSTERING PARAMETERS, INTERNAL VALIDATION SCORES USED FOR UNSUPERVISED SELECTION, AND THE EX-POST HOMOGENEITY SCORE

Clustering Method	PCA	Number of Clusters	ϵ	MinPts	Noise	Calinski–Harabasz	DBCV	Silhouette Score	Homogeneity
K-Means++	11	84	N/A	N/A	N/A	18019.61	N/A	0.71	0.96
DBSCAN	36	89	12.31	5	0.04	N/A	0.52	0.40	0.91

TABLE IX

THE DIFFERENT TYPES OF TOKENIZATION CONSIDERED IN OUR EXPERIMENTS

Tokenization Type	Example	Description
Element (ours)	SCSCF,MTAS,A:ICSCF,GTW,I-487	Described in Section IV-A.
Element with zero-padding	SCSCF,MTAS,A-0:ICSCF,GTW,I-487	Same as Element tokenization, but appends a ‘0’ to messages without a response code, making sequences more regular.
Triplet	SCSCF,MTAS,A:ICSCF,GTW,I-487	Divides each sequence in its temporal communication event to better capture structural patterns.
Triplet + BPE	SCSCF,MTAS,A:ICSCF,GTW,I-487	Applies Byte-Pair Encoding (BPE; [42]) to the triplet-divided sequences. The vocabulary is iteratively composed by starting from a token for each single character inside any triplet; at each step, the pair of consecutive tokens with the highest absolute frequency in the dataset is added as a new token, and the process continues until a fixed limit is reached. This ensures a limited size while acknowledging the most common patterns without increasing the chance of unknown tokens at inference time.
Triplet + WordPiece	SCSCF,MTAS,A:ICSCF,GTW,I-487	Similar to Triplet + BPE, but applies WordPiece algorithm [43] instead: at each step it adds the pair of tokens with the highest relative frequency (with respect to the frequencies of the two tokens alone), and differentiates between a starting token and a non-starting token.

Table VIII reports the selected configurations. K-Means++ proves to be the most effective method for capturing the structural properties of SIP signaling, achieving better overall clustering performance. In addition, it ensures higher class consistency, requires a more compact latent representation, and yields a more interpretable clustering structure.

Overall, K-Means++ emerges as the preferable solution for the SIP-Classifier pipeline, offering a superior trade-off among clustering quality, class consistency, compactness, and interpretability of the latent space. Notably, the selected K-Means++ solution lies at the upper boundary of the initial cluster search range. To verify that this result was not an artifact of the imposed cap, we extended the search up to $K = 100$. This additional analysis revealed a plateau in the best Silhouette values, with only minor local fluctuations and no substantial gain beyond $K = 84$.

E. Comparative Evaluation of Tokenization Schemes (Q4)

To ensure that the tokenization strategy adopted for the Transformer model was the most suitable for modeling SIP

sequences, we conducted a comparative evaluation using alternative tokenization methods. For each tokenization variant, as reported in Table IX, we repeated the full ablation study on the Transformer architecture, selected the best model according to its validation loss, embedded it in our pipeline, and computed standard evaluation metrics (accuracy, precision, recall, and F1-Score), following the methodology described in Section V-B. The results obtained, presented in Table X, show that the best performing combination of Transformer architecture, tokenization scheme, and clustering method makes use of the tokenization approach proposed in our methodology. In addition, it also outperforms alternative methods when the RoBERTa architecture is adopted, while achieving competitive, but sub-optimal results for the other configurations.

In particular, introducing a padding response code of 0 for messages without a response code to make sequences more regular does not provide significant benefits and can instead deteriorate performance in certain scenarios. On the other hand, considering each different triplet as a separate token leads to more variable and globally worse performance.

