IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 January 2024, accepted 13 February 2024, date of publication 16 February 2024, date of current version 26 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3366803

RESEARCH ARTICLE

Toward Auto-Modeling of Formal Verification for NextG Protocols: A Multimodal Cross- and Self-Attention Large Language Model Approach

JINGDA YANG, (Graduate Student Member, IEEE), AND YING WANG, (Member, IEEE)
School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ 07030, USA

Corresponding author: Ying Wang (ywang6@stevens.edu)

This work was supported by the Defense Advanced Research Project Agency (DARPA) under Grant D22AP00144.

ABSTRACT This paper introduces Auto-modeling of Formal Verification with Real-world Prompting for 5G and NextG protocols (AVRE), a novel system designed for the formal verification of Next Generation (NextG) communication protocols, addressing the increasing complexity and scalability challenges in network protocol design and verification. Utilizing Large Language Models (LLMs), AVRE transforms protocol descriptions into dependency graphs and formal models, efficiently resolving ambiguities and capturing design intent. The system integrates a transformer model with LLMs to autonomously establish quantifiable dependency relationships through cross- and self-attention mechanisms. Enhanced by iterative feedback from the HyFuzz experimental platform, AVRE significantly advances the accuracy and relevance of formal verification in complex communication protocols, offering a groundbreaking approach to validating sophisticated communication systems. We compare CAL's performance with state-of-the-art LLM-based models and traditional time sequence models, demonstrating its superiority in accuracy and robustness, achieving an accuracy of 95.94% and an AUC of 0.98. This NLP-based approach enables, for the first time, the creation of exploits directly from design documents, making remarkable progress in scalable system verification and validation.

INDEX TERMS Formal verification, cross-attention, self-attention, natural language protocol, formal flow graph.

I. INTRODUCTION

The third generation partnership project (3GPP) published its Release 17 (Rel-17) specifications at the end of the first quarter of 2022 [1]. Since then, parallel and subsequent releases have been rolled out to enhance and address new and unfulfilled requirements from previous releases. It ensures ongoing innovation and improvement in mobile communication technologies, aligning with the evolving needs and advancements in the field. 3GPP protocols consist of a multitude of technical specifications and documents that cover various aspects of mobile communication networks, including radio access, core network, and service capabilities. Encompassing a wide array of technical specifications, 3GPP protocols are extensive, with documents like the Radio Resource Control (RRC) in Release 17 spanning over a thousand pages [2]. The extensive nature of these documents, coupled with their distributed security protocols, renders manual verification and testing both time-intensive and susceptible to error. This complexity escalates in future network generations, exacerbating the risk of zero-day vulnerabilities.

Furthermore, the incorporation of O-RAN and network function virtualization introduces additional layers of complexity and potential attack vectors through cloud APIs into 5G and future G infrastructure [3]. These advancements offer enhanced functionality but expose operators to novel security

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Yuan Chen.

challenges, especially in the relatively uncharted territory of cloud security. This complexity, compounded by the involvement of various entities in development, heightens the risk of security breaches, particularly through misconfigured containers and exposed APIs.

Thus, in 5G and future G systems and networks, logical attacks exploiting protocol logic errors represent a significant vulnerability category. These attacks are challenging to detect due to the scale and complexity of the network systems. Formal verification encompasses a broad range of techniques used to prove or disprove the correctness of algorithms, protocols, systems, or software with respect to certain formal specifications. It also involves mathematical analysis to ensure that a system behaves as intended. Utilizing the Dolev-Yao (DY) formal attack model [4], [5], one of the widely adopted methodologies, formal verification has demonstrated its effectiveness in identifying flaws in infrastructure and communication protocols. However, this approach is limited to abstract specifications, and, as of now, the full automation of protocol verification and validation of their implementations remains beyond the reach of current technologies [6], [7].

With the recent advancements in Large Language Models (LLMs), an intriguing question emerges: Can LLMs contribute to accelerating the design and verification of large-scale protocols, and can they be effectively integrated with system validation processes that involve existing implementations?

To answer this question, two crucial conditions must be addressed: clarifying ambiguities and capturing design intentions. The first condition involves resolving the conflict between the ambiguity inherent in natural language processing and the need for explicitness in formal verification modeling. Resolving this conflict is essential to demonstrate the potential to provide measurable and verifiable trustworthiness. The second condition requires differentiating intentional relationships or dependencies from unintentional ones identified by LLMs in targeting protocol designs. The differentiation is crucial in transforming the design-intended relationships into mathematical and logical expressions in formal verification.

In the past several months, both academic and industry sectors have increasingly focused on applying LLMs in the realm of formal verification. Two major areas of this application are using prompt engineering and LLMs for hardware assertion [8], [9], [10] and software system Bounded Model Checking (BMC) [8], [9]. To our knowledge, no published work yet explores the use of LLMs in verifying communication-related protocols and specifications, especially for the large and complex protocols of 3GPP releases. One of the reasons is due to a common challenge in existing research is capturing trustworthy design intent in a set of assertions for use in formal verification or testing-based checking. This challenge intensifies as system distribution and complexity increase in the case of 5G and the future G, particularly when intertwined with a broad spectrum of various usage scenarios and verticals. The iterative prompts to the LLMs has limitations due to the complex and broad dependencies among identifiers, commands, and properties.

Further more, our previous work on non-LLM-based NLP approach for 5G and other communication protocols [11] marked a significant shift from manual to automated, accuracy-focused analysis in translating natural language oriented protocols into formal models. While it revealed limitations in handling complex semantic relationships with strong contextual control, it also highlighted the potential and direction for LLM-based NLP in formal modeling. We found this approach becomes particularly promising in detecting design intentions with greater intelligence and enhancing trustworthiness through its connection to learn from real-world experiments [12].

In this paper, we introduce a novel approach, named Auto-modeling of Formal Verification with Real-world Prompting for 5G and NextG protocols (AVRE), which addresses the need for scalable formal verification in the domains of 5G and NextG. AVRE uses LLMs to clarify ambiguities and capture design intent, transforming protocol descriptions into dependency graphs and formal models. Our method differs from existing techniques by integrating a transformer model with the LLM, allowing quantifiable dependency relationships to be generated under supervision and enabling transformative learning without human involvement. This system, enhanced by iterative feedback from an experimental platform: HyFuzz [12], fills a research gap by combining experience-based and logical dependency analyses in protocol documentation, thereby significantly improving the accuracy and relevance of formal verification in complex communication protocols.

A. RELATED WORK

1) CONVERTING INFORMAL NATURAL LANGUAGE SYSTEM DESIGNS AND PROTOCOLS TO FORMAL DESCRIPTION

Approaches to transforming natural language descriptions into formal models have seen considerable advancements, evolving through the introduction of diverse methodologies over the years. A decade ago, Drechsler et al. [13] proposed a paradigm that incorporated the Formal Specification Level (FSL), adeptly bridging the gap between informal textbook specifications and formal Electronic System Level (ESL) interpretations. Subsequently, Banarescu et al. [14] proposed a hybrid methodology that converted linguistic expressions into formal paradigms by merging symbolic and statistical techniques. With the advent of deep learning, Dong and Lapata [15] employed neural networks to convert natural language instructions into executable codes. This work was further enhanced by the contributions of Reddy et al. [16] in 2019, who focused on semantic parsing, utilizing denotations to transform complex linguistic structures into formalized notations. Meanwhile, the application of regular expressions has been identified as a viable means to extract formal specifications from natural language narratives, providing advanced components in deep learning frameworks [17]. Despite these developments, models based on these previous methodologies have achieved an accuracy threshold of approximately 90%, which is inadequate for ensuring complete recall. This poses a challenge in the precise conversion of natural language protocols into formal formulations.

