Expert Systems With Applications 261 (2025) 125509

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Evaluation of LLM-based chatbots for OSINT-based Cyber Threat Awareness

Samaneh Shafee *, Alysson Bessani, Pedro M. Ferreira

LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal

ARTICLE INFO

Keywords:
Cyber Threat Intelligence
Open-Source Intelligence
Natural Language Processing
Large Language Models
Chatbots

ABSTRACT

Knowledge sharing about emerging threats is crucial in the rapidly advancing field of cybersecurity and forms the foundation of Cyber Threat Intelligence (CTI). In this context, Large Language Models (LLMs) are becoming increasingly significant in the field of cybersecurity. This study surveys the performance of ChatGPT, GPT4all, Dolly, Stanford Alpaca, Alpaca-LoRA, Falcon, and Vicuna LLM-based chatbots in binary classification and Named Entity Recognition (NER) tasks using Open-source intelligence (OSINT) to detect and extract structured data about cybersecurity threats. We utilize data collected in previous research from Twitter to assess the competitiveness of these chatbots when compared to specialized state-of-the-art models trained for those tasks. In binary classification experiments, the commercial chatbot GPT-4 achieved an acceptable F_1 score of 0.94, and the open-source GPT4all achieved an F_1 score of 0.90. However, when applied for cybersecurity entity recognition, all evaluated LLM-based chatbots have limitations and are less effective. This study demonstrates the capability of LLM-based chatbots for OSINT processing and shows that they require further improvement in NER to effectively replace specially trained models. Our results highlight their strengths and limitations compared to specialized models. This provides insights for researchers to enhance LLM-based chatbots to reduce the effort required to integrate machine learning in OSINT-based CTI tools.

1. Introduction

Cybersecurity is a continuously evolving domain that involves experts publicly sharing their knowledge on cyber threats. This information is the primary source for Cyber Threat Intelligence (CTI) tools, and researchers have contributed to developing methods for extracting cyber threat intelligence from text sources (Alves, Bettini, Ferreira, & Bessani, 2021; Liao et al., 2016; Ritter et al., 2015). With the recent developments of Machine Learning (ML), significant advancements have been made in the field of cybersecurity. ML-based tools have deepened our understanding of cyber threats, enabling innovative and responsive solutions. A notable development in this domain is the emergence of Large Language Models (LLM), representing a substantial breakthrough in Natural Language Processing (NLP). These advancements have led to the creation of LLM-based chat assistants (chatbots) (Zheng et al., 2024) with instruction following and conversational capabilities. In this paper, we evaluate the chatbots' prompt answering capability to make yes/no decisions, and since we use the complete chatbot application for all considered chatbots, we collectively refer to them as LLM-based chatbots.

Such advancements are exemplified by academic works such as SecBot (Arora, Arora and McIntyre, 2023; Franco et al., 2020), which provide proof-of-concept chatbots to support cybersecurity planning and management, and recent high-profile products such as Microsoft Security Copilot (Microsoft, 2023). These efforts (among others, described in Section 2) highlight the practical use of chatbots to address cybersecurity and CTI challenges.

In today's cybersecurity landscape, the timely detection and response to emerging threats are crucial. To address this requirement, we focus on understanding the potential of LLM-based chatbots to improve cyber threat awareness and streamline the detection processes. Specifically, we investigate using LLM-based chatbots, such as the now ubiquitous ChatGPT, for binary classification and Named Entity Recognition (NER) tasks for processing cybersecurity information. The binary classification and NER are fundamental to *filter* cybersecurity-relevant information among the vast amount of OSINT data gathered from internet and *identify* key terms in the text (e.g., vulnerability type, affected products, type of attack), respectively. Together, these tasks can be put together in a OSINT processing pipeline where raw OSINT is obtained from the internet and processed to generate timely alerts or even Indicators of Compromise (IoCs).

This integration of LLM-based chatbots in this type of pipeline would enhances the field's capabilities, enabling organizations to strengthen their threat awareness without the burden of maintaining specially-crafted models. In particular, an LLM-based chatbot does not

* Corresponding author.
E-mail addresses: sshafee@ciencias.ulisboa.pt (S. Shafee), anbessani@ciencias.ulisboa.pt (A. Bessani), pmf@ciencias.ulisboa.pt (P.M. Ferreira).

https://doi.org/10.1016/j.eswa.2024.125509
Received 11 May 2024; Received in revised form 19 July 2024; Accepted 4 October 2024
Available online 11 October 2024

need to be re-trained by the pipeline owner. The important question remains of whether such chatbots can achieve the level of performance of specialized models (Minaee et al., 2021), which are known to achieve outstanding results using a wide variety of techniques for binary classification (Li et al., 2022), and NER (Jehangir, Radhakrishnan, & Agarwal, 2023) tasks.

This paper presents an empirical study that uses inductive reasoning and a comparative methodology to answer the following research question: *Are LLM-based chatbots competitive with state-of-the-art specialized models for detecting OSINT CTI and extracting pertinent information?* To answer this question, we use a publicly-available annotated cybersecurity dataset (Alves et al., 2021; Dionísio, Alves, Ferreira, & Bessani, 2019) collected from Twitter (currently named X),[1] a known reliable OSINT CTI source (Alves, Andongabo, Gashi, Ferreira, & Bessani, 2020), to produce the empirical results and inductively assess the performance of several LLM-based chatbots. Then, a comparative analysis of previous works using specialized models and this paper's results effectively answers the research question, allowing the discussion of the conclusion's implications and outlining possibilities for further research and improvements.

Our research specifically concentrates on GPT-style chatbots. In this category we include ChatGPT (OpenAI Platform, 2023), GPT4all (Anand, Nussbaum, Duderstadt, Schmidt, & Mulyar, 2023), Dolly (Conover et al., 2023), Stanford Alpaca (Taori et al., 2023), Alpaca-LoRA (Wang, 2024), Falcon (Falcon LLM, 2023), and Vicuna (Chiang et al., 2023). These systems represent the two prominent types of LLM-based chatbots widely used nowadays: commercially available as a service through APIs and open-source, built to run on local GPU servers. When planning the experiments, we picked all the available variants of GPT-like chatbots since they are standard in the field (López Espejel, Ettifouri, Yahaya Alassan, Chouham, & Dahhane, 2023).

Since utilizing LLM-based chatbots raises additional technical questions, we also study the impact of different utilization methods on the empirical results obtained to answer the main research question. These questions concern the capability to provide clear yes-or-no answers, and the cost (in terms of processing time) required to perform the NLP tasks.

Our contributions can be summarized as follows:

1. We present a state-of-the-art survey on the application of LLM-based chatbots to cybersecurity.
2. We investigate the extent to which the inherent flexibility of LLM-based chatbots can be tailored to meet the specific requirements of OSINT-based CTI applications.
3. The study provides a comparative analysis of the practical use and performance of LLM-based chatbots in specialized CTI tasks, including binary text classification and NER.

