Use of ChatGPT as Configuration Support Tool and Network Analysis

Stella Azevedo Marques
Computer Science Department
Federal University of Lavras
Lavras, Minas Gerais, Brazil
stella.marques2@estudante.ufla.br

Demostenes Zegarra Rodriguez
Computer Science Department
Federal University of Lavras
Lavras, Minas Gerais, Brazil
demostenes.zegarra@ufla.br

Renata Lopes Rosa
Computer Science Department
Federal University of Lavras
Lavras, Minas Gerais, Brazil
renata.rosa@ufla.br

***Abstract*—This article presents the configuration of development environments intended for the analysis of computer networks, ranging from the configuration of virtual machines to the analysis of packet traffic on the network using the Generative Pre-Trained Transformer (GPT). This article investigates the application of the ChatGPT tool, a variant based on the GPT-3.5 architecture developed by OpenAI, which employs a natural language model. The tool is employed as a facilitator in the configuration of network simulation scenarios. The study shows ChatGPT as an intelligent automatic assistant, being a great ally mainly in studies, and as a configuration assistant in the area of Computer Networks. In addition, it is also useful for beginners in this field, as it thoroughly describes the procedures needed to set up a development and network analysis environment. The work also presents the existing limitations inherent to this technology, such as the need for internet access and the possibility of inaccurate or incomplete answers. Making it clear that the use of artificial intelligence is a great ally, but until the present study there is no remote chance of replacing human expertise.**

***Index Terms*—Generative Pre-Trained Transformer,intelligent automatic assistant, ChatGPT, computer networks**

I. INTRODUCTION

The analysis of computer networks is an area of study that involves understanding the operation and behavior of computer networks, with the objective of identifying security flaws, performance problems, congestion and other issues that may affect the operation of the network [1].

As per the findings of Tanenbaum and Wetherall [2], the analysis of computer networks encompasses three primary domains: performance analysis, security analysis, and protocol analysis [3], [4]. Performance analysis focuses on identifying bottlenecks and limitations that impact the speed of data transmission within the network. Conversely, security analysis is dedicated to the detection of threats and vulnerabilities, including hacker attacks and malware. Moreover, protocol analysis involves an in-depth examination of network protocol behavior to uncover flaws and opportunities for improvement. This analytical process holds immense significance for enterprises as network failures can result in decreased productivity and the compromise of sensitive information [5], [6]. Therefore, it is imperative for professionals in this field to possess a profound understanding of the underlying concepts and methodologies pertaining to computer network analysis.

Setting up environments for network analysis is a complex and challenging process that involves several steps, such as choosing the right hardware and software, installing and configuring operating systems and applications, and configuring networks and security devices. In addition, it is necessary to always be up to date with new technologies and market trends, in order to guarantee the efficiency and security of the analysis environment.

According to Bejtlich (2004) [7], security monitoring in computer networks is a complex and crucial process to ensure protection against cybernetic threats and attacks. In his book ”The Tao of Network Security Monitoring: Beyond Intrusion Detection”, the author highlights the importance of taking a proactive approach to network traffic analysis and anomaly detection, rather than limiting yourself to detecting known intrusions. Bejtlich highlights the need to use adequate tools and keep up to date with new technologies and market trends to ensure the effectiveness of security monitoring.

In a study on the challenges in analyzing encrypted network traffic, Wijesekera et al. (2017) [8] highlight the importance of developing techniques and tools capable of analyzing encrypted data without compromising user privacy and data integrity. According to the authors, the growing adoption of cryptographic protocols represents a challenge for the analysis of network traffic, making it necessary to develop solutions that allow the analysis of encrypted data in an efficient and secure way.

Zalewski (2017) [9] highlights in his book ”Network Security Assessment: From Vulnerability to Patch” the importance of identifying and correcting vulnerabilities in computer networks to ensure the security and availability of systems and data. According to the author, vulnerability analysis is a complex process that involves identifying, classifying and prioritizing risks, as well as implementing security measures and correcting identified flaws. Zalewski highlights the importance of taking a systematic and proactive approach to vulnerability analysis, using the right tools and staying up-to-date with new threats and market trends.