2) LLMS BASED FORMAL VERIFICATION

LLMs have demonstrated impressive reasoning and assertion capabilities for formal verification [9], [18], [19]. Research in [8] and [9] has explored using LLMs to generate temporal logic specifications and assertions from unstructured natural languages. Meanwhile, studies in [20] and [21] focus on leveraging LLMs to enhance BMC for identifying software vulnerabilities and deriving counterexamples. In [10], the authors trained GPT-4 to generate correct SystemVerilog Assertions (SVA) through iterative prompt refinement with rules. However, it remains unclear how these models derive answers and whether they rely on simple heuristics rather than a generated chain-of-thought [18]. The current state of the art prioritizes producing formal specifications and properties quickly, albeit with slight inaccuracies, over generating perfect specifications or correctness statements [19]. The non-transparency related to LLM heuristics leads to a large number of irrelevant dependencies, resulting in low precision in dependency classification. To address this, our experimental platform connects to guide and refine the dependency graph range.

3) PROMPTING LIMITATIONS IN LLM ENABLED FORMAL VERIFICATION

Furthermore, the majority of existing work relies on prompt engineering. LLM-integrated applications blur the line between data and instructions [22]. LLMs can produce non-deterministic outputs, potentially yielding different results for the same prompt. This variability poses a potential threat to the validity of scientific conclusions unless researchers adapt their methods to account for it in their empirical analyses [23]. The adoption of prompting methods introduces challenges in iterative formal verification without human involvement. The randomness in LLMs is influenced by the sampling methods used during text generation, such as top-k sampling or nucleus sampling [24], limiting its application in classifiers or deterministic types of applications. To address the non-determinism and iterative formal verification, distinct from current formal verification methods that utilize prompt engineering, we designed an open-access LLM, integrated with a transformer model, to achieve supervised dependency.''

4) DIGITAL ENGINEERING AIDED FORMAL GUIDED SYSTEM VALIDATION

In the field of integrated Design Validation, there has been recent research progress in combining formal verification with simulation, resulting in a practical validation engine with reasonable run-time [25]. Experimental work in the context of 5G has gained significant attention over the past few years, shifting from the simulation-driven research used in previous mobile network generations to system implementation prototyping [26]. In our previous work, we explored a Formal-guided Fuzzing testing approach [27], [28] to bridge design verification and system validation. This approach complements the scalability limitations of formal verification and addresses the impacts of detected vulnerabilities. We have introduced a fuzzing digital twin [29] to provide an open and automated platform for systematically that enables an autonomous detection of vulnerabilities and unintended emergent behaviors in 5G infrastructures. However, a constraint of this initial approach is heavy reliance on expert insights to identify and articulate formal relationships.

B. CONTRIBUTION

The main original contributions of this work are summarised as follows:

- The research introduces a novel approach that utilizes LLMs to address critical challenges in formal verification, specifically in clarifying ambiguities and capturing design intent. This approach combines experience-based prior-probability distribution with logical dependency analysis in protocol documentation, and leverage experimental-based posterior-probability to enhances the accuracy and relevance of the dependency graph by the cross-attention-based LLM (CAL) model.
- The paper presents a new systematic solution named AVRE, which is equipped with CAL, a continuously-learning, cross-attention-based language model (LLM). CAL is designed to interpret and transform protocol descriptions into dependency graphs, which can then be converted into formal models. CAL is enhanced by incorporating iterative feedback from an experimental platform, HyFuzz, focusing on refining the capture of intentions and resolution of ambiguities.
- In CAL, we introduce a novel multi-session detection method that bypasses traditional token count barriers. By segmenting comprehensive protocols into manageable sections and concurrently processing them, it can output detailed and quality analysis without token constraints. CAL incorporates refined cross-attention mechanisms instead of prompting based scheme, achieves high efficiency to identify and interpret complex formal relationships with enhanced accuracy and insight.
- We have developed a scalable cross-sessions dependency graph that supports the hierarchy of formal analysis, facilitating the revelation of in-depth relationships embedded within protocols to systematically pinpoint and counteract vulnerabilities.

The rest of our paper is organized as follows. Section I-A listed the existing literature in the field of LLMs, dependency graph, formal verification, and experimental based validation. We introduce our proposed EVRA and show the architecture

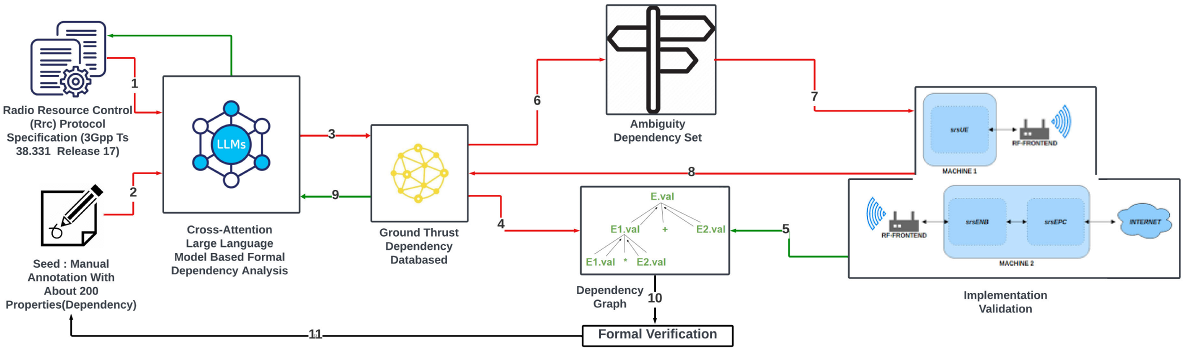

Radio Resource Control (Rrc) Protocol Specification (3Gpp Ts 38.331 Release 17)
Seed : Manual Annotation With About 200 Properties(Dependency)
LLMs
Cross-Attention Large Language Model Based Formal Dependency Analysis
Ground Thrust Dependency Databased
Ambiguity Dependency Set
E.val
E1.val
+
E2.val
E1.val
*
E2.val
Dependency Graph
Formal Verification
srsUE
RF-FRONTEND
MACHINE 1
RF-FRONTEND
srsENB
srsEPC
INTERNET
MACHINE 2
Implementation Validation
1
2
3
4
5
6
7
8
9
10
11

FIGURE 1. System overview of auto-modeling and trustworthy for formal verification and validation in 5g and nextg security protocols. Red line shows the process from informal system protocols to dependency graph and formal expression, and green line shows the formal guided fuzz testing feedback to the CAL and refine the results.

of the system in Section II, followed by the detailed description of the system in Section III and Section IV. We provide the performance analysis and in Section V. Lastly, we conclude our work with future research directions in Section VI.

II. SYSTEM OVERVIEW

In addressing the critical need for scalable formal verification in the rapidly evolving domains of 5G and NextG protocols and specifications, in this paper, we present a pioneering approach that utilizes LLMs to overcome key challenges of clarifying ambiguities and capturing design intent. We present a novel systematic solution AVER as shown in Fig. 1: Auto-modeling of Formal Verification with Real-world Prompting for 5G and NextG protocols.

At the heart of AVRE is a continuously-learning, CAL designed to interpret and transform protocol descriptions into dependency graphs, which are subsequently converted into formal models. Differing from the current state of the art in using LLM prompting for formal verification, our approach integrates the transformer model with the LLM model, enabling it to generate a quantifiable dependency relationship under supervision. This facilitates the transformative learning of the LLM without human involvement, meeting the requirements for formal explicitness. This system is further enhanced by incorporating iterative feedback from our experimental platform: HyFuzz [12], aimed at refining intention capture and ambiguity resolution.

CAL is trained on labeled identifiers (step 1 and step 2) and formal properties, designed to predict formal properties (step 3) with a more controlled environments. The design of cross-attention mechanisms utilization the LLM model improve the capability of the model in complex protocol.

The experimental platform serves as two purpose: intention and the trustworthy. As shown in step 4 and step 5 in Fig. 1, it provides relevancy and design intention information, which is used with the dependency relationships detected by CAL together to form the dependency graph. For trustworthy, as shown in step 6 and step 7, the low confident prediction is set to the experimental platform to return evidence that could be added to ground truth dependence database. With multiple iterations, a robust dependency graph will be generated to sent to formal method verification. The detected vulnerability will be further used in protocol fortification.