The remainder of this paper is organized as follows. Section 2 provides background on LLMs, discusses related works on LLM-based chatbots for cybersecurity, and their evaluation in NLP tasks. Section 3 presents a deep exploration of LLM-based chatbots and highlights their significance and capabilities. In Section 4, we shift our focus to the evaluation methodology, detailing our dataset, methods, and the comparison criteria employed. Section 5 explores strategies aimed at optimizing the utilization of these chatbots, including prompt fine-tuning and text length control. Section 6 presents our main experimental results which are discussed in Section 7. Finally, we conclude the paper in Section 8 by summarizing the key contributions and insights derived from this study.

[1] Since the dataset was collected and published before the name change, we will keep the name Twitter in this paper.

2. Background and related work

In this section, we briefly present the required background for this paper. We start by analyzing transformer models, the foundational elements facilitating recent progress in NLP. Next, we discuss LLMs by examining their development, abilities, and significant influence on various specific areas. Building on this foundation, we review research on how LLMs can effectively be utilized for cybersecurity-related NLP, exploring in detail the role of OSINT in cybersecurity. We end the section by examining LLM-based chatbots' evolution and NLP capabilities to address cybersecurity concerns and the literature gap we aim to contribute with this paper.

2.1. Transformers

The introduction of transformers (Vaswani et al., 2017) revolutionized the field of NLP, as they became the preferred architecture for various NLP tasks due to their ability to effectively capture the extensive dependencies and contextual associations of textual data (Choi & Lee, 2023; Lin, Wang, Liu, & Qiu, 2022). This ability enables transformers to overcome the limitations of previous methods such as Recurrent Neural Networks (RNNs) (Medsker & Jain, 2001), Convolutional Neural Networks (CNNs) (Kim, 2015), and Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997). Instead of processing inputs one at a time, transformer models handle tokens or words simultaneously. This makes it easier to model the global interactions (Farooq, Awais, Ahmed, & Kittler, 2021; Sanford, Hsu, & Telgarsky, 2024) and dependencies (Lin et al., 2022). This feature makes transformers highly effective for various tasks, including but not limited to machine translation, sentiment analysis, and text generation. Transformers have emerged as the basic foundation for models such as Bidirectional Encoder Representations Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2019) and Generative Pre-trained Transformers (GPT) (Radford, Narasimhan, Salimans, Sutskever, et al., 2018). These models have demonstrated exceptional performances across diverse NLP benchmarks, leading to progress in language understanding, text generation, and question answering (Min et al., 2023).

2.2. Large language models

LLMs have emerged as significant innovations that revolutionize the human language process, generation, and understanding. These models are trained on large text datasets, including immense volumes of language data, which enables them to perform tasks that demand contextual comprehension and generate coherent and meaningful responses (Yang et al., 2024). In terms of applications, the latest generation of language models has been applied across diverse fields. For instance, the utilization of LLMs within mathematics, physics, and chemistry problem-solving has been evaluated by Arora, Singh and others (2023). Agrawal (2023) evaluated the ability of LLMs for human-like reasoning tasks. The findings of the study indicate that LLMs have strong abilities in analogical and moral reasoning but face challenges in spatial reasoning tests. Chatbots powered by LLMs have attracted interest as powerful tools for data annotation in NLP domains (Ding et al., 2023). This interest arises from chatbots' proficiency in language tasks and the critical role of data annotation in developing NLP systems. An illustrative study (Gilardi, Alizadeh, & Kubli, 2023) compared ChatGPT with human crowd workers in annotation tasks. The research emphasized that ChatGPT surpassed human workers in terms of performance, agreement among annotators, and cost-effectiveness. Moreover, the Self-HWDebug framework (Akyash & Kamali, 2024) leverages instruction-tuned LLMs to automatically generate debugging instructions for hardware vulnerabilities. This innovative approach reduces human intervention and improves the quality of the debugging process by extending solutions across related security issues.

There are two well-known broad families of LLMs: GPT and BERT. While these may be the most frequently used, many other variations remain relevant. GPT-style models utilize a regressive transformer architecture to capture contextual dependencies and relationships within text (Yang et al., 2024), while BERT-style language models utilize a bidirectional transformer architecture and adopt a masked language modelling objective. Both approaches gained popularity for NLP capabilities because of their ability to model contextual dependencies. Besides the difference in the context they consider to make text predictions, they also differ in the number of parameters and training dataset size, with GPT-style models requiring more resources in both cases.

Yang et al. (2024) provides a complete guide explaining how to use LLMs from various perspectives, including model selection, data consideration, and task specificity. It thoroughly investigates the details of pre-training and the significance of the training and test data and offers insights into tasks that require a rich knowledge base, natural language comprehension, generation capabilities, and other emergent features. In particular, LLMs such as the Generative Pre-trained Transformer 3 (GPT-3), have gained attention due to their remarkable ability to acquire a broad spectrum of knowledge during pre-training and apply it to downstream NLP tasks (Ding et al., 2023), with significant potential for chatbots (Chiang et al., 2023; Conover et al., 2023; Falcon LLM, 2023; OpenAI Platform, 2023; Taori et al., 2023; Touvron et al., 2023; Wang, 2024).

2.3. Leveraging LLMs for cybersecurity NLP

Employing the capabilities of LLMs in ML tasks, specifically within the scope of cybersecurity, offers several possibilities (Zhang, Zhang, Ren, Li, & Yang, 2023). Two problems that stand out are text classification, which categorizes text according to its relevance to a cybersecurity context, and NER, which recognizes specific cybersecurity entities in the text, for example, if it describes a vulnerability, which products are affected. LLMs have shown outstanding effectiveness in many NLP tasks, including binary classification and NER (Kocoń et al., 2023; Min et al., 2023).

2.4. OSINT for cybersecurity

In the CTI field, NLP tasks such as text classification and NER, which involve exploring various cybersecurity threats and these concepts are crucial. In a previous study, SYNAPSE (Alves et al., 2021), an OSINT processing pipeline was implemented to efficiently identify and concisely present various cybersecurity incidents to security analysts. SYNAPSE was motivated by the results of previous works, e.g., Alves et al. (2020) and Sabottke et al. (2015), which analyzed the completeness and timeliness of cybersecurity-related OSINT on Twitter.

SYNAPSE employed Support Vector Machines (SVM) (Cortes & Vapnik, 1995) for binary classification and a novel stream clustering approach to aggregate related tweets. To improve SYNAPSE, Dionísio et al. (2019) designed a new tool that employs a CNN to detect cybersecurity-related texts gathered from Twitter and a BiLSTM network for performing NER on the detected tweets to identify the type of threat, affected products, and other information. That work was further extended using a multitask Deep Learning (DL) approach (Dionisio, Alves, Ferreira, & Bessani, 2020) that could simultaneously perform classification and NER.