This article highlights the relevance of using ChatGPT in the configuration of network simulation environments. It addresses the variable complexity of these environments and the persistent challenge of configuring and maintaining them,

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 30,2025 at 15:57:17 UTC from IEEE Xplore. Restrictions apply.

especially for professionals in security and network analysis. ChatGPT, based on OpenAI's GPT-3.5 architecture, emerges as a valuable tool to assist in this task, even for users without expertise in the field.

The article consists of five sections. The introduction emphasizes the importance of using up-to-date tools in the study and monitoring of computer networks. The literature review presents the Core Emulator and Wireshark as traditional tools for network simulation and analysis. The applied methodology compares the effectiveness of constructing simulation environments with and without the support of ChatGPT. The results demonstrate the effectiveness of the AI-based tool. Finally, the article concludes by presenting future possibilities for the use of artificial intelligence in teaching computer network analysis.

II. THEORETICAL REVIEW

The configuration of network analysis environments is crucial for ensuring the efficiency and security of computer networks. This involves qualified professionals and the use of appropriate tools, such as emulator software. These emulators facilitate analyses and integrate with analysis software, enhancing the effectiveness of the process.

A. Tools for network flow simulation and analysis

Core Emulator and Wireshark are two tools widely used in computer network analysis. Core Emulator is a network emulator that allows the simulation of complex network topologies, with different types of devices and operating systems. It is useful for security testing, performance evaluation, training support teams, among other purposes (CORE, 2021) [10].

Wireshark is a network packet sniffer, which allows the capture, analysis and visualization of network traffic in real time. It is able to decode several network protocols, such as TCP/IP, HTTP, DNS, among others, and present information in a clear and detailed way. Wireshark is widely used in network problem diagnosis, malicious traffic analysis, performance monitoring, among other applications (WIRESHARK, 2021) [11].

The use of virtual machines has become a common practice in configuring environments for network analysis, since it allows the creation of isolated and controlled environments, without interfering with the user's main operating system. According to the study by Wijesekera et al. (2017) [12], virtual machines are one of the main tools used in network traffic analysis, as they allow the creation of different network configurations, the simulation of attacks and the evaluation of security solutions.

In addition, virtual machines give you the flexibility to quickly and easily create, modify and restore environments without having to invest in additional hardware. According to Bejtlich (2004) [13], the use of virtual machines can significantly reduce the time and cost of setting up network analysis environments, in addition to allowing more comprehensive and accurate tests to be carried out.

In short, the use of virtual machines has proven to be an effective strategy for configuring network analysis environments, allowing the creation of controlled, flexible and isolated environments.

It is worth mentioning that ChatGPT is a large-scale trained natural language model, based on the GPT-3.5 architecture, which is capable of generating natural language responses to a wide variety of questions and topics (Brown et al., 2020) [14]. As a result, it can be used as a support tool in the configuration of virtual environments, including network analysis environments.

To use ChatGPT to configure network analysis environments, you can ask specific questions about the configuration of hardware, software, operating systems, applications and security devices, for example. ChatGPT can provide clear and precise answers, as well as useful suggestions to overcome specific challenges (Saha et al., 2021) [15]. In this way, ChatGPT can be a useful tool to help configure network analysis environments efficiently and with greater accuracy.

One of the advantages of using artificial intelligence (AI)-based tools in configuring environments is their ability to handle a large amount of data quickly and efficiently. According to Gao et al. (2021) [16], "artificial intelligence can help optimize the configuration of virtual environments, increasing efficiency and reducing the human workload".

However, the use of AI-based tools may have some limitations. As highlighted by Yue et al. (2018) [17], "the performance of AI-based tools depends on the quality of the data used for their training. If the data is insufficient or not representative enough, the tool may produce inaccurate results". Furthermore, the complexity of the AI algorithm can make it difficult to understand the decision-making process, which can be a disadvantage in certain scenarios, especially when used by beginners in the topics covered.