The design of the system AVRE uniquely fills a gap in the existing research landscape by integrating experience-based prior-probability distribution with logical dependency analysis in protocol documentation using LLMs. Moreover, by leveraging experimental-based posterior-probability through real-world prompting, AVRE establishes an iterative learning loop, significantly enhancing the accuracy and relevance of the intention of the CAL model. This advancement marks a pivotal contribution to the field, offering new directions in the formal verification of complex communication protocols.

III. PROTOCOL ANALYSIS AND DATA PRE-PROCESS AND ANNOTATION

A. PROTOCOL ANALYSIS AND FORMAL PROPERTIES DEFINITION

In this paper, we use 3GPP Release 17 Radio Resource Control [30] as an example to illustrate AVRE. We identify two fundamental elements: procedures and identifiers. Procedures are defined as sequences of actions and interactions among different entities. Identifiers in the protocols are specific labels or names used to uniquely recognize various elements within mobile telecommunications networks. For instance, IMSI (International Mobile Subscriber Identity) is a unique number associated with all cellular networks, used primarily for identifying individual subscribers for billing and identification purposes. IMEI (International Mobile Equipment Identity) is defined as a unique number to identify mobile devices, primarily used for identifying the device itself, rather than the subscriber [30]. The procedures define the structure of a set of identifiers that are connected dependently. The dependent relationship between the identifiers, framed by the structure defined in the procedures, generates the dependency graph, which is then converted for formal

TABLE 1. RRCSetupRequest-IEs field descriptions.

Identifier	Description
establishmentCause	Provides the establishment cause for the `RRCSetupRequest` in accordance with the information received from upper layers. gNB is not expected to reject an `RRCSetupRequest` due to unknown cause value being used by the UE.
ue-Identity	UE identity included to facilitate contention resolution by lower layers.

analysis. As shown in Table. 1, while the extraction of entities from protocol identifiers is straightforward, the classification of their dependency is a problem that CAL targets. However, the procedures include intricate interactions and designs, with unspecified space for various vendors to implement based on their existing infrastructures and devices. In this paper, we innovatively connect to real-world or simulated experimental platforms to aid in procedure formulation.

Considering confidentiality, integrity, and availability (CIA) triad which is widely accepted security model, along with the Dolev-Yao model for communication protol formal verification, we have proposed four essential properties to describe the dependency among identifiers [27]. These properties, each addressing a distinct facet of security augmentation within the specifications, are defined as follows:

1) **Confidentiality**: the capability of source to prevent private information from leakage of the destination. The source should be the selected encryption algorithm or the key of encryption algorithm, and the destination should be command or specific identifier, which are confidentiality protected by source.
2) **Integrity**: the capability of source to keep the destination information unmodified. Similar to confidentiality, the source should be the selected integrity algorithm or the key of integrity algorithm, and the destination should be the command or specific identifier, which are integrity protected by source.
3) **Authentication**: the ability of source to help User Equipment (UE) or gNodeB (gNB) to identify where and when the destination was sent from. The source should be the distinctive identifier, which can uniquely represent a entity (like gNB) or a communication session (like UE id).
4) **Accounting**: the source of accounting relationship is the counting identifier which can sequentially track and distinguish each transmitted command. The destination should be the sequence-protected command.
5) **Include**: the destination identifier is included in the source identifier.
6) **Generate**: the destination identifier is generated by the source identifier.

B. DATA PREPARATION

In our study, we annotated dependencies (formal properties) of identifiers in the 5G Radio Resource Control (RRC) protocol [30], focusing on Sections 5.3.1 to 5.3.5. From these sections, approximately 16,428 samples were annotated, including source, destination, and dependency relationships. Out of these, 1,218 samples were identified by domain experts who analyzed the documents to determine relationships. The remaining samples consist of source and destination pairs lacking a dependency relationship. In our proposed model, we treat this as a classification problem for all potential source-destination pairs, encompassing both relevant and non-relevant ones. Consequently, the large number of source and destination pairs results in highly imbalanced data.

In our analysis of the annotated data samples, we have identified an imbalance in the distribution between positive and negative samples. To address this issue and improve the model's performance, we propose employing a weight-balanced binary cross-entropy loss. Fig. 2 illustrates the size of each property type and the intersection sizes among them. It is clear from Fig. 2 that 'include' has the largest number of detections. Furthermore, the intersections of 'include' with 'account' and 'integrity' represent the two largest intersection counts, indicating a strong correlation between these properties. It is also important to note the significant presence of 'integrity' in multiple intersections, which is consistent with the importance of integrity in RRC protocols 3GPP in release 17. It is noteworthy that a substantial proportion of the intersection of 'include' and 'generate' is observed. This phenomenon arises from the practice wherein numerous identifiers generate child identifiers based on their prior information, thus conducting the inclusion of the generated identifier within nested structures.

IV. METHODOLOGY

A. BUILDING MULTIMODAL CROSS- AND SELF-ATTENTION LARGE LANGUAGE MODEL

In this work, we proposed CAL, an LLM embedded with cross- and self-attention model, as delineated in Fig. 3, considering both the contextual information from the original protocols and learns the defined dependency relationships among identifiers. CAL model employs a pretrained LLM (GPT-2), which consists of N transformer layers, to extract hidden insights from protocol descriptions. Here, considering the scalability of training and the performance accuracy, we select $N = 12$.

We incorporate cross-attention mechanisms to discern the relationships between the extracted latent information and query entities [31]. To further enrich the contextual understanding, we deploy self-attention frameworks that evaluate inter-relations among all positions from the preceding stage, guided by weighted considerations. In the final stage of the presented model, a linear classifier is implemented to infer probabilities associated with distinct formal attributes.

Self-attention generates contextual representations for a single sequence by computing weighted averages of all tokens, while cross-attention evaluates interdependent contextual relationships between the query sequence and

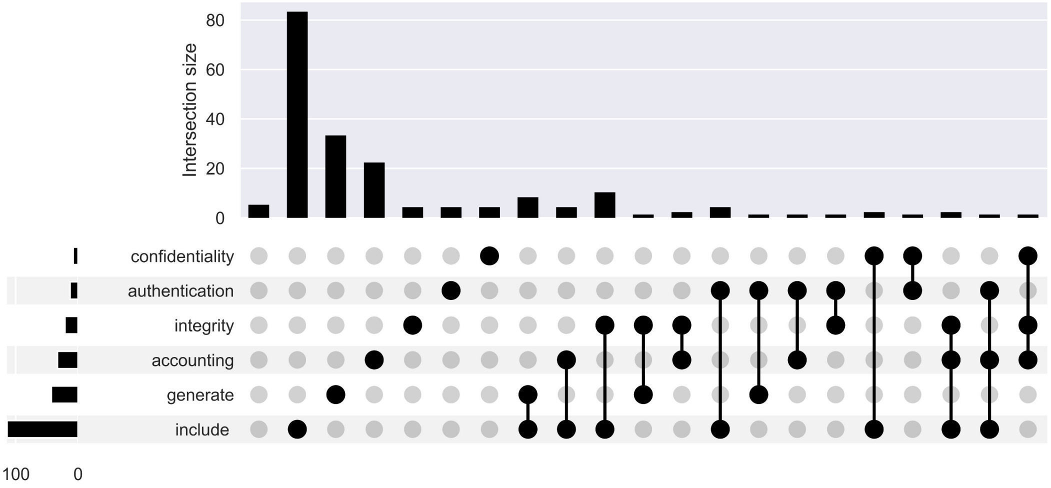


FIGURE 2. Formal properties statistical counts and intersections.