Although there are many ML techniques for binary classification and NER (Jehangir et al., 2023; Li et al., 2022), to the best of our knowledge, the architecture of Dionísio et al. (2019) and Dionisio et al. (2020) is recognized (Altalhi & Gutub, 2021) as the state-of-the-art for OSINT-based CTI extraction on Twitter data, reaching almost 95% for different quality metrics. This work also shows superior performance when compared with classifiers based on SVMs and Multi-Layer Perceptrons (MLPs) and several alternative NER approaches. Due to this, we consider the architecture proposed by Dionisio et al. (2020) as the reference specialized model in our comparative analysis.

2.5. Chatbots for cybersecurity

Although traditional and DL techniques have paved the way for advancements in cybersecurity text classification and NER, a new wave of NLP tools offers even more refined capabilities. Among these tools, LLMs stand out for their exceptional capabilities in interpreting and generating human-like text.

The specific case of ChatGPT raised discussions and reviews on its potential applications in the cybersecurity field. Al-Hawawreh, Aljuhani, and Jararweh discusses implications and presents practical applications of ChatGPT in the cybersecurity domain, considering both attacks and defenses. A use case of false data injection attack to an industrial control system is given (Al-Hawawreh et al., 2023). Okey, Udo, Rosa, Rodríguez, and Kleinschmidt uses topic modeling and sentiment analysis to investigate the opinions of users regarding security issues with ChatGPT and its use to assist cybersecurity professionals (Okey et al., 2023). The use of ChatGPT for analyzing program language code throughout the software development life cycle to uncover vulnerabilities, bugs, or security risks (Noever & Williams, 2023) has been demonstrated. Beyond finding security holes and vulnerabilities in submitted code, ChatGPT can also write code to exploit those weaknesses. ChatGPT has the potential to function as a honeypot, allowing attackers to engage with its interface as if it were a simulated honeypot (McKee & Noever, 2023). A flexible environment can be developed by mimicking the terminal commands of Linux, Mac, and Windows operating systems, and creating an interface for tools such as TeamViewer, nmap, and ping. This environment can respond to cyber attackers' actions, thus providing valuable insights into their tactics, techniques, and procedures. Other works have designed chatbots specifically to help with cybersecurity analysis. SecBot (Franco et al., 2020) is a cybersecurity-driven conversational chatbot that extracts information from a conversation to support cybersecurity planning and management. It was developed and evaluated on a small dataset utilizing the Rasa framework (Rasa, 2024), achieving 100% accuracy in extracting the intent of attacks and associated named entities.

2.6. Evaluation of LLM-based chatbots

Given the widespread attention LLM-based chatbots are attracting, several works tried to evaluate the strengths and limitations of the technology (Akyash & M Kamali, 2024). An analysis of ChatGPT's performance (Kocoń et al., 2023; Sun et al., 2023) showed that despite its achievements, it frequently fell behind supervised baselines in various NLP tasks. Several factors influence this, including limitations on the number of tokens, a misalignment with specific NLP tasks due to its generative nature, and challenges inherent to LLMs, such as hallucination, which involves making false positive predictions. In their effort to optimize ChatGPT, the authors have proposed solutions, including multiple prompts, task-specific fine-tuning, and strategies to counter hallucination. These methods were thoroughly tested across 21 datasets, covering ten critical NLP tasks, including NER. The testing of the solutions resulted in significant performance improvements for ChatGPT, with instances where it outperformed state-of-the-art models in existing benchmarks (Sun et al., 2023). The study by Megahed, Chen, Ferris, Knoth, and Jones-Farmer (2023) shows that ChatGPT excels in structured tasks like code translation and explaining established concepts. However, it faces challenges when handling nuanced tasks such as recognizing unfamiliar entities and generating code from scratch.

Recent research findings show that the ChatGPT may encounter challenges and limitations in accurately identifying entities, including locations, names, and organizations. Qin et al. (2023) present the results of their experiments on the performance of GPT-3.5, ChatGPT, and fine-tuned models on the multi-domain CONLL dataset for recognizing entities. According to the findings reported in this paper, ChatGPT and GPT-3.5 achieved F_1 score of 53.7% and 53.5%, respectively. Sun et al. (2023) investigated the factors contributing to the sub-optimal

performance of GPT-based chatbots in NLP tasks, such as NER. They have identified several underlying causes and have proposed a set of generalized modules to mitigate these challenges in different NLP tasks. Kocoń et al. (2023) examined the capabilities of ChatGPT on a diverse set of subjective analytical tasks and objective reasoning tasks, revealing that when compared to state-of-the-art models, the average quality loss is about 25% in zero-shot and few-shot settings.

As LLM-based chatbots have become widespread, concerns about their own vulnerabilities and their role as tools for cyber-attacks have increased (Qammar et al., 2023). The usefulness of ChatGPT extends the time-to-conquer or delay attacker timelines, making it valuable for organizations seeking to enhance their cybersecurity posture (McKee & Noever, 2023). Additionally, the performances of the ChatGPT and GPT-3 chatbots are evaluated for vulnerability detection in code (Cheshkov, Zadorozhny, & Levichev, 2023). Based on a real-world dataset, this evaluation focuses on binary and multi-label classification tasks related to *common weakness enumeration* of vulnerabilities. The findings indicate that ChatGPT does not outperform the baseline classifier in classification tasks for code vulnerability detection (Cheshkov et al., 2023). More precisely, the performance of both GPT models was assessed by accuracy, precision, recall, F_1 score, and *area under the curve*. The highest F_1 score of 0.67 was achieved using the text-davinci-003 model for binary classification. In contrast, the F_1 score of all ChatGPT models remained below 0.53 for multilabel classification.

Sentiment analysis plays a crucial role in extracting user opinions and emotions from textual data to assess threats. Recent research has been directed towards developing sustainable strategies to diminish threats, vulnerabilities, and data manipulation within chatbots, ultimately improving the scope of cybersecurity. To achieve this objective, researchers created an interactive chatbot using the Bot Libre platform[2] and placed it on social media platforms such as Twitter for the specific purpose of cybersecurity (Arora, Arora et al., 2023). This study employs a sentiment analysis strategy by deploying chatbots on Twitter and subsequently analyzing Twitter data to anticipate forthcoming threats and cyberattacks.

2.7. The research gap

The reviewed papers in this section primarily focused on evaluating and testing the commercial ChatGPT chatbot, with limited attention given to assessing open-source LLM-based chatbots in the context of cybersecurity applications. Moreover, to the best of our knowledge, there is a notable absence of comprehensive comparative studies explicitly dedicated to open-source LLM-based chatbots within the specialized field of OSINT-based CTI. This gap in existing research underscores the need to examine such chatbots, their effectiveness, and their potential contributions to enhancing cyber threat awareness and detection. Although there are specialized state-of-the-art models, including DL models that excel at CTI binary classification and NER tasks (Alves et al., 2021; Dionísio et al., 2019; Dionisio et al., 2020), no comparative study has been conducted to determine whether LLM-based chatbots can compete with their performance. If they can provide competitive results, CTI tools could be changed to integrate them into the OSINT processing pipeline, decreasing the tools' complexity and maintenance costs. By employing general-purpose LLM-based chatbots, tasks related to CTI data collection, curation, labelling, and model training and updating would no longer be as necessary as they are for specialized models.