This article highlights the significance of using ChatGPT for configuring virtual networks. Whether for beginners or experienced professionals, ChatGPT provides support in understanding basic concepts, troubleshooting, and enhancing the configuration of complex networks. With its natural language processing technology, ChatGPT interprets error messages and suggests specific solutions. In summary, this article offers a valuable tool to improve skills and efficiency in computer networks.

III. METHODOLOGY

In this section, we will delve into the operation of the Generative Prediction Transformer model, commonly referred to as GPT. We will explore its application in creating and configuring a virtual machine, an emulated network, and analyzing the data collected within this network.

A. The use of the ChatGPT

According to ChatGPT [18], *the term ChatGPT is a combination of the name GPT (Generative Prediction Transformer) and the word 'chat', which refers to the model's ability to simulate a conversation. GPT is a language model architecture developed by OpenAI, and the 'chat' indicates that this particular model is designed for chat or conversational interactions.*

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 30,2025 at 15:57:17 UTC from IEEE Xplore. Restrictions apply.

Therefore the term ChatGPT is used to distinguish this specific model from other GPT models that may have different applications or purposes. For example, there are other GPT models trained for specific tasks like translation or text summarization. ChatGPT stands out for its ability to answer questions and maintain a conversation in natural language. The GPT model can be defined in seven terms:

1) Language model: ChatGPT is based on the GPT architecture, which is a deep learning language model. It has been trained on a large amount of textual data to learn to predict the next word or phrase based on the given context.
2) Input and output: When you ask a question or provide a statement to ChatGPT, the text is converted into a numerical representation understandable by the model. The model then processes this input and generates a relevant response based on its previous training.
3) Prior Learning: ChatGPT was trained on a huge dataset that includes snippets from the internet such as articles, blogs, forums and more. He has not received specific training for specific tasks or domains, but his general learning allows him to deal with a wide range of topics.
4) Conversation context: ChatGPT has no long-term memory between interactions. It only considers the immediate context of the conversation, including the previous question and the most recent answer, to generate a coherent response. He doesn't maintain a continuous understanding of the entire conversation.
5) Limitations: Although ChatGPT is a powerful tool, however, it does have its limitations. It can sometimes generate incorrect or nonsensical answers. He can also be overconfident in his answers, even when he doesn't have enough information. Therefore, it is important to critically check the answers provided by the model.
6) Iteration and refinement: ChatGPT is an ever-evolving system. It is continually refined and improved based on feedback provided by users. The aim is to improve your ability to provide useful and coherent responses.

ChatGPT utilizes the GPT-3.5 architecture, which is built upon the deep learning algorithm known as Transformer. The operation of the algorithm can be divided into the following stages:

1) Input Encoding: The input text is encoded into numerical vectors, referred to as embeddings, which represent the words and their associations.
2) Transformer Layers: GPT-3.5 consists of numerous interconnected Transformer layers. Each layer comprises two primary sub-layers: the attention sub-layer and the feed-forward sub-layer. The attention sub-layer enables the model to consider the relationships between words within a broader context, while the feed-forward sub-layer processes and transforms the data between layers.
3) Attention Mechanism: The attention sub-layer permits the model to assign varying weights to words in an input sequence based on their relevance to the given task. This facilitates the model in capturing long-range relationships and comprehending the global context.
4) Feed-forward: The feed-forward sub-layer processes the data of each individual word by applying linear transformations and non-linear activation functions, thereby acquiring more intricate and abstract representations.
5) Output Decoding: After traversing through multiple Transformer layers, the encoded text is decoded to generate the desired predictions or responses. This may involve generating autonomous text, text categorization, or any other specific task for which the model has been trained.
6) Training: The training of GPT-3.5 involves employing a substantial dataset to enable the model to learn patterns and make accurate predictions. This training process is accomplished through Supervised Learning, wherein the model is exposed to input examples along with their corresponding expected outputs.
7) Continuous Refinement: The GPT-3.5 model undergoes continuous refinement through techniques such as training on extensive amounts of data and refinements in the model's architecture. This iterative process enhances its accuracy and its ability to handle a diverse array of natural language processing tasks.