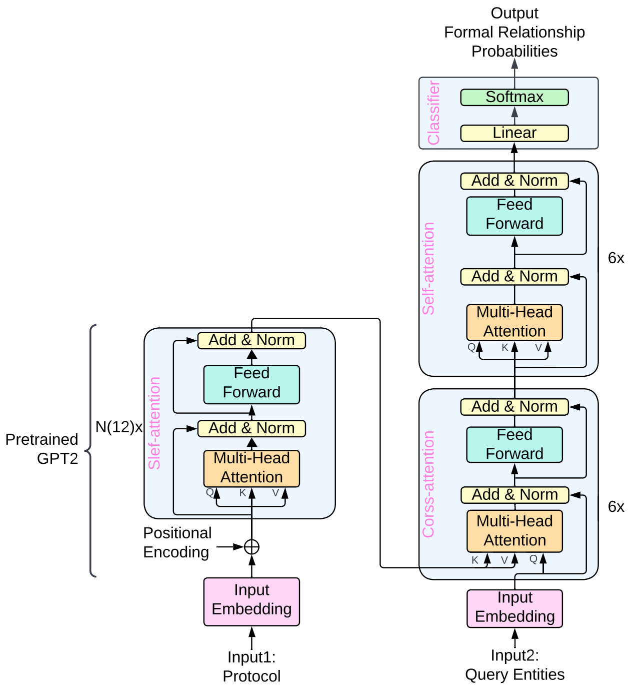


FIGURE 3. Attention driven formal identifier and property abstraction model.

the context sequence in transformer models [32], [33], [34]. In the proposed CAL model for converting complex contextual protocols into explicit dependency graphs, self-attention serves as the mechanism to understand the context within the protocols, while cross-attention understands and recognizes the relationships across the identifier sequences.

Self-attention calculates a weighted average of feature representations, where the weight represents the similarity score between pairs of feature representations. As defined in [32], an input sequence of n tokens of dimensions d, $X \in R^{n\times d}$, is extracted by three projection matrices $W_Q \in R^{d\times d_q}$, $W_K \in R^{d\times d_k}$ and $W_V \in R^{d\times d_v}$ ($d_q = d_k$). The input consists of queries and keys with dimension d_k, and values with dimension d_v. Three different feature representations $Q = XW_Q, K = XW_K$, and $V = XW_V$, where $Q \in R^{n\times d_q}$, $K \in R^{n\times d_k}$ and $V \in R^{n\times d_v}$. Intuitively, Q, K and V are separately regarded as query, key and value. Normalized dot-product of query and key is used to represent the attention score $A \in R^{n\times n}$ of each query and paired keys as:

$$A = softmax(\frac{QK^T}{\sqrt{d_q}}) \tag{1}$$

where division by $\sqrt{d_q}$ normalizes the dot-product of Q and K^T, ensuring that the distribution of the dot-product aligns with expectation $\mathrm{E} = 0$ and variance $\mathrm{Var} = 1$. Ultimately, attention value is calculated as follow,

$$\text{Self-attention}(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_q}})V = AV \tag{2}$$

where output $\text{Self-attention}(Q, K, V) \in R^{n\times d_v}$ is the weighted average of extracted features. Normally, a residual linear feed forward layer is incorporated to further distill the acquired knowledge represented by the weighted average features.

Utilizing the self-attention mechanism, cross-attention takes two input sequences with different sequence length $X_1 \in R^{n_1\times d}$ and $X_2 \in R^{n_2\times d}$ as inputs. Subsequently, $Q = X_1W_Q, K = X_2W_K$, and $V = X_2W_V$ are derived from X_1 and X_2 respectively and the dimension of attention value can be succinctly expressed as $R^{n_1\times d_v}$.

B. BALANCED LOSS FUNCTION

In Section III-B, we observed the imbalance in the data distribution of relationships amongst entities, and is evident in Table 2. In response to this challenge, we adopt a

TABLE 2. Imbalanced nature of formal property data.

Relationship	Class	Count	Relationship	Class	Count
Confidentiality	0	16421	Accounting	0	16399
Confidentiality	1	7	Accounting	1	29
Integrity	0	16408	Include	0	16329
Integrity	1	20	Include	1	99
Authentication	0	16417	Generate	0	16389
Authentication	1	11	Generate	1	39

weight-balanced binary cross-entropy loss, articulated in Equation 3. In this equation, N signifies the total number of cases, y_i represents the ground truth, $p(y_i)$ is the predicted probability, and $n_{(y_i)}$ denotes the proportion of class y_i within all cases. Utilizing the inverse class ratio amplifies the impact of the underrepresented class while tempering the dominance of the majority class, offering a balanced approach to mitigate the skewed data distribution.

$$L_i = -[y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))]$$
$$L = \frac{1}{N} \sum_{i=1}^{N} L_i \cdot (1 - \frac{n_{(yi)}}{N}) \tag{3}$$

C. CONNECTION TO EXPERIMENTAL PLATFORM

Inspired by the iterative prompting to the LLM, the experimental platform serves as the prompting server in real world. In order to clarify the ambiguity in the LLM model, the counterpart experiments configuration could be auto-generated and performed, in which the results generated is feedback to the LLM to improve the trustworthiness. Through the digital engineering module, which facilitates connections to available real-world execution platforms or digital twins, the process effectively bridges the gap between design intentions and real-world operations for mission-critical infrastructures.

The experimental platform, HyFuzz, employs a hybrid system model [12] that incorporates two distinct platforms: a ZeroMQ virtual model and an Over the Air (OTA) physical model. In addition to running 5G tests in various scenarios, the uniqueness of the HyFuzz platform is its ability to serve as relay nodes to perform various fuzzing tests. The relay node is capable of modifying and permutating commands and accessing messages exchanged between the UE and the gNB. HyFuzz facilitates multi-step Man-In-The-Middle (MITM) attacks, enabling the identification and analysis of vulnerabilities and the detection of various prompts due to the uncertainty of CAL model outputs and providing an accurate real-world ground truth. An overview of the HyFuzz is shown in Fig. 4.

As illustrated in Fig. 1, the experimental platform is seamlessly integrated with the CAL model, supporting it from two key aspects: design intention and trustworthiness. Regarding design intention, the platform supplies relevant information and design intents to the CAL model, enabling it to continue training based on novel evidence of dependency relationships, thus effectively constructing an accurate dependency graph. In terms of trustworthiness, low-confidence predictions are referred back to the experimental platform for further validation, potentially contributing new data to the ground truth dependency database. This iterative process aims to create a robust dependency graph, which will be utilized for formal method verification. Detected vulnerabilities will then be leveraged in protocol fortification."

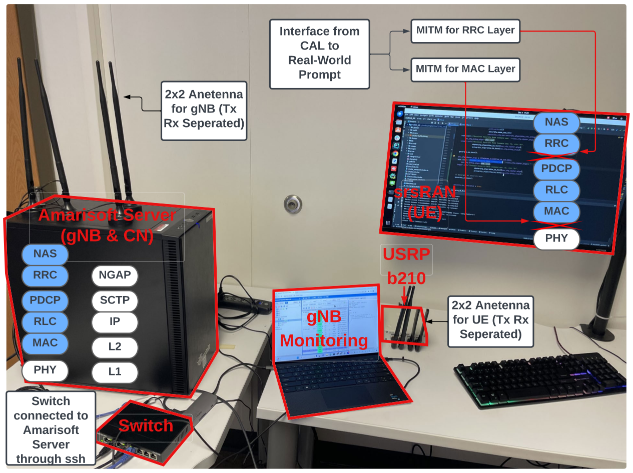

Interface from CAL to Real-World Prompt
MITM for RRC Layer
MITM for MAC Layer
2x2 Anetenna for gNB (Tx Rx Seperated)
NAS
RRC
PDCP
RLC
MAC
PHY
srsRAN (UE)
USRP b210
Amarisoft Server (gNB & CN)
NAS
RRC
PDCP
RLC
MAC
PHY
NGAP
SCTP
IP
L2
L1
gNB Monitoring
2x2 Anetenna for UE (Tx Rx Seperated)
Switch connected to Amarisoft Server through ssh
Switch

FIGURE 4. Virtual and over the air (OTA) mode experimental of hyFuzz [12].