Our study aims to fill this gap by carefully comparing LLM-based open-source and publicly available commercial chatbots in the context of an OSINT-based CTI application, considering two downstream NLP tasks: binary classification and NER.

3. LLM-based chatbots

LLM-based chatbots simulate human-like conversations with users through text or speech interaction. They offer users more intelligent and contextually relevant responses to queries by utilizing LLMs' language comprehension and generation capabilities. This section reviews the eight state-of-the-art commercial and open-source LLM-based chatbots available at the time of writing, namely, LLaMA, GPT4all, Dolly 2.0, Stanford Alpaca, Alpaca-LoRA, Vicunna, Falcon, and ChatGPT. We used these chatbots in our experimental results (Section 6) to assess their effectiveness in detecting cybersecurity-related texts and identifying relevant entities.

LLaMA. LLaMA (Touvron et al., 2023) is a compilation of 7 to 65 Billion (B) parameter-based foundation language models. These were trained on trillions of tokens, demonstrating the possibility of training cutting-edge models using only publicly accessible datasets. Their training method is comparable to that described in prior research (Brown et al., 2020) and is influenced by the Chinchilla scaling laws (Hoffmann et al., 2022). Specifically, LLaMA-13B demonstrated superior performance compared to GPT-3 (175B) across a wide range of benchmarks, whereas LLaMA-65B exhibited similar performance levels to leading models, such as Chinchilla-70B and PaLM-540B. LLaMA models undergo training on substantial textual datasets by employing a conventional optimizer and large-scale transformers (Touvron et al., 2023).

Vicunna. An open-source chatbot, Vicuna-13B (Chiang et al., 2023), was developed by fine-tuning LLaMA with user-shared conversations gathered from the 70K ShareGPT. In Vicuna, gradient checkpointing (Chen, Xu, Zhang, & Guestrin, 2016) and flash attention (Dao, Fu, Ermon, Rudra, & Ré, 2022) alleviate the memory demand. The Vicuna report states that like other large language models, it has limitations. For example, it is not adept at tasks requiring logic or mathematics, and may have limitations in correctly identifying itself or assuring the factual performance of its outputs (Chiang et al., 2023).

GPT4all. This chatbot (Anand et al., 2023) utilizes LLaMA, which operates under a non-commercial license. The data for the assistant come from OpenAI's GPT-3.5-turbo, which has restrictions that prevent the development of models that directly compete with OpenAI in commercial applications. GPT4all underwent several iterations with different versions featuring different parameter sizes. While preparing this article, the initial version had 7B parameters; however, the latest iteration used 13B.

Dolly. The Dolly model (Conover et al., 2023) operated by slightly modifying an open-source model with 6B parameters sourced from EleutherAI (EleutherAI, 2023). These modifications enabled Dolly to possess instruction-following capabilities, such as brainstorming and text generation, which were not initially present in the base model. These modifications are implemented using the data from Alpaca (Taori et al., 2023). Subsequently, Dolly-v2-7b emerged as a highly advanced 6.9B parameter causal language model derived from EleutherAI's Pythia-6.9b. Although Dolly-v2-7b may not be considered a state-of-the-art model, it exhibits instruction-following capabilities of remarkably high quality, which are not typically associated with its foundational model.

The most recent version available is Dolly-v2-12b (Databricks, 2023), a model with 12B parameters developed based on EleutherAI's Pythia-12b. It has been finely tuned using a dataset called databricks-dolly-15k, which consists of an instruction corpus created by employees of Databricks.[3]

Stanford Alpaca. Stanford Alpaca (Touvron et al., 2023) is an instruction-following language model that is fine-tuned from Meta's

[2] https://www.botlibre.com/.

[3] https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

LLaMA 7B model. It was instructed using 52k self-instruct-style demonstrations (Wang et al., 2023). Alpaca displays several shortcomings in language models, such as hallucination, toxicity, and stereotypes. Specifically, hallucination appear to be a recurring issue in Alpaca (Taori et al., 2023).

Alpaca-LoRA. This model (Wang, 2024) reproduces the Stanford Alpaca results using low-rank adaptation (LoRA) (Hu et al., 2021). LoRA fine-tuning is a strategy for reducing memory requirements by employing a limited set of trainable parameters, known as adapters, rather than updating all model parameters, which remain constant.

ChatGPT. OpenAI's ChatGPT (OpenAI Platform, 2023) has generated substantial interest and sparked extensive discussion within the NLP community as well as in various other domains. The lack of clarity regarding the training process and architectural specifics of ChatGPT poses a significant obstacle to both research endeavors and the advancement of open-source innovation within this domain. Moreover, the distinction between the ChatGPT API and the web version should be acknowledged. Recent research shows considerable variability in the performance and behavior of GPT-3.5 and GPT-4 chatbots over time (Chen, Zaharia, & Zou, 2023). Another feature of generative AI models such as ChatGPT is that they do not produce repetitive responses to specific prompts (Megahed et al., 2023).

Falcon. The Falcon family (Falcon LLM, 2023) consists of two primary models: Falcon-40B and its smaller counterpart, Falcon-7B. The Falcon-7B and Falcon-40B models underwent training on a corpus of 1.5 trillion and 1 trillion tokens, respectively (Falcon LLM, 2023). A feature of Falcon models is their utilization of multiquery attention (Shazeer, 2019). In the vanilla multihead attention scheme, each head is associated with a query, key, and value. However, in the multiquery approach, a single key and value are shared across all heads.

4. Chatbots evaluation

We divided this section into three topics relevant to the evaluation: dataset, experimental methodology, and experimental results evaluation criteria. First, we introduce the Twitter dataset, the foundational source for generating subsequent prompts. The methodology subsection outlines the techniques and strategies used to assess and compare chatbot[4] performances. Finally, in the evaluation criteria subsection, we explore the metrics and standards by which the chatbots are assessed.

4.1. Dataset

We leveraged a comprehensive Twitter dataset made available by Alves et al. (2020) in their work to retrieve IoCs from Twitter OSINT. This dataset contains a combined total of 31 281 tweets collected during two distinct periods: one from November 21, 2016, to March 27, 2017, and the other from June 1, 2018, to September 1, 2018. After collection, the tweets were filtered using specific keywords and manually labelled as positive or negative, considering their relevance to the cybersecurity of an IT infrastructure, thus creating labelled datasets suitable for supervised learning. Later, Dionísio et al. (2019) labeled the dataset for NER and republished it.

The dataset consists of 31 281 tweet records, including the timestamp of the tweet, specific keywords found in the tweet, the original tweet, a pre-processed tweet cleaned from some special characters, a binary label marking the tweet as relevant for cybersecurity or not, and a string identifying the named entities in the pre-processed tweet. Table B.1 in Appendix B shows representative examples of dataset records.

In this work, we used the pre-processed tweets, the cybersecurity relevance, and the sequences of named entity tags, to create a customized dataset for the binary classification and NER experiments through which the LLM-based chatbots will be evaluated.