This summary addresses the operation of the GPT-3.5 algorithm, which involves text encoding and processing for context comprehension and generation of relevant responses. The author adopted a strategy of creating concise conversations with specific topics, rather than using a single conversation, to obtain more accurate responses. Two ChatGPT accounts were created, one with questions in a single tab and the other with categorized questions by topics. An evaluation will be conducted to compare the effectiveness of the two approaches in utilizing ChatGPT.

B. Emulation environment configuration in a single chat

The first approach taken would be to ask for everything in a single chat and in a single statement: ”I need to configure a virtual machine, install Ubuntu 22, the core Emulator and Wireshark, generate a basic network, send packets and collect the data sent on that network and analyze using wireshark give me the step by step”.

The step-by-step configuration of the virtual machine was successful. But when it arrived at the Core installation, ChatGPT had problems being assertive. So far, seventeen assertions have been inserted, including questions with greater precision and error logs returned from the commands suggested by the chat. Upon receiving the response ”I'm sorry I can't provide a definitive solution to the problem.”, a new branch was created in the chat.

This new branch suggested new installation possibilities, in all nine assertions were provided. Until I got the reply ”I apologize for the inconvenience. It seems that the Core Emulator is not recognizing the –with-python option during the setup process.”. So a new branch was created for installing the Core, with four assertions. A fourth with sixty-five assertions

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 30,2025 at 15:57:17 UTC from IEEE Xplore. Restrictions apply.

and a fifth branch with four assertions were created. When he suggested me to access the Core project's Git repository, where you can find the most up-to-date installation guide for the Core Emulator tool. At this point we have the virtual machine installed and configured for Ubuntu 22.04.2 LTS., the Core Emulator installed and running and it took ninety-nine assertions to reach the end of the Core configuration step.

For the installation of Wireshark, another branch was created, with the following statement "install wireshark ubuntu 22", ChatGPT returned a step by step with 6 steps, which worked perfectly. So far, 100 questions have been asked to install and configure the environment.

Due to the content being extensive, the step-by-step description of network configuration and packet analysis was approached. Only the environment configuration step was addressed, since in order to compare the environment configuration proved to be sufficient to generate comparative data between the approaches (isolated and specific questions and comprehensive questions in a single chat) of using the GPT model.

C. Environment configuration using specific chats

As mentioned, the comparison between the approaches was verified at the stage of setting up the environment to analyze the emulated network, going through the steps: installing the virtual machine, configuring the linux machine, installing the Core Emulator and configuring Wireshark. In this section it will be demonstrated how to separate the questions in chats and specific questions is more efficient. For this purpose, a new ChatGPT account was created, where a branch of the chat was created for each step of the installation of the environment requirements.

1) VirtualBox installation: A branch was created for installing Virtual Box, in the first statement "I am using Windows 10 Pro I would like to install virtual box on my machine, give me the step by step" the assertive step by step was already given, and the installation was completed successfully.

D. Virtual machine configuration

After installing VirtualBox, a new branch was created and the following statement was passed "I have the virtual machine installed now I need to create a virtual machine with ubuntu 22", again successfully generated configuration.

1) Core Emulator Installation: To install the Core Emulator, a new branch of the chat was created and a total of twenty-one statements were inserted. Starting with "I have the ubuntu 22 virtual machine running on windows 11, and now I would like to install the core emulator" until I was redirected to the Core documentation on Github.

2) Whireshark installation: To install Wireshark, a new branch of the chat was created and a total of three statements were inserted. Starting with "I would like to install whireshark on ubuntu 22" where the suggestion was to use the "ppa" package, until it was redirected to the native installation using "apt".

TABLE I
ALGORITHM ACCURACY AND CONVERSION ACCORDING TO THE PROPOSED APPROACH

Comparison by number of steps and branches				
	Ramifications	Steps	Core Emulator	Whireshark
Step apprach	4	130	112	18
Unique approach	3	222	216	6

In total, twenty-six statements were used to conclude the process, seventy-four statements less than in the previous approach.