TABLE X

CLASSIFICATION PERFORMANCE OF OUR PIPELINE WITH DIFFERENT TOKENIZATION SCHEMES. FOR EACH TRANSFORMER ARCHITECTURE AND CLUSTERING METHOD, ACCURACY, PRECISION, RECALL, AND F1-SCORE ON THE TEST SET ARE REPORTED

Transformer	Clustering	Tokenization	Accuracy	Precision	Recall	F1-Score
LLaMa	K-Means++	Element	0.98	0.98	0.95	0.96
		Element with zero-padding	0.97	0.97	0.92	0.94
		Triplet	0.96	0.91	0.96	0.94
		Triplet + BPE	0.98	0.96	0.96	0.96
		Triplet + WordPiece	0.98	0.98	0.94	0.96
	DBSCAN	Element	0.88	0.77	0.83	0.80
		Element with zero-padding	0.94	0.83	0.99	0.90
		Triplet	0.90	0.75	0.99	0.86
		Triplet + BPE	0.91	0.76	1.00	0.86
		Triplet + WordPiece	0.91	0.77	0.99	0.87
GPT	K-Means++	Element	0.93	0.90	0.87	0.88
		Element with zero-padding	0.93	0.90	0.87	0.88
		Triplet	0.96	0.96	0.92	0.94
		Triplet + BPE	0.97	0.98	0.93	0.95
		Triplet + WordPiece	0.96	0.97	0.91	0.94
	DBSCAN	Element	0.93	0.82	0.98	0.89
		Element with zero-padding	0.92	0.78	1.00	0.88
		Triplet	0.85	0.94	0.55	0.69
		Triplet + BPE	0.93	0.81	1.00	0.89
		Triplet + WordPiece	0.93	0.80	1.00	0.89
RoBERTa	K-Means++	Element	0.97	0.95	0.95	0.95
		Element with zero-padding	0.93	0.94	0.80	0.87
		Triplet	0.96	0.89	0.98	0.93
		Triplet + BPE	0.95	0.86	0.98	0.92
		Triplet + WordPiece	0.91	0.94	0.75	0.83
	DBSCAN	Element	0.88	0.72	0.99	0.83
		Element with zero-padding	0.90	0.76	0.98	0.85
		Triplet	0.91	0.89	0.80	0.84
		Triplet + BPE	0.91	0.87	0.80	0.83
		Triplet + WordPiece	0.89	0.82	0.82	0.82

Finally, adopting state-of-the-art techniques such as WordPiece [43] or Byte-Pair Encoding (BPE; [42]), while optimal for large language models, does not provide significant enhancements with respect to our domain-based approach: they contribute to small improvements in specific configurations (e.g., when using a GPT architecture or the DBSCAN clustering algorithm), while being barely outperformed in others, including the optimal one. In summary, the competitive performance of our element-wise tokenization scheme, together with its smaller vocabulary and its use of a more intuitive and interpretable approach, confirms its appropriateness and effectiveness.

F. Comparison With State-of-the-Art Methods (Q5)

To position *SIP-Classifier* with respect to prior SIP-dialog analysis approaches, we compared it against four representative baselines from the literature: an HMM Bayesian model enriched with n-gram Markov observations [11], an LSTM model [25], a CNN model [26], and a correlation-based classifier [12]. We also included a Transformer Encoder baseline, introduced in this study to assess whether adopting a more recent neural architecture alone is sufficient to address the problem. All baselines were reevaluated on the same real-world SIP-IMS dataset and split adopted for *SIP-Classifier*: training on benign dialogs only and evaluation on the same held-out test set used throughout the paper, which contains 700 *valid* and 300 *anomalous* dialogs.

Concretely, training used 7,369 benign dialogs; for the neural baselines, this set was further split into 5,895 dialogs for model fitting and 1,474 for validation. For the correlation-based baseline, duplicates were removed as prescribed by the original method, yielding a benign knowledge base of 1,772 unique dialogs. To remain faithful to the original formulations, the baselines were evaluated on a message-level abstraction of each dialog rather than on the full IMS-aware triplet representation used by *SIP-Classifier*. Specifically, from each hop we retained only the SIP method or response code contained in the third field of the triplet, while discarding source and destination network-function identifiers. This mirrors the representation assumed by the compared methods, which model SIP dialogs primarily as sequences of message tokens rather than as full IMS routing paths.

Whenever a paper proposed multiple model variants, we reimplemented the alternatives described by the authors and, for brevity, Table XI reports only the strongest-performing variant under the described evaluation settings. In particular, the LSTM model corresponds to the best of the reimplemented LSTM variants from [9], [25], namely a two-layer model with 256 hidden units per layer, Dropout rate of 0.5, and a final softmax classifier, combined with anomaly detection through thresholding of the maximum softmax score on benign training dialogs. For the correlation-based method, we evaluated both the normalized and non-normalized variants and retained the normalized one, which performed better on our data.