[4] From now on, we use the term *chatbot* to refer to LLM-based chatbots.

4.2. Experimental methodology

As discussed in previous work (Alves et al., 2021; Dionísio et al., 2019; Dionisio et al., 2020), the extraction of actionable CTI from OSINT entails three foundational tasks: text classification, extraction of pertinent information from textual sources, and the synthesis of the information gathered into concise summaries. Considering the role of summarization to aggregate correlated threat indicators prior to dissemination, this study directs its attention towards evaluating and comparing chatbot performance within the two first tasks. These NLP tasks, being more amenable to LLM methodologies (Min et al., 2023), are deemed as primary focal points for research within the scope of this study.

From a practical point of view, we interacted with the chatbots using Python language scripts to iterate through the dataset entries. For each entry, different prompts were used to address the two tasks of interest: deciding on the relevance of a tweet for cybersecurity using a binary classification formulation and extracting relevant information from the tweets using NER.

Binary classification. We designed prompts to determine whether tweets are related to the cybersecurity field, aiming to generate a response including 'yes' or 'no'. We focused on constraining the chatbot to provide concise yes or no answers without additional explanations. This restriction simplifies the process of extracting binary labels (0 or 1) from the responses. For instance, in a scenario where the "pre-processed tweet" column within our dataset contained the text "cyber infosec kenno media SQL injection", to compose the specific prompts we desired, we added the phrase *"Is the sentence"* at the beginning of each pre-processed tweet. Then, we appended the phrase *"related to cybersecurity? Only respond with yes or no."* at the end. Consequently, the final prompt becomes: *Is the sentence 'threatmeter dos microsoft internet explorer 9 mshtml cdisp node::insert sibling node use-after-free ms13-0' related to cybersecurity? Just answer yes or no.*

This process was applied to all the tweets in the "pre-processed tweet" column to create a question dataset. We administered three distinct tests, each consisting of the 31 281 questions.

- Test 1- Normal Dataset: In this test, we assess the performance of chatbots using the Twitter dataset described before, keeping the order of the rows unchanged. The objective is to assess how well chatbots can accurately identify tweets related to cybersecurity.
- Test 2- Shuffled dataset: To examine the impact of question order on the resulting answers, the question dataset was shuffled and provided as input to the chatbots.
- Test 3- Isolated Prompt Testing: In this test, we indicate that each question was regarded as an isolated prompt without considering the context of past interactions. This test aims to clear the chatbot's history and context between consecutive questions during testing. It allows for clean and unbiased interaction for each question. For this test, we employ a methodology that closes the chatbot session after each question and initializes a fresh session for the subsequent question. This ensures the chatbot does not retain any knowledge or bias from previous interactions, thus providing a fair evaluation of its performance on individual questions.

We conducted research on two versions of commercial ChatGPT, namely GPT-3.5-turbo and GPT-4, that were available on August 5th, 2023. The API provided by OpenAI was used to send requests to ChatGPT for each tweet in the dataset. Various parameters can be configured by utilizing the ChatGPT API. One particularly influential parameter is referred to as "temperature". This parameter governs the level of creativity or randomness of the generated text. A higher temperature setting (e.g., 0.7) produces a more varied and imaginative output, whereas a lower setting (e.g., 0.2) yields a more predictable and concentrated output. We adjusted the temperature parameter to 0.2 to

reduce output randomness while leaving the other parameters at their default values.

Named entity recognition. The assessment of performance in NER involves two distinct methodologies. The initial approach, which is called Entity-Specific Prompting (ESP), was chosen based on a careful evaluation of the various experimental results. It was evident from these experiments that attempting to extract all required entities from a single tweet using a single question yielded significant misrecognition, rendering the approach impractical. Consequently, we decided on a more precise strategy involving the creation of specific prompts for each entity within each tweet. After sending several prompt requests for each entity to the chatbots, we found that the organization names and product version entities were the most extractable.

The revised ESP approach focuses on two specific entities: organization names (B-ORG) and product versions (B-VER). We selected these entities for extraction by interacting with ChatGPT-3.5-turbo, ChatGPT-4, GPT4all, and Dolly chatbots. The analysis for ChatGPT is based on the version released on July 13th, 2023. Since only 11 074 out of 31 281 questions in the dataset had been tagged with NER labels, we limited our analysis for the NER task to these 11 074 tweets. In our experiment, we employed various prompts to choose the proper one; ultimately, we selected the following prompts to identify the organization name and product version:

- *Find the name of organizations in the following sentence: 'threatmeter dos microsoft internet explorer 9 mshtml cdisp node::insert sibling node use-after-free ms13-0'. Give the shortest answer, and only use sentence segments in your response.*
- *Find only product version numbers without any product, vulnerability, and company names in the following sentence: 'threatmeter dos microsoft internet explorer 9 mshtml cdisp node::insert sibling node use-after-free ms13-0'. Give the shortest answer, and only use sentence segments in your response.*

The second approach, which is called Guide-Line Prompting (GLP), exclusively employed for ChatGPT-4, involves a comprehensive specification of all entities within the 'GUIDELINES_PROMPT' section. This approach attempts to extract seven entities in a single prompt and considers only ChatGPT because the GLP feature is absent in open-source chatbots. In this guideline, we include two examples of tweets from 11 074 NER-tagged tweets in the dataset, each annotated with their respective entities. In addition, we have included the output format as a means to direct ChatGPT's response for subsequent processing. We then sent a dedicated prompt for each of the 11 074 tweets and systematically covered all the extracted entities. The prompt guideline is given in Appendix A. The GLP experiments were based on the ChatGPT released on August 2, 2023.

4.3. Evaluation criteria

This section describes our evaluation criteria for assessing the LLM-based chatbots' performance in addressing binary classification and NER tasks. Our evaluation methodology focuses on performance and quality.

Performance. To assess the considered chatbots' performance in binary classification, we used the F_1 score, which is a metric that computes the harmonic mean of Precision and Recall (Lipton, Elkan, & Naryanaswamy, 2014). It strikes a balance between the proportion of true positive results in all positive predictions and the proportion of true positive results in all actual positives. The formula of this criterion is detailed in Appendix C.

Quality. For the classification task, the response quality relates to providing precise yes or no answers to the prompts. The assessment is based on three distinct response modes: no response, correct response, and implicit response.

- No response: Instances in which the LLM-based chatbots failed to respond.
- Precise response: The questions to which answers were precisely aligned with the desired yes or no response.
- Implicit response: Answers in which the response did not explicitly mention yes or no. However, careful inference from the generated answers revealed that the intended response was either yes or no.

By analyzing these distinct response modes from the output files, we gain valuable insights into the performance and capabilities of the evaluated chatbots to effectively detect and address cybersecurity-related questions. The experimental results section provides a detailed explanation of the experimental findings, focusing on the quality and accuracy evaluation criteria.

5. Optimal chatbot utilization strategies

The fundamental principle behind creating prompts is ensuring clear and precise instructions. To have prompts with these features, we explain how to reach the optimal prompts for the considered chatbots in the following sections.