V. SYSTEM PERFORMANCE ASSESSMENT

A. CAL EXPERIMENT SETTING

Considering the openness and accessibility of the LLMs, in this paper, we select the pre-trained GPT-2 model with 12 transformer layers and 12 attention heads as embedded pre-trained LLM included in CAL. The GPT-2's embedding size is set to 768, and the sequence length is configured to 1024, allowing for the effective extraction of hidden information from the extended protocols. Subsequent to the application of the LLM, the extracted hidden information is processed through 6 layers of self-attention with 6 attention heads and 6 layers of cross-attention with 6 attention heads. This structure aids in recognizing and processing the comprehensive relationships between the protocols and the associated query entities. Further details of the model's configuration are presented in Table 3.

B. CAL EXPERIMENT RESULT ANALYSIS

As depicted in Fig. 5, the consistent trends in both training and validation accuracy of CAL underscore the model's proficiency in extracting the dependency relationship information from the examined protocols and entities. In our experiment setting, formal properties are designated as positive, while their absence is categorized as negative. As presented in Table4, our model performs stable accuracy of 95.94%, and achieves 100% accuracy in recall, indicating the high and stable performance in formal property prediction and the trustworthiness of the model.

TABLE 3. Configuration of model.

Parameter	Value
GPT-2's Embedding Size	768
GPT-2's Sequentce Length	104
GPT-2's Layers	12
GPT-2's Attention Head	12
Cross-attention Layer	6
Cross-attention Attention Head	6
Self-attention Layer	6
Self-attention Attention Head	6
Learning Rate	$1e^{-7}$
Epoch	100
dropout rate	0.1
Train/Validation Ratio	9 : 1

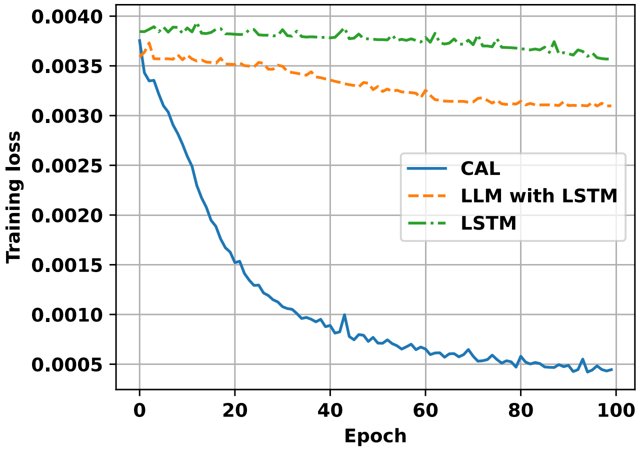


FIGURE 5. Training Loss: the training loss indicate the incapability of traditional models like LSTM in processing complex dependency and text.

We further compared our model with other state of art model performance. Table 4 leads to an explicit conclusion that the CAL significantly outperforms all other models delineated in Section V-A. Fig. 6 reveals that only the LSTM model and the hybrid LLMs with LSTM model exhibit a deficiency in the efficient extraction of information during the training phase, resulting in trapping in local optima. The result also indicates the necessity of LLMs in processing complex protocols and standards.

To compare the performance of transformer-based models with traditional non-transformer models, the LSTM, known for effectively capturing long-range dependencies and typically performs better than CNN and RNN, is chosen to replace the cross-attention and self-attention. To ensure consistency of experimentation, the LSTM is set with a hidden state size of 768 and 6 recurrent layers. Additionally, the LSTM alone was used to directly process natural language to predict formal properties, also with a hidden state size of 768 and 6 layers.

The Area Under the Curve (AUC) is commonly described as a statistical measure used to evaluate the performance of a classification model. As shown in Fig.8, we can see that via the True Positive Rate (TPR) against the False Positive Rate (FPR), CAL has an AUC value of 0.98, indicating excellent performance.

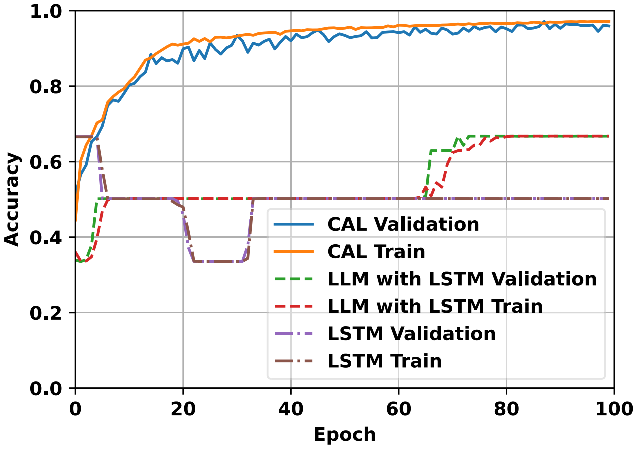


FIGURE 6. Training Result: Suboptimal models, such as those utilizing llm with lstm and standalone lstm, often get trapped in local minima due to class imbalance and specially designed loss.

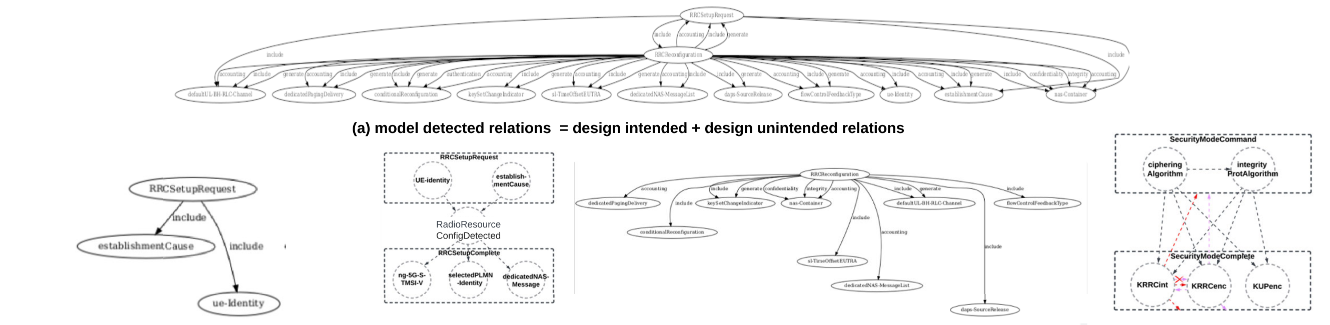
TABLE 4. Accuracy of different models.

Model	Accuracy
CAL	95.94%
LLM with LSTM	66.76%
LSTM	50.21%

Then, we further analyze the AUC for each type of dependency, as shown in Fig. 9. Confidentiality: This curve appears to be the closest to the top-left corner, suggesting it has the best performance among the four criteria. Integrity, Authentication, Accounting: These curves are further from the top-left corner compared to Confidentiality. While they still demonstrate good performance, they are not as effective as the Confidentiality curve.

To better visualize the interactions between relevant entities and protocol procedure descriptions, we utilize the Actions related to the transmission of "RRCSetupRequest" message as an example. This verification process aims to determine whether our model can effectively extract valuable information from the human-written protocol procedure descriptions.

In transformer-based models, each transformer layer has the capability to extract distinct information through different attention scores. In this study, we leverage the average score of cross-attention layers to visualize the interactions between relevant entities and protocol procedure descriptions. Taking the entities "establishmentCause" and "RRCSetupRequest", along with their corresponding procedure as an example, we generate an attention map (Fig. 10). The attention map clearly reveals that these two entities place more emphasis on specific terminologies, such as "establishmentCause", "mps-PriorityAccess", and "RRC", as well as the levels of indentation, such as "1>" and "2:". This observation highlights the cross-attention component's ability to effectively focus on and extract essential information from key terms within the input text.