TABLE XI

COMPARISON WITH STATE-OF-THE-ART METHODS ON THE COMMON 1,000-DIALOG TEST SET. EACH ENTRY REPORTS THE POINT ESTIMATE FOLLOWED BY ITS 95% PERCENTILE BOOTSTRAP CONFIDENCE INTERVAL IN BRACKETS ($B = 1,000$ RESAMPLES)

Method	Accuracy	Precision	Recall	F1-score
HMM [11]	0.77 [0.74, 0.79]	0.81 [0.74, 0.89]	0.29 [0.24, 0.34]	0.43 [0.37, 0.48]
LSTM [25]	0.44 [0.40, 0.47]	0.32 [0.28, 0.35]	0.76 [0.71, 0.81]	0.45 [0.40, 0.48]
CNN [26]	0.44 [0.41, 0.47]	0.35 [0.32, 0.38]	1.00 [1.00, 1.00]	0.52 [0.48, 0.55]
Normalized Correlation [12]	0.73 [0.71, 0.76]	1.00 [1.00, 1.00]	0.11 [0.07, 0.14]	0.19 [0.13, 0.25]
Transformer Encoder	0.44 [0.41, 0.47]	0.34 [0.31, 0.37]	0.93 [0.90, 0.96]	0.50 [0.46, 0.53]
SIP-Classifier (ours)	0.98 [0.97, 0.99]	0.98 [0.96, 0.99]	0.95 [0.92, 0.97]	0.96 [0.95, 0.98]

A further methodological point deserves clarification: the compared baselines were originally proposed under partially different formulations of the problem. In particular, the LSTM and CNN methods [25], [26] were designed to distinguish previously seen dialogs from unseen ones, rather than to directly perform binary valid-versus-anomalous classification. The HMM-based method [11] and the correlation-based classifier [12] also rely on benign/known signaling patterns learned from training data and identify anomalies through a rejection rule at inference time. To enable a consistent comparison on our task, we therefore evaluated all baselines under a common binary anomaly-detection protocol: acceptance of a test dialog was mapped to the *valid* class, whereas rejection was mapped to the *anomalous* class. For the Transformer Encoder baseline introduced in this work, the anomaly decision was likewise obtained by thresholding its maximum softmax score on benign training dialogs. This common adaptation allows all baselines to be compared consistently on the same binary anomaly-detection task.

To quantify the uncertainty of the individual performance estimates associated with the finite test set, we computed non-parametric 95% percentile bootstrap confidence intervals for all reported metrics. For each method, we generated $B = 1,000$ bootstrap samples by resampling the 1,000 test dialogs with replacement and recomputed accuracy, precision, recall, and F1-score on each sample. The empirical 2.5th and 97.5th percentiles were used as the bounds of the 95% confidence intervals. The anomalous class was treated as the positive class.

Table XI shows that SIP-Classifier provides the strongest overall balance between anomaly detection and false-alarm control. It achieves an accuracy of 0.98 (95% CI: 0.97–0.99) and an F1-score of 0.96 (0.95–0.98). By comparison, the strongest baseline in terms of F1-score is the CNN, which reaches 0.52 (0.48–0.55), while the HMM obtains the highest baseline accuracy, 0.77 (0.74–0.79). The results also highlight the markedly different error profiles of the baselines. The normalized-correlation method obtains perfect precision by issuing very few anomalous predictions, but its recall is only 0.11. Conversely, the CNN obtains perfect recall but has a precision of only 0.35, resulting in a large number of false alarms. Similarly, the LSTM and Transformer Encoder exhibit high recall but low precision, indicating that they identify most anomalous dialogs while also rejecting many valid dialogs. SIP-Classifier instead maintains both high precision, 0.98 (0.96–0.99), and high recall, 0.95 (0.92–0.97), explaining its substantially higher F1-score. The low precision observed for

the LSTM and CNN baselines is consistent with the mismatch between the smaller, synthetic, and controlled dataset used in their original validation and the substantially greater complexity of our operational dataset. Their rejection mechanisms successfully identify many anomalous dialogs, but also reject a substantial number of valid dialogs characterized by more heterogeneous SIP procedures, much longer sequences, and signaling paths involving multiple network functions, whose roles are not represented by these baselines. Conversely, the high precision and low recall of the correlation-based method indicate a substantially more conservative rejection policy. These results therefore characterize the limitations of transferring their original formulations to heterogeneous real-world SIP-IMS anomaly-detection tasks, rather than their effectiveness under the experimental settings for which they were originally proposed.