5.1. Prompt engineering approaches

Prompt engineering (Liu et al., 2023) is crucial for optimizing the performance of LLM-based chatbots by enhancing the clarity and specificity of the given instructions. Thorough prompt design and testing improve the ability of the chatbot to comprehend requests, making it a more effective tool for generating desired outcomes. These approaches enhance the possibility of directing the chatbot to the desired output while decreasing the chances of receiving irrelevant or incorrect responses.

We devoted considerable attention to formulating suitable prompts to maximize accurate and relevant responses during the experiments. The objective was to design prompts that effectively allow chatbots to capture the essence of the CTI task in the information present in tweets.

Following the best practices for prompt engineering (Sahoo et al., 2024), we progressively refined our prompts using two approaches. For NER, we considered first a prompt guideline template approach but with unsatisfactory results (prompt template is shown in Appendix A). We used a zero-shot methodology for binary classification, considering that LLMs are proficient zero-shot reasoners (Kojima, Gu, Reid, Matsuo, & Iwasawa, 2022). Since this approach produced very good results, for the NER tests, we transformed the prompt guideline into a sequence of zero-shot questions, one for each entity we aim to extract.

To leverage the zero-shot capability of LLMs, we explored the assumption that chatbots perfectly model the cybersecurity concept. Following this assumption, we designed prompts by starting the question with information on the cybersecurity issue and finishing it with precise instructions on the expected outcome. The process of selecting the final prompt was iterative and relied on a few trial and error cycles, using the principles described before. Using a sample of dataset entries, we progressively queried the chatbots and evaluated the answers until consistent answers were achieved concerning the instructions given. This refinement cycle resulted in the determination of the final prompt. By formulating questions such as *Is the sentence 'vuln oracle java se cve-2016-5582 remote security vulnerability' related to cybersecurity? Just answer yes or no.*, we aimed to elicit yes or no responses. Using spaces and adequately employing the apostrophe (') played a significant role in clarifying the prompt.

This methodology allowed us to assess the capabilities of the LLM-based chatbots to detect cybersecurity concepts and their ability to generate meaningful and contextually appropriate responses. Having formulated the desired prompt, we automated the process of sending the entire set of questions to the chatbots.

5.2. Text length control

LLM-based chatbots can produce text of varying lengths based on specific tasks. In our experiments, minimizing the number of answer tokens was essential because of the high volume of questions and the time required for the chatbot to answer each question. The parameter that defines the length of the answer or the number of tokens in locally executed chatbots is represented as *N_predict*. It plays a critical role in significantly reducing the execution time for answering questions across all open-source LLM-based chatbots. To optimize the execution time, we advise setting a parameter that controls the response length to the smallest suitable value based on a specific task. After some initial experimentation, we consistently set the value of *N_predict* to 15 across all open-source chatbots.

In the context of ChatGPT-3.5-turbo and ChatGPT-4, the *max_tokens* parameter constrains the length of the responses generated by the chatbot. This is achieved by establishing a predetermined upper limit for the number of tokens that can be words or characters within the generated output. Using more extended responses in ChatGPT-4 can increase token consumption, potentially increasing the usage costs. After some initial experimentation with prompts with extreme lengths, the value 70 was assigned to *max_tokens* in the experimental tests. It is worth noting that the chosen token length of 15 for the open-source LLM-based chatbots applies exclusively to the generated answers. By contrast, the selected token length of 70 for ChatGPT chatbots encompasses questions and answers.

The careful utilization of the *N_predict* and *max_tokens* parameters is of utmost importance, as a low setting may lead to truncation of the response, potentially producing incomplete or nonsensical answers. Balancing the desired response length with the need for completeness and coherence is a crucial factor to consider.

6. Experimental results

In this section, we present the results of the empirical assessment of LLM-based chatbots thorough evaluations of their capabilities across multiple dimensions. First, we discuss the evaluation of binary classification, focusing on how chatbots classify user inputs proficiently. Next, we present the evaluation of NER tasks by examining the effectiveness of chatbots in identifying and classifying entities present within user inputs. The collective findings from these experiments offer comprehensive insights into LLM-based chatbots operational strengths and potential areas for improvement.

It is essential to highlight that we ultimately elaborate on the common and default parameters shared by all open-source LLM-based chatbots. In most of these, the maximum size of the context window[5] and its default value are set to 512. An exception is Dolly, which has a maximum context window size of 1024. However, ChatGPT exhibits variability across its final versions, each with a distinct context window size. For instance, ChatGPT-4 is available in two versions with window sizes of 8k and 32k, whereas ChatGPT-3.5-turbo is available in 4k and 16k versions. In our experiments, we utilized a server equipped with multiple GPU units, including an NVIDIA A30 GPU (memory capacity: 24,576 MiB) and an NVIDIA RTX A6000 GPU (memory capacity: 49,140 MiB), with 264 GB of RAM.

6.1. Binary classification

We present the results obtained by the LLM-based chatbots, focusing on two critical dimensions: quality and performance.

Quality and performance. In terms of quality, different categories are shown in different colors. The presence of red in Fig. 1 highlights a noteworthy observation regarding the unanswered questions. Precise responses are visually represented by the green bars on the chart, effectively indicating the successful accomplishment of the intended goal. The orange bars in the diagram represent acceptable answers with minor imperfections, showing the capacity to offer responses that were contextually in line with the desired result, albeit with slight deviations. Vicuna encountered significant challenges in answering 17 420 questions despite our efforts to use iteration loops to generate answers. To address these unanswered questions, we included them in the loop until the model could respond. However, a considerable number of questions have become trapped in an infinite loop and remain unanswered. This problem is unique to Vicuna, with no other chatbots showing it. Fig. 1 demonstrates that ChatGPT and Stanford Alpaca produced consistent responses, with all 31 281 questions beginning with either "yes" or "no". By contrast, GPT4all achieved a slightly lower rate, with 98.80% of its responses starting with either yes or no. For Dolly and Falcon, the rates are 97.32% and 94.10%, respectively. GPT4all delivered conceptual yes or no responses to 378 questions, accounting for 1.2%, whereas Dolly produced responses for 839 questions, constituting 2.68%. Interestingly, however, neither explicitly used the words yes or no in these conceptual answers. For example, the conceptual answers were: *it is related to cybersecurity*, or *the sentence is not related to cybersecurity*.

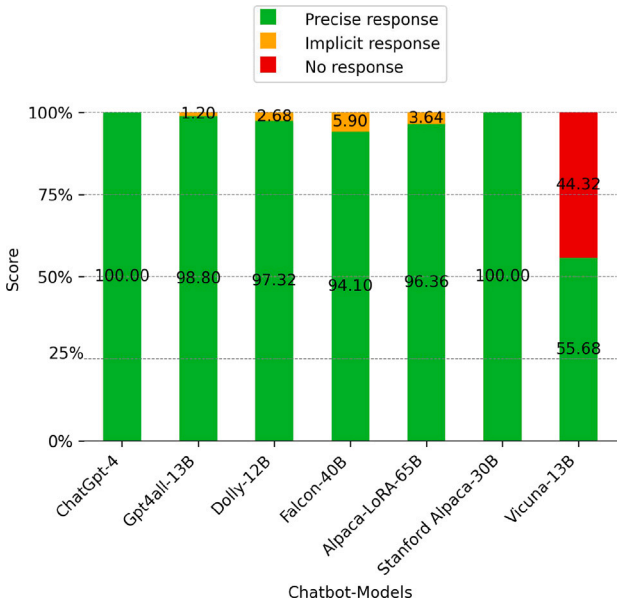

Fig. 1. Comparison of LLM-based chatbots response modes—Test 1.