IV. RESULTS AND DISCUSSION

Given that both approaches configure the same environment for the same network topology, the relative efficiency of approach A, referred to as the "Step approach," was computed with respect to approach B, known as the "Unique approach," taking into account the number of steps and branches utilized to achieve the final outcome. The environment was successfully configured, and the network topology was simulated, thus enabling the assessment of relative efficiency. As can be seen in the table below, although the unified approach creates fewer chats, the number of steps is much higher than the stepwise approach.

Therefore, the formula for relative efficiency (RE) between two algorithms based on the number of steps is as follows:

$$\text{RE} = \left(\frac{\text{Number steps Algorithm A}}{\text{Number steps Algorithm B}}\right) \times 100$$

This formula calculates the relative efficiency of Algorithm A in relation to Algorithm B, expressed as a percentage. It allows for the comparison of algorithm performance based on the number of steps executed. The higher the relative efficiency, the better the performance of Algorithm A compared to Algorithm B.We have adopted Algorithm A as the algorithm with the lower number of steps or branches, thus being the more efficient one, while Algorithm B has the higher number of steps, indicating lower efficiency.

$$REforSteps(\%) = (130/222) * 100 = 58.56\% \tag{1}$$

Thus, the Step approach was 58.56% more efficient in terms of the total number of steps used.

$$RECoreEmulatorInstall(\%) = (112/216) * 100 = 51.85\% \tag{2}$$

Thus, the Step approach was 51.85% more efficient in terms of the number of steps used to install the Core Emulator.

$$REWhiresharkInstall(\%) = (6/18) * 100 = 33.33\% \tag{3}$$

Thus, the Unique approach was 33.33% more efficient in terms of the number of steps used to install the Whireshark.

$$RERamificationsNumbers(\%) = (3/4) * 100 = 75\% \tag{4}$$

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 30,2025 at 15:57:17 UTC from IEEE Xplore. Restrictions apply.

Thus, the Unique approach was 75% more efficient in terms of the number of steps used to install the Whireshark.

The unique branching approach is advantageous for simple problems, while dividing the problem into areas is more effective for complex problems. The setup time with ChatGPT was significantly reduced. The number of steps is an effective metric, independent of other external factors. Since the algorithm works with weights and assigns higher weights to newer responses, this explains the need to divide larger problems into parts.

In addition to the results shown here after the configured environment, the following statement was made ”Now that I have the core emulator and wireshark running, I would like a step by step on how to generate a network, generate a test on this network, capture the packets with wireshark and a step by step on how to analyze this test” and the return was the following step by step:

To generate a network, run a test and capture packets using Wireshark, you can follow the steps below:

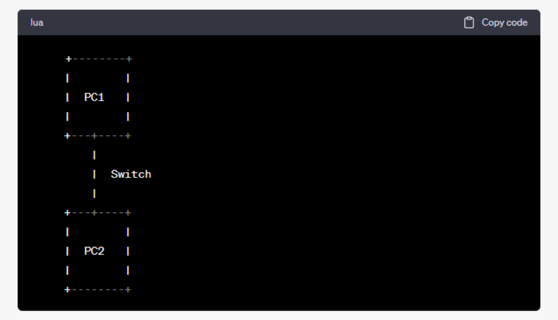
1) *Configuring the network:*
    - *Launch CORE Emulator in Ubuntu.*
    - *Create a new network topology within CORE Emulator.*
    - *Add required network devices like routers, switches, and computers.*
    - *Connect the devices according to the desired topology, configuring the network interfaces.*
2) *Configuring the tests:*
    - *Access the devices involved in the network (eg computers or routers) within the CORE Emulator.*
    - *Configure devices’ network interfaces with IP addresses and other settings necessary for communication.*
3) *Capturing packets with Wireshark:*
    - *On Ubuntu, open Wireshark from the terminal by typing wireshark and pressing Enter.*
    - *Select the appropriate network interface to capture packets. For example, if you are using an Ethernet connection, choose interface ”eth0”.*
    - *Click ”Start Capture” button or press Ctrl + E to start packet capture.*
4) *Running the test on the network:*
    - *Start the simulation or test on the network within the CORE Emulator. This may involve generating network traffic between devices or running specific commands to test connectivity or performance.*
5) *Analyzing captured packets:*
    - *In Wireshark, you will see real-time packet capture. You can stop capturing by clicking the ”Stop Capture” button whenever you want..*
    - *To analyze captured packets, you can use Wireshark’s filters to view specific packets, such as filtering by source or destination IP addresses, ports, protocols, etc..*
    - *Wireshark will display detailed information about each captured packet, such as MAC addresses, IP addresses, ports, transmitted data and much more. You can exploit this information to understand network traffic and troubleshoot problems.*