Since all methods were evaluated on the same test dialogs, we performed pairwise two-sided McNemar tests between SIP-Classifier and each baseline. For each pair, we constructed a 2×2 contingency table of classification correctness, where n_{11} denotes dialogs correctly classified by both methods, n_{10} dialogs correctly classified only by SIP-Classifier, n_{01} dialogs correctly classified only by the baseline, and n_{00} dialogs misclassified by both. The continuity-corrected statistic is

$$\chi_{cc}^2 = \frac{(|n_{10} - n_{01}| - 1)^2}{n_{10} + n_{01}}. \quad (1)$$

The statistic depends on the discordant cells n_{10} and n_{01} , while all contingency counts are reported for completeness. Raw p -values were adjusted using Holm's procedure across the five comparisons, with $\alpha = 0.05$. As shown in Table XII, n_{10} is substantially larger than n_{01} in every comparison. For example, SIP-Classifier correctly classifies 224 dialogs that the HMM misclassifies, whereas the HMM correctly classifies only 13 dialogs missed by SIP-Classifier. Against the CNN, the corresponding counts are 556 and 17. The paired accuracy gains range from 21.10 to 54.20 percentage points, and all comparisons remain significant after Holm correction, with adjusted p -values between 3.60×10^{-114} and 2.29×10^{-42} . We finally highlight that McNemar's test assesses paired classification correctness rather than differences in precision, recall, or F1-score, whose uncertainty is instead quantified by the bootstrap intervals.

Overall, these results support one of the main claims of the paper: the evaluated methods, originally developed and validated on synthetic or simplified SIP-dialog benchmarks, do not transfer effectively to the operational SIP-IMS setting

TABLE XII

COMPLETE PAIRWISE MCNEMAR COMPARISONS BETWEEN SIP-CLASSIFIER AND THE BASELINES ON THE COMMON 1,000-DIALOG TEST SET. THE TEST IS TWO-SIDED AND CONTINUITY-CORRECTED; RAW AND HOLM-ADJUSTED p -VALUES ARE REPORTED

Baseline	n_{11}	n_{10}	n_{01}	n_{00}	$\Delta\text{Acc (pp)}$	χ^2_{cc}	Raw p	Holm p
HMM [11]	754	224	13	9	+21.10	186.08	2.29×10^{-42}	2.29×10^{-42}
LSTM [25]	425	553	11	11	+54.20	518.94	7.21×10^{-115}	3.60×10^{-114}
CNN [26]	422	556	17	5	+53.90	505.14	7.25×10^{-112}	2.90×10^{-111}
Normalized Correlation [12]	724	254	8	14	+24.60	229.10	9.35×10^{-52}	1.87×10^{-51}
Transformer Encoder	423	555	18	4	+53.70	501.39	4.74×10^{-111}	1.42×10^{-110}

considered here. In this setting, dialogs traverse multiple intermediate network functions and exhibit substantially richer structural variability. Therefore, the results indicate that the main limitation is not merely the choice of classifier, but the underlying problem formulation itself. Replacing LSTM or CNN blocks with a more recent Transformer encoder is not sufficient when the input representation remains restricted to message-only SIP patterns and does not model the complete sequence of interactions among IMS network functions.

VI. DISCUSSION

The evaluation of the SIP-Classifier demonstrates the effectiveness of combining unsupervised representation learning with clustering techniques to analyze SIP signaling flows in IMS environments. The model achieves high accuracy in distinguishing valid and anomalous sequences, highlighting the robustness and practical applicability of the approach in real-world scenarios where data uncertainty and limited label reliability are common.

A. Scalability and Real-Time Applicability

SIP-Classifier currently suits post-call or near-line analysis better than strict real-time monitoring, because the framework processes each SIP dialog only after receiving the full signaling sequence. It operates by encoding each complete dialog with a decoder-only autoregressive Transformer, projecting the representation through PCA, and assigning it to a cluster. Thus, it acts as a batch-oriented anomaly analysis pipeline rather than a per-message online detector.