(a) model detected relations = design intended + design unintended relations

(b) design intended for RRC Setup Requet (c) user friendly visulaization of (b) (d) design intended for Secuirty Mode Command (e) user friendly visulaization of (d)

FIGURE 7. Design Intention: (a) is the raw dependency relationships that detected by CAL, and (b) is the experiments filtered design intended dependency graph.

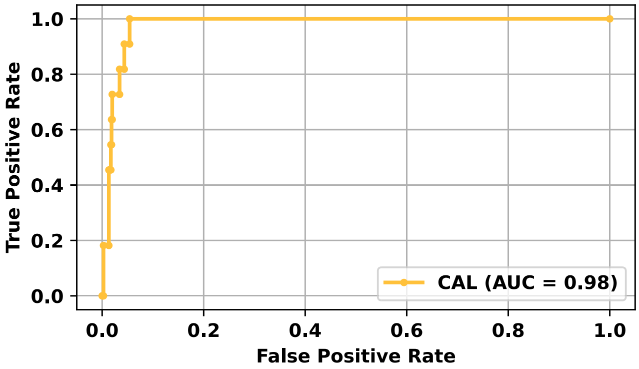


FIGURE 8. Receiver operating characteristic (ROC) curve of CAL model.

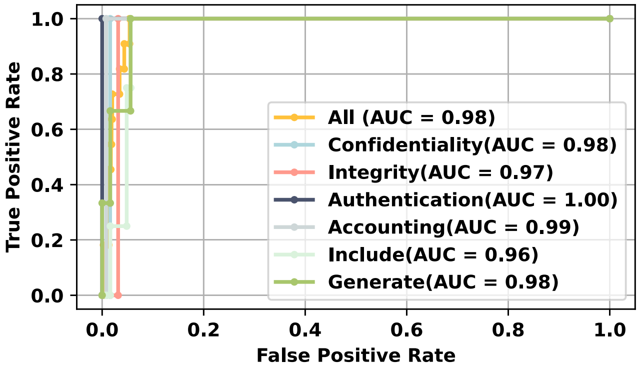


FIGURE 9. Dependency relations specific ROC curve of CAL model.

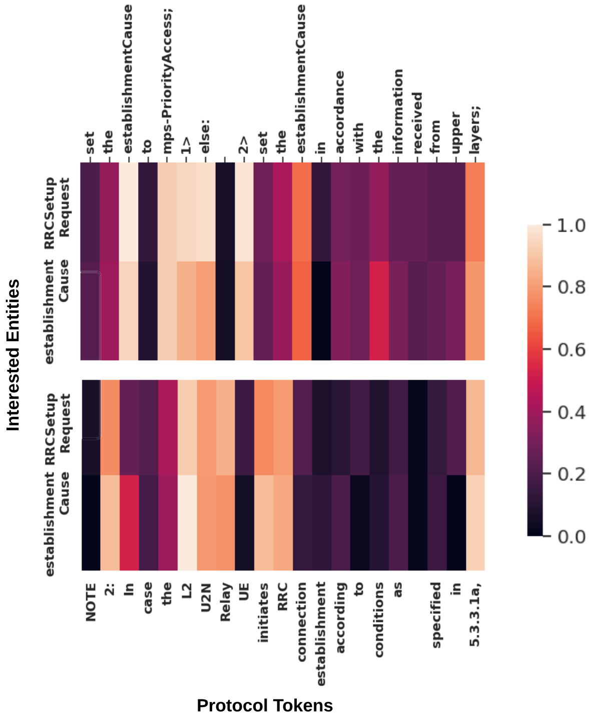


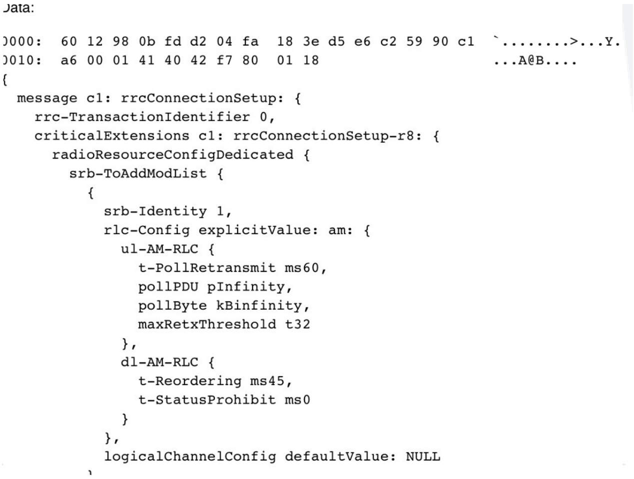
FIGURE 10. Attention map depicting average attention scores across layers: Lighter shades indicate higher attention; darker shades signify reduced attention.

C. CASE STUDY OF DESIGN INTENTION CAPTURING AND TRUSTWORTHY ENHANCEMENT VIA THE CONNECTION TO REAL-WORLD TESTBED

Case 1: Design Intention With the CAL model, the generated dependencies include both intended and unintended ones. Compared to extraction from human expertise, a portion of unintended dependency relationships is also detected. These unintended dependencies can be filtered out via connection to a real-world experimental platform.

We took the example of "RRC_setup_request" and "Security_mode_command", as shown in Fig. 7. Figure 7(a) illustrates the raw dependency relationships detected by CAL, while Figure 7(b) shows the experimentally filtered, design-intended dependency graph. Using the information flow from the experiment platform, as shown in Fig.11, we can derive the design-intended dependency graph presented in Fig. 7(b) and Fig. 7(d). Figures 7(c) and 7(e) are equivalent to Figures 7(b) and 7(d) respectively, offering a more user-friendly visualization.

Case 2: Trustworthy For dependencies predicted by CAL with low confidence, the uncertain dependency can be converted into test scripts and sent to an experimental platform to generate evidence. This evidence can be used to confirm or refute the existence of the dependency. As shown in Fig. 12,



```
Data:

0000:  60 12 98 0b fd d2 04 fa  18 3e d5 e6 c2 59 90 c1  `........>...Y.
0010:  a6 00 01 41 40 42 f7 80  01 18                    ...A@B....
{
  message c1: rrcConnectionSetup: {
    rrc-TransactionIdentifier 0,
    criticalExtensions c1: rrcConnectionSetup-r8: {
      radioResourceConfigDedicated {
        srb-ToAddModList {
          {
            srb-Identity 1,
            rlc-Config explicitValue: am: {
              ul-AM-RLC {
                t-PollRetransmit ms60,
                pollPDU pInfinity,
                pollByte kBinfinity,
                maxRetxThreshold t32
              },
              dl-AM-RLC {
                t-Reordering ms45,
                t-StatusProhibit ms0
              }
            },
            logicalChannelConfig defaultValue: NULL
```

FIGURE 11. Information flow-graph detected in experiment platform.