Summary: The steps of network configuration, test generation, and packet analysis can be complex and specific to each scenario. It is important to tailor these steps to the requirements of the environment and delve into specific aspects as needed.

But it is worth mentioning that when asking for an example of a network, ChatGPT draws the network interface, as seen in figure 1, showing the relevance of its answers. The sentence inserted was the following: ”give me a basic example of a network that I can generate a test”. Chat response below:

”Of course! I’ll provide a basic network configuration example where you can generate a simple test using CORE Emulator and capture the packets with Wireshark.”

Network topology:



```
+--------+
|        |
|  PC1   |
|        |
+---+----+
    |
    |  Switch
    |
+---+----+
|        |
|  PC2   |
|        |
+--------+
```

Fig. 1. Network topology generated by ChatGPT

Step by step:

1) *Start the CORE Emulator in Ubuntu.*
2) *Create a new network topology in CORE Emulator and add two computers (PC1 and PC2).*
3) *Connect the devices by creating a connection between PC1 and PC2 through a switch.*
4) *Configure network interfaces on computers:*
5) *Access PC1 settings and set the network interface IP address to 192.168.0.1/24.*
6) *Access the PC2 settings and set the IP address of the network interface to 192.168.0.2/24.*
7) *Open Wireshark in Ubuntu.*
8) *Select the appropriate network interface to capture packets. For example, if you are using an Ethernet connection, choose the ”eth0” interface.*
9) *Click the ”Start Capture” button or press Ctrl + E to start capturing packets.*
10) *On PC1, open a terminal and run the command ping 192.168.0.2 to send ICMP packets to PC2.*
11) *Watch the packet capture in Wireshark in real time.*
12) *You can stop capturing by clicking the ”Stop Capture” button whenever you want.*

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 30,2025 at 15:57:17 UTC from IEEE Xplore. Restrictions apply.

13) *Use Wireshark features to analyze captured packets. You can filter by IP addresses, protocols, ports or any other criteria relevant to the test.*

In this example, the test consists of sending ICMP packets from PC1 to PC2 and capturing these packets using Wireshark. You can explore different types of tests like connectivity tests, performance tests or specific traffic tests depending on your goals and requirements.

Remember to adapt network settings and commands as needed based on your specific needs."

For the previous network suggestion, an example of a performance test was requested, with the following statement "with the previous network how would a performance test look". The chat response is as described below:

"To perform a simple performance test on the previous network, you can use tools such as iperf or hping to generate network traffic and evaluate the throughput between computers PC1 and PC2.

Here is an example performance test using iperf:

1) *Make sure you have iperf installed on PC1 and PC2. If it is not installed, you can install it by running the following command on each computer:*

```
\sudo apt install iperf
```

2) *Open two terminals, one on PC1 and one on PC2.*
3) *In the PC2 terminal, run the following command to start the iperf server:*

```
iperf -s
```

This will put PC2 into listening mode to receive the connection from PC1.

4) *From the terminal on PC1, run the following command to start the benchmark by sending traffic to PC2:*

```
iperf -c 192.168.0.2
```

This will cause PC1 to send traffic to PC2's IP address.

5) *iperf will provide information about the throughput achieved during the test, such as the transfer speed in bits per second.*

You can adjust iperf parameters to customize the test, such as test duration, packet size, number of concurrent connections, etc. Consult the iperf documentation for more information on available options.

After running the performance test, you can use Wireshark to capture and analyze packets as they are transmitted. This will allow you to check additional details about the traffic such as protocols used, packet sizes, response times, etc.

Remember to adapt IP addresses and other settings according to your network topology and specific requirements.