Extracting features with the Transformer dominates the computational cost, whereas the downstream stages remain lightweight. On a held-out test set of 1,000 previously unseen sequences, extracting features on a single NVIDIA L40S GPU averaged 42.6 ± 3.3 s over 10 runs. By contrast, projecting via PCA and assigning clusters on an AMD Ryzen 7000 CPU averaged just 39.8 ± 4.3 ms. This demonstrates that the Transformer stage drives the inference time, while PCA and clustering add negligible overhead. In a batch setting, the framework processes sequences at an average normalized cost of 42.6 ms per sequence, without requiring distributed infrastructure.

This corresponds to a throughput of approximately 23.5 dialogs per second on the considered hardware. The reported per-sequence value represents an amortized cost under batch inference and should not be interpreted as the latency of an isolated online request. Since completed dialogs can be processed

independently, larger operational workloads could be handled through GPU batching, micro-batching, or multiple parallel inference workers. However, the effective processing capacity also depends on dialog length, since the computational and memory requirements of standard self-attention increase with the length of the tokenized sequence.

Using PCA does not hinder deployment because we train it offline on Transformer embeddings extracted from the training set. During inference, the framework applies the stored normalization parameters and projection matrix to each new dialog, avoiding the need to recompute the covariance matrix on incoming traffic. Similarly, we train K-Means++ offline, so inference only requires assigning the nearest centroid in the reduced latent space. A main limitation is that the current framework does not yet continuously adapt the projection basis and cluster centroids under streaming traffic and concept drift.

This limitation does not affect the static evaluation reported in this work, but it may reduce the effectiveness of both near-line and online deployments over extended operational periods. Since SIP and the main IMS signaling procedures are standardized and evolve relatively slowly, the structural drift of the protocol itself is expected to be limited. Nevertheless, operational changes in network topology, routing policies, software implementations, service configurations, traffic composition, or failure conditions may alter the distribution of observed signaling dialogs. In practice, drift could be monitored through persistent changes in the Transformer likelihood distribution, nearest-centroid distances, or cluster occupancy. Such changes could trigger periodic retraining or controlled updates of the dimensionality-reduction and clustering components.

While SIP-Classifier currently serves as a near-line anomaly detection tool, the decoder-only autoregressive Transformer allows future extensions for earlier detection. Because we train the model to predict the next token from the observed prefix, future iterations can exploit partial signaling prefixes to predict anomalies before a call terminates. Combining this approach with lighter sequence models, more efficient attention mechanisms, cached embeddings, and incremental clustering strategies will move the framework closer to online deployment.

However, prefix-level decision criteria, the stability of successive predictions, and the trade-off between detection delay and false alarms require a separate evaluation.

B. Limitations and Generalizability

As a limitation of this study, we empirically evaluated our approach on a single, albeit comprehensive and realistic,

SIP-IMS dataset. We made this choice mainly because publicly available operational SIP datasets representative of modern multi-operator 5G environments are currently lacking. As discussed in Section II-B, most existing benchmarks rely on synthetic or outdated traffic traces that do not adequately capture the architectural complexity, protocol diversity, and noise of contemporary deployments.

Although the evaluation relies on a single corpus, the dataset aggregates traffic collected from multiple operational 5G networks and therefore captures a certain degree of heterogeneity across operators, network configurations, and signaling scenarios. However, operator provenance is not retained at the individual-dialog level, which prevents a leave-one-operator-out evaluation on the available data. Moreover, operator-specific IP addresses are mapped to semantic network-function identifiers during preprocessing, reducing the dependence of the representation on operator-specific addressing schemes. Nevertheless, evaluation on a multi-network corpus does not replace an explicit transfer experiment on an independently collected dataset or on a fully held-out operator. Accordingly, the reported results demonstrate generalization to previously unseen signaling sequences within the evaluated multi-network corpus, but do not provide conclusive evidence of transferability to an independent operator. As discussed in Section III, currently available public SIP benchmarks do not enable a controlled transfer evaluation under the same input representation and ground-truth definition.

While this limits the immediate generalizability of the results, the adopted dataset provides a realistic and challenging evaluation setting, making it a meaningful benchmark for assessing the proposed framework. Within this context, the combination of a LLaMa-like Transformer and K-Means++ achieved the best performance among the considered configurations. However, readers should interpret these findings with respect to the evaluated setting, rather than as evidence of universal optimality across all network anomaly detection scenarios.