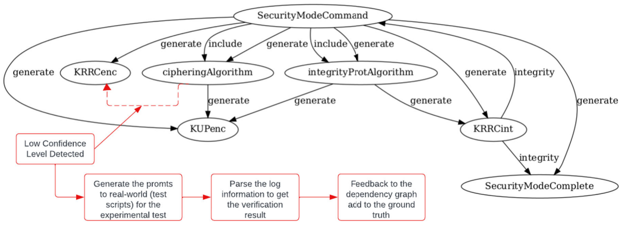
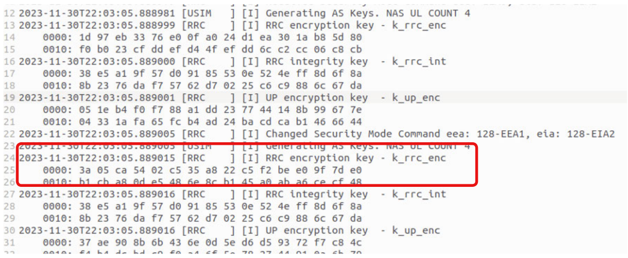

SecurityModeCommand
generate
include
generate
include
generate
generate
KRRCenc
cipheringAlgorithm
integrityProtAlgorithm
generate
integrity
generate
generate
generate
KUPenc
KRRCint
integrity
SecurityModeComplete
Low Confidence Level Detected
Generate the promts to real-world (test scripts) for the experimental test
Parse the log information to get the verification result
Feedback to the dependency graph add to the ground truth
generate

FIGURE 12. Information flow-graph detected in experiment platform.



```
2023-11-30T22:03:05.888981 [USIM    ] [I] Generating AS Keys. NAS UL COUNT 4
2023-11-30T22:03:05.888999 [RRC     ] [I] RRC encryption key - k_rrc_enc
    0000: 1d 97 eb 33 76 e0 0f a0 24 d1 ea 30 1a b8 5d 80
    0010: f0 b0 23 cf dd ef d4 4f ef dd 6c c2 cc 06 c8 cb
2023-11-30T22:03:05.889000 [RRC     ] [I] RRC integrity key  - k_rrc_int
    0000: 38 e5 a1 9f 57 d0 91 85 53 0e 52 4e ff 8d 6f 8a
    0010: 8b 23 76 da f7 57 62 d7 02 25 c6 c9 88 6c 67 da
2023-11-30T22:03:05.889001 [RRC     ] [I] UP encryption key  - k_up_enc
    0000: 05 1e b4 f0 f7 88 a1 dd 23 77 44 14 8b 99 67 7e
    0010: 04 33 1a fa 65 fc b4 ad 24 ba cd ca b1 46 66 44
2023-11-30T22:03:05.889005 [RRC     ] [I] Changed Security Mode Command eea: 128-EEA1, eia: 128-EIA2
2023-11-30T22:03:05.889005 [USIM    ] [I] Generating AS Keys. NAS UL COUNT 4
2023-11-30T22:03:05.889015 [RRC     ] [I] RRC encryption key - k_rrc_enc
    0000: 3a 05 ca 54 02 c5 35 a8 22 c5 f2 be e0 9f 7d e0
    0010: b1 cb a8 0d e5 48 6e 8c b1 45 a0 ab a6 ce cf 48
2023-11-30T22:03:05.889016 [RRC     ] [I] RRC integrity key  - k_rrc_int
    0000: 38 e5 a1 9f 57 d0 91 85 53 0e 52 4e ff 8d 6f 8a
    0010: 8b 23 76 da f7 57 62 d7 02 25 c6 c9 88 6c 67 da
2023-11-30T22:03:05.889016 [RRC     ] [I] UP encryption key  - k_up_enc
    0000: 37 ae 90 8b 6b 43 6e 0d 5e d6 d5 93 72 f7 c8 4c
```

FIGURE 13. Fuzzing log file to parse evidence information.

the dependency between KRRCenc and CipherAlgorithm shows low confidence in the detected "Integrity". This is automatically converted into fuzzing scripts to generate evidence, proving that there is no dependency between them, as indicated in the parsed information from the log file shown inf Fig.13.

D. FORMAL VERIFICATION AND ATTACK MODEL

Based on the predicted relationship between entities, our next objective is to perform the formal verification, as shown in Fig. 14 and Fig. 15. The interconnected sections and communicated commands consisting of identifiers. We can first fill up the interconnected sections and communicated commands, which is depicted in natural language protocols, shown in Fig. 16 to Fig. 19. And from the command table in natural language protocols, we can easily identify which identifiers are included in the commands.

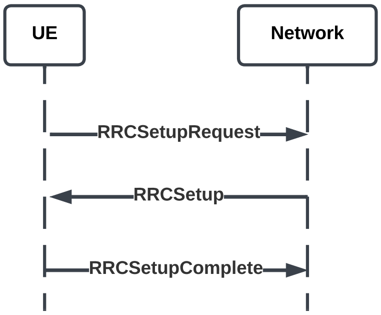


FIGURE 14. RRC connection flow graph.

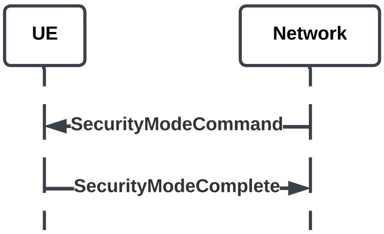


FIGURE 15. Security mode command flow graph.

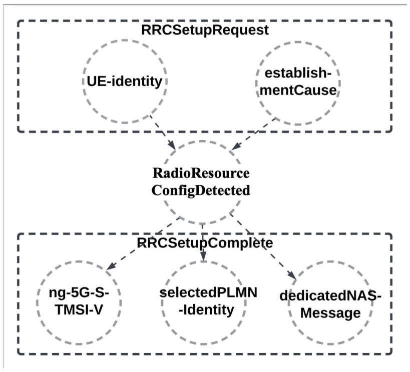


FIGURE 16. Ground truth RRC connection dependency graph.

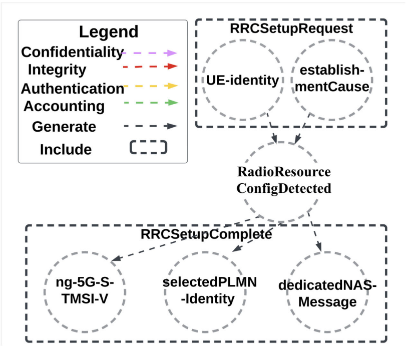


FIGURE 17. Predicted security mode command dependency graph.

Utilizing the identified formal properties, we construct a comprehensive dependency graph, which facilitates the derivation of formal dependencies through this graph.

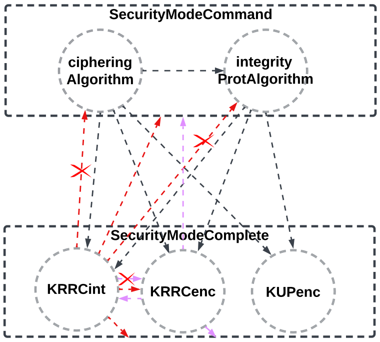


FIGURE 18. Ground truth security mode command dependency graph.

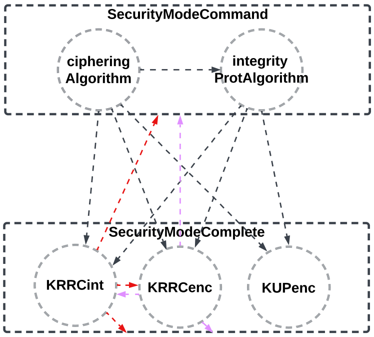


FIGURE 19. Comparison between predicted and ground truth dependency graph.

We consider the RRC connection establishment procedure as an example. Nodes were manually extracted from the natural language protocol, as depicted in Figs. 16 and 17. The visual representation employs boxes to signify "Include" and uses various arrow types to delineate distinct formal properties. Leveraging our predicted formal properties, we generated the RRC connection dependency graph (see Fig. 16). Remarkably, this aligns perfectly with the ground truth of the RRC connection dependency graph. However, during the Security Mode procedure, there were discrepancies between the predicted formal properties (as presented in Fig. 18) and the ground truth (see Fig. 19). Therefore, subsequent verification is advisable. This verification could be performed in the real-world prompt or by experts manually, as indicated in the system design of Fig. 1. Notably, while there may be the inclusion of extraneous formal properties, none are omitted. Building on previous related work [27], this dependency graph can be transformed into Proverif code, enabling formal verification. Compared to solely manual labeling of formal properties, our proposed model streamlines the process by narrowing down the entirety of natural language protocols to the task of parsing redundant formal properties from the crafted formal dependency graph.