In terms of performance, we explain the results of the tests conducted on LLM-based chatbots. Table 1 provides an overview of the tests conducted for each chatbot and its version. It includes details regarding the number of parameters of the chatbot, achieved F_1 score, precision, recall, and execution time. The F_1 score, when evaluated on the selected dataset, provides a measure of how accurately the chatbot predicts labels compared to the true labels. A high F_1 score indicates that the chatbot correctly predicts the positive and negative labels, while a low score suggests it is struggling to accurately classify the data. The confusion matrices corresponding to each chatbot are provided in Appendix C. By analyzing the F_1 score values in Test 1, we can assess the chatbot's effectiveness in accurately responding to the given questions. Based on the results presented in Table 1, it is evident that the GPT4all achieved the highest accuracy among the open-source variants, as indicated by its F_1 score of 0.90. Dolly achieves an accuracy of 0.86, Falcon 0.85, Alpaca-LoRA 0.84, and Stanford Alpaca a score of 0.64. Although GPT4all achieved higher accuracy among open-source

[5] Upper limit for the range of tokens considered to provide an answer.

Table 1
Accuracy of LLM-based chatbots for cybersecurity binary classification.

Model	Test number	Parameters	Precision	Recall	F_1 score	Execution time
ChatGPT-3.5-turbo (16k context)	Test 1	175B	0.9570	0.9280	0.9431	11 h 23 m
ChatGPT-3.5-turbo (16k context)	Test 2	175B	**0.9700**	0.9200	**0.9489**	11 h 23 m
ChatGPT-3.5-turbo (16k context)	Test 3	175B	–	–	UECH	–
ChatGPT-4 (8k context)	Test 1	1.7T	0.9580	0.9240	0.9410	11 h 50 m
ChatGPT-4 (8k context)	Test 2	1.7T	0.9590	0.9230	0.9403	11 h 43 m
ChatGPT-4 (8k context)	Test 3	1.7T	–	–	UECH	–
GPT4all	Test 1	13B	0.9490	0.8630	0.9049	132 h 05 m
GPT4all	Test 2	13B	0.9490	0.8410	0.8927	132 h 02 m
GPT4all	Test 3	13B	0.9470	0.8280	0.8844	136 h 05 m
Dolly 2.0	Test 1	7B	0.8890	0.8000	0.8470	10 h 38 m
Dolly 2.0	Test 1	12B	0.9470	0.7900	0.86120	10 h 16 m
Dolly 2.0	Test 2	12B	0.9480	0.7910	0.8631	10 h 00 m
Dolly 2.0	Test 3	12B	–	–	–	LET
Falcon	Test 1	7B	0.8120	0.8500	0.8304	16 h 02 m
Falcon	Test 1	40B	0.8980	0.8200	0.8511	54 h 03 m
Falcon	Test 2	40B	0.8990	0.8080	0.8502	54 h 55 m
Falcon	Test 3	40B	0.8990	0.7880	0.8330	71 h 10 m
Alpaca-LoRA	Test 1	65B	0.8980	0.7940	0.8477	10 h 12 m
Alpaca-LoRA	Test 2	65B	0.8990	0.8000	0.8451	10 h 44 m
Alpaca-LoRA	Test 3	65B	0.8980	0.7610	0.8241	11 h 20 m
Stanford Alpaca	Test 1	7B	0.2260	0.5000	0.3112	13 h 03 m
Stanford Alpaca	Test 1	13B	0.3240	0.6000	0.4209	13 h 21 m
Stanford Alpaca	Test 1	30B	0.6980	0.6050	0.6415	15 h 48 m
Stanford Alpaca	Test 2	30B	0.6990	0.5920	0.6401	15 h 04 m
Stanford Alpaca	Test 3	30B	0.6990	0.5810	0.6395	16 h 18 m
Vicuna	Test 1	13B	0.4390	0.3100	0.3611	11 h 23 m
Dionisio et al. (2020)	Test 1	–	0.9570	**0.9363**	0.9470	00 h 43 m

* LET: Long Execution Time. * UECH : Uncertainty of Erasing Conversation History.

Table 2
Comparison of NER task accuracy achieved using two different approaches for 11 074 questions.

Chatbot	Approach	Entity	F_1 score	Execution time
ChatGPT-4	ESP	Organization	0.36	4 h 02 m
ChatGPT-4	ESP	Version	0.43	4 h 23 m
GPT-3.5-turbo	ESP	Organization	0.07	5 h 10 m
GPT-3.5-turbo	ESP	Version	0.03	4 h 54 m
Dolly	ESP	Organization	0.01	4 h 06 m
Dolly	ESP	Version	0.009	4 h 21 m
GPT4all	ESP	Organization	0.004	21 h 03 m
GPT4all	ESP	Version	0	21 h 14 m
ChatGPT-4	GLP	All entities	0.10	3 h 09 m

variants, it is noteworthy that the commercial ChatGPT (GPT-4 and GPT-3.5-turbo) achieved an F_1 score of 0.94. ChatGPT-3.5-turbo with a 16k window context size achieves the same F_1 score as ChatGPT-4 with an 8k context window size. These results highlight the better accuracy of GPT4all and ChatGPT, emphasizing their effectiveness for this particular task.

Based on the shuffled dataset test, GPT4all achieved an F_1 score of 0.89%, whereas Dolly attained an F_1 score of 0.86%. This indicates a slight decrease in accuracy of approximately 1% for GPT4all compared with the first test, which can be considered insignificant. It is essential to mention that shuffling the prompt does not affect the accuracy of ChatGPT, Falcon, Stanford Alpaca, and Alpaca-LoRA, since the F_1 score is equal to that of the first test.

Upon applying the isolated prompt test to the dataset, the F_1 score for GPT4all attained a value of 0.88. This test results in a 2% decrease compared with that of Test 1. Conducting the test for Stanford Alpaca did not result in a significant reduction in accuracy. When testing Alpaca-LoRA, we observed a two percent reduction in the F_1 score of 0.82. The result of this test (0.83) on Falcon 40B is two percent less than the first test. We ignored this test on Dolly because it was time-consuming, beyond the available time and resources. For ChatGPT, we used the available API to send requests. In this case, Test 3 is not feasible because isolated prompt functionality is not available, i.e., the conversation history of the chatbot model cannot be completely reset, and we have no means to reinitialize the chatbot for each prompt. It is worth noting that running this test increases the execution time because of the need to launch the chatbot for each question. Nevertheless, the remaining results indicate that we should not expect a significant decrease in accuracy.