More broadly, we designed SIP-Classifier as a modular framework, allowing users to adapt its internal components to the characteristics of different data distributions and operational environments. To further assess generalizability, future work will evaluate the framework on additional datasets from heterogeneous network contexts.

In a deployment involving a new operator, the protocol-aware model could provide an initial representation, since SIP methods, response codes, and the semantic roles of the main IMS network functions are shared across deployments. If the local traffic distribution differs substantially from the training distribution, the Transformer could be adapted through self-supervised fine-tuning on representative unlabeled local traffic, followed by refitting the normalization parameters, PCA projection, and clustering components. Expert review would still be required to interpret the resulting clusters. Where operator-level provenance is available, future work will also consider leave-one-operator-out experiments.

Another limitation concerns how we treat sequences labeled unknown. We excluded these samples from the binary anomaly

detection evaluation, since they lack a reliable ground-truth assignment for the valid/anomalous decision, preventing a meaningful computation of binary evaluation metrics. As a result, the current study discriminates only between reliably labeled valid and anomalous signaling sequences. Future work will extend the framework to handle unknown sequences, for instance in an open-set or out-of-distribution detection setting.

In our dataset, the unknown label reflects uncertainty in the expert annotation rather than a well-defined third class. An unknown dialog may correspond to a rare but valid signaling procedure, an anomalous behavior whose cause cannot be confidently established, or a previously unseen operational pattern. Importantly, unknown dialogs are retained during self-supervised Transformer training and therefore contribute to learning the representation of operational SIP-IMS traffic. Their exclusion applies only to the downstream stages that require an unambiguous binary ground truth, such as cluster-to-class mapping and the computation of binary classification metrics.

In a practical deployment, the binary formulation could be complemented by an open-set rejection mechanism. Samples characterized by a large distance from the nearest centroid, an ambiguous assignment among multiple clusters, or a low likelihood under the autoregressive model could be marked as uncertain and flagged for expert inspection rather than being automatically classified as anomalous. Recurrent uncertain patterns could subsequently be clustered and incorporated into model updates after expert validation. The design, calibration, and quantitative evaluation of this extension are left for future work.

VII. CONCLUSION

This paper presented *SIP-Classifier*, an unsupervised framework for analyzing SIP signaling flows in IMS networks through protocol-aware representation learning. By combining a domain-specific tokenization strategy with autoregressive Transformer-based modeling and clustering, the proposed methodology captures the structural semantics of SIP procedures and enables the automatic separation of valid and anomalous signaling sequences without requiring labeled data.

Our results on a large real-world dataset collected from operational 5G IMS cores show that this combination is effective in practice and achieves better performance than the considered baselines on previously unseen sequences from the evaluated corpus. In particular, the evaluation highlights that protocol-aware tokenization is a key enabler for learning meaningful latent representations of signaling behavior, while the unsupervised design makes the framework applicable to complex operational settings where manual labeling is expensive and often impractical. At the same time, the current implementation is better suited to offline or near-line analysis than to strict real-time monitoring, and the present evaluation is limited to a single, although realistic and comprehensive, operational dataset. Future work will therefore focus on extending the framework to additional heterogeneous datasets, handling unknown or previously unseen SIP dialog patterns, and investigating lighter models and incremental clustering

strategies to enable earlier and more scalable anomaly detection in higher-throughput settings.

ACKNOWLEDGMENT

The authors thank Eng. Mauro Felici and the working group at RESI Informatica S.p.A., for providing the raw data from which the dataset subject of this work was extracted.