VI. CONCLUSION

In conclusion, we present AVRE, a novel system for the formal verification of NextG protocols, leveraging LLMs to transform protocol descriptions into dependency graphs and formal models. Enhanced by the HyFuzz experimental platform, AVRE demonstrates significant advances in the accuracy and relevance of formal verification in complex communication protocols. The research underscores the efficiency of CAL, a continuously-learning, cross-attention-based LLM, in extracting formal properties and dependencies, outperforming traditional methods. The study emphasizes the potential of LLMs in enhancing trustworthiness and clarifying ambiguities in protocol verification, marking a significant contribution to the field. By reducing reliance on manual labeling and associated human errors, our method offers a more efficient approach, focusing only on the most pertinent formal relationships. Our work in extracting formal relationships from natural language protocols enhances the clarity and understanding of these protocols, ensuring a more robust, reliable, and efficient approach for protocol verification and system validation in large-scale, complex systems.

ACKNOWLEDGMENT

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

REFERENCES

[1] *5G Evolution Toward 5G Advanced: An Overview of 3GPP Releases 17 and 18*, 3GPP, Eur. Telecommun. Standards Inst., France, 2022.

[2] 3GPP, "Nr; radio resource control (RRC); protocol specification," 3rd Gener. Partnership Project (3GPP), Tech. Specification (TS) 38.331, Eur. Telecommun. Standards Inst., France, 2021. [Online]. Available: https://www.3gpp.org

[3] *Open Ran—5g Hacking Just Got a Lot More Interesting*, Karsten Nohl, Secur. Res. Labs, Berlin, Germany, 2022.

[4] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[5] I. Cervesato, "The Dolev-Yao intruder is the most powerful attacker," in *Proc. 16th Annu. Symp. Log. Comput. Sci.-LICS*, vol. 1, 2001, pp. 1–2.

[6] M. Ammann, L. Hirschi, and S. Kremer, "Dy fuzzing: formal Dolev-Yao models meet cryptographic protocol fuzz testing," in *Proc. 45th IEEE Symp. Secur. Privacy*, 2024.

[7] I. Rakotonirina, G. Barthe, and C. Schneidewind, "Decision and complexity of Dolev-Yao hyperproperties (technical report)," in *Proc. Symp. Princ. Program. Lang. (POPL)*, 2024.

[8] R. Kande, H. Pearce, B. Tan, B. Dolan-Gavitt, S. Thakur, R. Karri, and J. Rajendran, "LLM-assisted generation of hardware assertions," 2023, *arXiv:2306.14027*.

[9] M. Cosler, C. Hahn, D. Mendoza, F. Schmitt, and C. Trippel, "Nl2spec: Interactively translating unstructured natural language to temporal logics with large language models," 2023, *arXiv:2303.04864*.

[10] M. Orenes-Vera, M. Martonosi, and D. Wentzlaff, "Using LLMs to facilitate formal verification of RTL," 2023, *arXiv:2309.09437*.

[11] S. Yuan, J. Yang, S. Arya, C. Lipizzi, and Y. Wang, "From ambiguity to explicitness: NLP-assisted 5G specification abstraction for formal analysis," 2023, *arXiv:2308.03277*.

[12] J. Yang and Y. Wang, "A nextg hybrid testing platform for multi-step deep fuzzing and performance assessment from virtualization to over-the-air," in *Proc. IEEE Cloudnet*, 2023, pp. 1–6.

[13] R. Drechsler, M. Soeken, and R. Wille, "Formal specification level: Towards verification-driven design based on natural language processing," in *Proc. Forum Specification Design Lang.*, Sep. 2012, pp. 53–58.

[14] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, "Abstract meaning representation for sembanking," in *Proc. 7th Linguistic Annotation Workshop Interoperability Discourse*, 2013, pp. 178–186.

[15] L. Dong and M. Lapata, "Language to logical form with neural attention," 2016, *arXiv:1601.01280*.

[16] S. Reddy, D. Chen, and C. D. Manning, "CoQA: A conversational question answering challenge," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 249–266, Nov. 2019.

[17] C. Hahn, F. Schmitt, J. J. Tillman, N. Metzger, J. Siber, and B. Finkbeiner, "Formal specifications from natural language," 2022, *arXiv:2206.01962*.

[18] A. Saparov and H. He, "Language models are greedy reasoners: A systematic formal analysis of chain-of-thought," 2022, *arXiv:2210.01240*.

[19] P. Srikumar, "Fast and wrong: The case for formally specifying hardware with LLMS," in *Proc. Int. Conf. Architectural Support Program. Lang. Operating Syst. (ASPLOS)*. New York, NY, USA: ACM Press, 2023.

[20] N. Tihanyi, T. Bisztray, R. Jain, M. A. Ferrag, L. C. Cordeiro, and V. Mavroeidis, "The FormAI dataset: Generative AI in so ware security through the lens of formal verification," in *Proc. PROMISE*, 2023, p. 33.

[21] Y. Charalambous, N. Tihanyi, R. Jain, Y. Sun, M. Amine Ferrag, and L. C. Cordeiro, "A new era in software security: Towards self-healing software via large language models and formal verification," 2023, *arXiv:2305.14752*.

[22] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in *Proc. 16th ACM Workshop Artif. Intell. Secur.*, Nov. 2023, pp. 79–90.

[23] S. Ouyang, J. M. Zhang, M. Harman, and M. Wang, "LLM is like a box of chocolates: The non-determinism of ChatGPT in code generation," 2023, *arXiv:2308.02828*.

[24] K. Krishna, Y. Chang, J. Wieting, and M. Iyyer, "RankGen: Improving text generation with large ranking models," 2022, *arXiv:2205.09726*.

[25] L. Li and M. Thornton, *Digital System Verification: A Combined Formal Methods and Simulation Framework*. Berlin, Germany: Springer Nature, 2022.

[26] Y. Wang, A. Gorski, and A. P. da Silva, "Development of a data-driven mobile 5G testbed: Platform for experimental research," in *Proc. IEEE Int. Medit. Conf. Commun. Netw. (MeditCom)*, Sep. 2021, pp. 324–329.

[27] J. Yang, S. Arya, and Y. Wang, "Formal-guided fuzz testing: Targeting security assurance from specification to implementation for 5G and beyond," 2023, *arXiv:2307.11247*.

[28] J. Yang and Y. Wang, "Formal and fuzzing amplification: Targeting vulnerability detection in 5G and beyond," 2023, *arXiv:2307.05758*.

[29] D. Dauphinais, M. Zylka, H. Spahic, F. Shaik, J. Yang, I. Cruz, J. Gibson, and Y. Wang, "Automated vulnerability testing and detection digital twin framework for 5G systems," in *Proc. IEEE 9th Int. Conf. Netw. Softwarization (NetSoft)*, Jun. 2023, pp. 308–310.

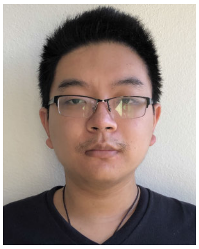
[30] *Radio Resource Control (RRC)*, document Ts 38.331, 3GPP, 2022.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

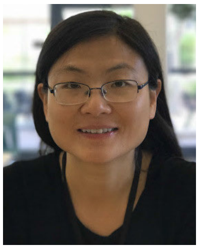
[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[34] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, and A. Askell, "Language models are few-shot learners," 2020, *arXiv:2005.14165*.



JINGDA YANG (Graduate Student Member, IEEE) received the B.E. degree in software engineering from Shandong University, and the M.Sc. degree in computer science from The George Washington University. He is currently pursuing the Ph.D. degree with the School of System and Enterprises, Stevens Institute of Technology. His research interests include formal verification and vulnerability detection of wireless protocol in 5G.



YING WANG (Member, IEEE) received the B.E. degree in information engineering from the Beijing University of Posts and Telecommunications, the M.S. degree in electrical engineering from the University of Cincinnati, and the Ph.D. degree in electrical engineering from the Virginia Polytechnic Institute and State University. She is currently an Associate Professor with the School of System and Enterprises, Stevens Institute of Technology. Her research interests include cybersecurity, wireless AI, edge computing, health informatics, and software engineering.