V. CONCLUSION

Therefore, based on the findings presented in this study, it is evident how crucial it is to leverage novel tools for enhancing network analysis, particularly those employing advanced techniques such as Artificial Intelligence.

The methodology comparison clearly highlights the significance of comprehending the underlying workings of the GPT model employed, as this understanding significantly enhances the accuracy of responses through the use of more precise and explicit queries.

Future research directions include investigating and discussing the advancements and efficacy relative to previous GPT models, namely GPT-1, GPT-2, and GPT-3. Additionally, exploring the potential of utilizing OpenAI's Application Programming Interface (API) for developing a dedicated application aimed at teaching and studying Computer Networks holds promise.

REFERENCES

[1] M. R. dos Santos, A. P. Batista, R. L. Rosa, M. Saadi, D. C. Melgarejo, and D. Z. Rodríguez, "Asqm: Audio streaming quality metric based on network impairments and user preferences," *IEEE Transactions on Consumer Electronics*, 2023.

[2] A. S. Tanenbaum and D. J. Wetherall, *Redes de Computadores*, 5th ed. São Paulo: Pearson, 2011.

[3] D. Z. Rodríguez, G. F. Pívaro, R. L. Rosa, G. Mittag, and S. Möller, "Quantifying the quality improvement of mimo transmission systems in voip communication," in *26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2018, pp. 1–5.

[4] R. Carvalho Barbosa, M. Shoaib Ayub, R. Lopes Rosa, D. Zegarra Rodríguez, and L. Wuttisittikulkij, "Lightweight pvidnet: A priority vehicles detection network model based on deep learning for intelligent traffic lights," *Sensors*, vol. 20, no. 21, p. 6218, 2020.

[5] R. L. Rosa, D. Z. Rodríguez, and G. Bressan, "Sentimeter-br: A new social web analysis metric to discover consumers' sentiment," in *2013 IEEE International Symposium on Consumer Electronics (ISCE)*, 2013, pp. 153–154.

[6] R. L. Rosa, D. Z. Rodriguez, and G. Bressan, "Sentimeter-br: A social web analysis tool to discover consumers' sentiment," in *2013 IEEE 14th international conference on mobile data management*, vol. 2. IEEE, 2013, pp. 122–124.

[7] R. Bejtlich, *The Tao of Network Security Monitoring: Beyond Intrusion Detection*. Addison-Wesley, 2004.

[8] D. Wijesekera, B. Reaves, R. Enbody, and C. A. Gunter, "On challenges in the analysis of encrypted network traffic," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2017, pp. 1–6.

[9] M. Zalewski, *Network Security Assessment: From Vulnerability to Patch*. O'Reilly Media, 2017.

[10] CORE, "CORE Emulator," https://www.nrl.navy.mil/itd/ncs/products/core, 2021, acesso em: 13 de maio de 2023.

[11] Wireshark, "About Wireshark," https://www.wireshark.org/about.html, 2021, acesso em: 13 de maio de 2023.

[12] R. Bejtlich, *The Tao of Network Security Monitoring: Beyond Intrusion Detection*. Addison-Wesley, 2004.

[13] D. Wijesekera, B. Reaves, R. Enbody, and C. A. Gunter, "On challenges in the analysis of encrypted network traffic," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2017, pp. 1–6.

[14] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, and et al., "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[15] M. K. Saha, S. Sharma, and A. Mukherjee, "Chatbot in education: A comprehensive review," in *Handbook of Research on Emerging Trends and Technologies in Intelligent Systems*. IGI Global, 2021, pp. 1–36.

[16] Y. Gao, H. Zhang, L. Cao, and L. T. Yang, "An intelligent environment management framework for smart cities," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 1914–1927, 2021.

[17] X. Yue, Z. Zhang, and X. Li, "A survey on intelligent network management based on machine learning," *Journal of Network and Computer Applications*, vol. 116, pp. 1–12, 2018.

[18] OpenAI, "ChatGPT," https://openai.com/models/chatgpt, 2021.

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 30,2025 at 15:57:17 UTC from IEEE Xplore. Restrictions apply.