REFERENCES

- [1] Statista. (2007). *Number of Smartphones Sold to End Users Worldwide From 2007 to 2023*. Accessed: Jul. 18, 2025. [Online]. Available: <https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/>
- [2] M. Poikselkä, H. Holma, J. Hongisto, J. Kallio, and A. Toskala, *Voice Over LTE (VoLTE)*. Hoboken, NJ, USA: Wiley, 2012.
- [3] M.-I. T. Training, "Introduction to voice over new radio," in *Proc. MPiRICAL*, 2020.
- [4] *IP Multimedia Subsystem (IMS); Stage 2* (Release 5), document TS 23.228, 3GPP, 2004.
- [5] J. Rosenberg et al., "SIP: Session initiation protocol," Internet Eng. Task Force (IETF), Fremont, CA, USA, Tech. Rep. RFC 3261, 2002.
- [6] A. Uzelac and Y. Lee, "RFC 6405: Voice over IP (VOIP) SIP peering use cases," Internet Eng. Task Force (IETF), Fremont, CA, USA, Tech. Rep. RFC 6405, 2011.
- [7] V. Fajardo, A. J. Loughney, and G. Zorn, *Diameter Base Protocol*, document RFC 6733, Oct. 2012. [Online]. Available: <https://www.rfc-editor.org/info/rfc6733>
- [8] H. Schulzrinne, S. Casner, R. Frederik, and V. Jacobson, *RTP: A Transport Protocol for Real-time Applications*, document RFC 3550, Jul. 2003. [Online]. Available: <https://www.rfc-editor.org/info/rfc3550>
- [9] D. Pereira, R. Oliveira, and H. S. Kim, "Abnormal signaling SIP dialogs detection based on deep learning," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–6.
- [10] D. Pereira and R. Oliveira, "Detection of signaling vulnerabilities in session initiation protocol," in *Proc. 12th IFIP WG 5.5/SOCOLNET Adv. Doctoral Conf. Comput., Elect. Ind. Syst. Technol. Innov. Appl. AI Syst. Cham, Switzerland: Springer*, 2021, pp. 209–217.
- [11] D. Pereira, R. Oliveira, and H. S. Kim, "A machine learning approach for prediction of signaling SIP dialogs," *IEEE Access*, vol. 9, pp. 44094–44106, 2021.
- [12] C. Feio, D. Pereira, R. Oliveira, and P. Amaral, "Correlation-based abnormal SIP dialog identification: A performance comparison with Bayesian and deep learning approaches," *IEEE Access*, vol. 13, pp. 134484–134496, 2025.
- [13] M. Nassar, R. State, and O. Festor, "Labeled voip data-set for intrusion detection evaluation," in *Proc. Meeting Eur. Netw. Universities Companies Inf. Commun. Eng. Cham, Switzerland: Springer*, 2010, pp. 97–106.
- [14] A. Vaswani et al., "Attention is all you need," in *Proc. Advances Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.
- [15] H. Touvron et al., "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.
- [16] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (PCA)," *Comput. Geosci.*, vol. 19, no. 3, pp. 303–342, 1993.
- [17] *Signaling Flows for the IP Multimedia Call Control Based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP), Stage 3*, document TS 24.228, 3GPP, 2006.
- [18] J. Xiao, C. Huang, and J. Yan, "A flow-based traffic model for SIP messages in IMS," in *Proc. GLOBECOM - IEEE Global Telecommun. Conf.*, Nov. 2009, pp. 1–7.
- [19] J. Fabini, N. Jordan, P. Reichl, A. Poropatich, and R. Huber, "IMS in a bottle: Initial experiences from an OpenSER-based prototype implementation of the 3GPP IP multimedia subsystem," in *Proc. Int. Conf. Mobile Bus.*, Jun. 2006, p. 13.
- [20] A. A. Khudher, A. Munther, and S. Praptodiyono, "Efficient IPv4-IPv6 translation mechanism for IMS using SIP proxy," *Int. J. Internet Protocol Technol.*, vol. 15, no. 1, pp. 41–52, 2022.
- [21] H. Khelifi and J.-C. Gregoire, "IMS application servers: Roles, requirements, and implementation technologies," *IEEE Internet Comput.*, vol. 12, no. 3, pp. 40–51, May 2008.
- [22] A. M. Hagalisletto and L. Strand, "Formal modeling of authentication in SIP registration," in *Proc. 2nd Int. Conf. Emerg. Secur. Inf. Syst. Technol.*, Aug. 2008, pp. 16–21.
- [23] Y. Bai, X. Ye, and Y. Ma, "Formal modeling and analysis of SIP using colored Petri nets," in *Proc. 7th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Sep. 2011, pp. 1–5.
- [24] S. Matetic and M. B. Babac, "Model-based verification of the SIP invite scenario," in *Proc. 36th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2013, pp. 841–846.
- [25] D. Pereira, R. Oliveira, and H. S. Kim, "Classification of abnormal signaling SIP dialogs through deep learning," *IEEE Access*, vol. 9, pp. 165557–165567, 2021.
- [26] D. Pereira and R. Oliveira, "Detection of abnormal SIP signaling patterns: A deep learning comparison," *Computers*, vol. 11, no. 2, p. 27, Feb. 2022.
- [27] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proc. SIGCOMM workshop Mining Netw. data*, Sep. 2006, pp. 281–286.
- [28] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection," in *Proc. 17th Usenix Conf. Secur. Symp.*, 2008, pp. 139–154.
- [29] A. Radford. (2018). *Improving Language Understanding With Unsupervised Learning*. Accessed: Oct. 25, 2024. [Online]. Available: <https://openai.com/blog/language-unsupervised>
- [30] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," in *New Directions in Statistical Physics: Econophysics, Bioinformatics, and Pattern Recognition*. Berlin, Germany: Springer, 2004, pp. 273–309.
- [31] T. Kurita, "Principal component analysis (PCA)," in *Computer Vision: A Reference Guide*. Cham, Switzerland: Springer, 2021, pp. 1013–1016.
- [32] I. Jolliffe, "Principal component analysis," in *Encyclopedia of Statistics in Behavioral Science*. Chichester, U.K.: Wiley, 2005.
- [33] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 2006-13, 2006.
- [34] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognit.*, vol. 36, no. 2, pp. 451–461, Feb. 2003.
- [35] K. V. Mardia, J. T. Kent, and C. C. Taylor, *Multivariate Analysis*. Hoboken, NJ, USA: Wiley, 2024.
- [36] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. (2019). *Language Models Are Unsupervised Multitask Learners*. Accessed: Oct. 25, 2024. [Online]. Available: <https://cdn.openai.com/better-language-models/languagemodelsareunsupervisedmultitasklearners.pdf>
- [37] T. Brown et al., "Language models are few-shot learners," in *Proc. Advances Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.
- [38] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
- [39] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited: Why and how you should (Still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, pp. 1–21, Sep. 2017.
- [40] F. Sun, S. Wang, C. Zhang, and H. Zhang, "Clustering of unknown protocol messages based on format comparison," *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107296.
- [41] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.
- [42] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1715–1725.
- [43] Y. Wu et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.



Antonio Iacobelli is currently pursuing the Ph.D. degree in computer science and engineering with the Alma Mater Studiorum–Università di Bologna, Italy. His research focuses on cybersecurity, particularly on the security of industrial control systems, leveraging program analysis, formal methods, and artificial intelligence for the detection of cyber threats and anomalous behavior.



Giorgio Franceschelli received the M.Sc. degree in computer engineering and the Ph.D. degree in computer science and engineering from the University of Bologna, Italy. He is currently a Research Fellow with the Department of Computer Science and Engineering, University of Bologna, working on various areas of artificial intelligence, including computational creativity, deep learning, and reinforcement learning.



Marco Prandini (Member, IEEE) received the Ph.D. degree in electronics and computer engineering from the Department of Computer Science and Engineering, University of Bologna, Italy, in 2000. He is currently an Associate Professor with the Department of Computer Science and Engineering, University of Bologna. His research activities started in the field of public-key infrastructures and later moved to subjects related to the security of computing infrastructures, specifically investigating microservice-based architectures, software-defined networks, the IoT, and industrial control systems.



Lorenzo Rinieri (Member, IEEE) received the Ph.D. degree in computer science and engineering from the Alma Mater Studiorum–Università di Bologna, Bologna, Italy, in 2026. He is currently a Post-Doctoral Researcher with the Alma Mater Studiorum–Università di Bologna. His research interests include industrial control systems security, network security, and software-defined networking.



Mirco Musolesi (Member, IEEE) is currently a Full Professor of computer science with the Department of Computer Science, University College London. He is also a Full Professor of computer science with the Department of Computer Science and Engineering, University of Bologna. Previously, he held research and teaching positions at Dartmouth, Cambridge, St Andrews, and Birmingham. The focus of his laboratory is on machine learning/artificial intelligence and their applications to a variety of domains, including networked and distributed systems.



Franco Callegati (Senior Member, IEEE) is currently a Full Professor with the University of Bologna, Italy. He has been active in EU-funded research projects since FP4. His research interests include the modeling and experimental validation of advanced networking solutions, programmable data planes, industrial control systems, and the security aspects of such architectures.