LLM-based chatbots are based on a varying number of parameters. This study involved chatbot models ranging from 7B to 65B in parameter counts. Based on our experiment, the number of parameters significantly influences the effectiveness of the chatbots in answering questions. Generally, 7B parameters exhibited a comparatively lower performance (Bi et al., 2024) and, based on our experiments, failed to provide binary (‘yes’ or ‘no’) answers.

Remarkably, GPT4all, with 7B parameters, faced limitations in providing yes or no responses to every question. In response to each question, the output consisted solely of explanations, leading to a significant expenditure on human effort to discern whether the response was yes or no, thus preventing the automated processing of the answers. Consequently, we were unable to calculate the F_1 score for the 7B parameter chatbots. In addition, Dolly and Stanford Alpaca, both of which have precisely 7B parameters, have lower F_1 score than the chatbot models with 13B parameters, as shown in Table 1.

Execution time. In our results, the chatbot execution times varied significantly. The implementation process involves running LLMs on a GPU server, which requires several days of continuous execution. GPT4all took the longest, with a total execution time of 132 h and five minutes. Falcon followed with 54 h and 3 min. Stanford Alpaca completed its tasks in 15 h and 48 min, while Dolly had a slightly shorter execution time of 13 h and 38 min. ChatGPT-4 was more efficient, ranking third with an execution time of 11 h and 50 min. The fastest among them was Alpaca-LoRA, with an execution time of 10 h and 12 min. The chattbot models with 30B and 65B parameters exhibit longer execution times when answering questions. An impressive aspect of Alpaca-LoRA with 65B parameters is that despite its significantly

Table 3
NER identification responses are shown for two distinct prompts.

Chatbot	Entity	Chatbot responses
ChatGPT-4 (8k context)	Organization	Microsoft
ChatGPT-4 (8k context)	Version	9
GPT-3.5-turbo	Organization	The name of organizations in the given sentence is "Microsoft".
GPT-3.5-turbo	Version	9, ms13-0
Dolly 2.0	Organization	Microsoft Internet Explorer 9
Dolly 2.0	Version	ms13-0 is 9.0.8112.16421
GPT4all	Organization	Microsoft, Mozilla (Firefox), and Google Chrome
GPT4all	Version	546

larger parameter count than Dolly, which has 7B and 12B parameters, both exhibit equal execution times and F_1 score.

The last row of Table 1 is devoted to the multitask model (Dionisio et al., 2020) discussed in Section 2.4 of the related work. Since this specialized model achieves the best performance in OSINT-based CTI extraction, it was selected as the evaluation reference. Furthermore, the Twitter dataset employed for both this model and the LLM-based chatbots is identical, guaranteeing a fair evaluation under comparable conditions. This model is a Bidirectional Long Short-Term Memory (BiLSTM) trained for binary classification tasks and NER. It achieved an accuracy of 0.94, equivalent to the ChatGPT accuracy. However, the execution time is only 43 min, much lower than the 11+ h required by ChatGPT. Based on the experimental results, it is evident that ChatGPT, with a context window size of 512, outperforms the open-source LLM-based chatbots in terms of F_1 score. One key contributing factor to this accuracy gap is the significantly larger context window size employed by the ChatGPT.

6.2. Named entity recognition

Table 2 presents the NER performance metrics obtained using the ESP approach (top 8 rows) for identifying B-VER and B-ORG entities and using the GLP approach (last row) to extract all entities. ChatGPT-4 achieved an F_1 score of 0.43 and 0.36 for B-VER and B-ORG extraction, respectively. The F_1 score results for GPT-3.5-turbo, Dolly, and GPT4all are close to zero. The GLP approach result is unexpectedly low, with ChatGPT-4 reaching an F_1 score of 0.1. Collectively, these results demonstrate a significant degradation from previous NER results using a specialized DL model (Dionisio et al., 2020), which achieved an F_1 score of 0.94.

A representative example of the responses generated by each chatbot is provided in Table 3. In the two NER prompts mentioned in Section 4.2, the annotations in the dataset show 'Microsoft' as an organization entity and '9' as a product version. The phrase *'without any product, vulnerability, and company names'* in the version prompt plays a crucial role in constraining the chatbots' interpretation of version numbers. ChatGPT-4 demonstrated precision in identifying 'Microsoft' as B-ORG and '9' as B-VER, accurately matching the annotations in the dataset. By contrast, the remaining chatbots exhibited varying degrees of accuracy, failing to reach the level of correctness achieved by ChatGPT-4 in these specific instances.

7. Discussion

The evaluation of the chatbots considered involves two critical aspects. First, it includes considering timeliness, which is particularly important when integrating chatbots into real-time systems such as those connected to Twitter like SYNAPSE (Alves et al., 2021). Second, it requires the skill of writing structured and clear prompts to ensure the generation of precise and relevant responses. It is essential to compose clear, concise, and controlled-length prompts that guide the chatbot in providing an anticipated response. Long answers not only extend execution times but also impose an additional workload on human resources for response validation. Moreover, these responses may not have contained sufficient meaningful content.

Automated interaction with chatbots for our target tasks is another issue requiring specific needs. This includes the ability to generate relevant and precise prompts automatically. An automated system must also interpret and adapt to various response formats and deal with the complexities of its data input.

Our evaluation primarily focused on a specific Twitter dataset and other OSINT resources, such as security blog posts and forums (even on the dark web), which remain unexplored. Twitter offers several benefits as a source of cybersecurity information. It includes almost perfect vulnerability coverage, with most vulnerabilities reported early on having high or critical impact, timely discussion of vulnerabilities, widespread use for sharing and disseminating information, a structured stream of short and focused texts suitable for studying and developing text-based streaming information processing systems, and its use in state-of-the-art results for comparative analysis.

It is important to note that a manual review of responses following the automated checking phase may introduce potential human errors into the assessment process. Furthermore, such verification is time-consuming and thus not feasible in the day-to-day operation of a security operating center.

LLM-based chatbots present opportunities in numerous applications but face challenges that hinder their effective utilization. The following outlines several challenges and limitations encountered during our experiments for binary classification and NER tasks. A common challenge was the generation of effective prompts, which required progressive refinement until a satisfactory performance was achieved.

Binary classification Challenges. In Section 4.2, we discussed how each chatbot displayed unique response behaviors when answering questions. This necessitated a cleaning step after collecting responses to ensure answer consistency, particularly because we aimed to obtain a precise binary (yes or no) response. However, some responses implied a 'no' or a 'yes' without explicitly using these words. For instance, a chatbot might answer the question, *This is not related to cybersecurity*. Consequently, reviewing and validating the answers in the output file is crucial.

Therefore, our evaluation process involved two key steps to ensure reliability and accuracy. First, we implemented an automated validation method for each output file to confirm the presence of 'yes' or 'no' responses at the beginning of the responses. This helped to filter out potentially incorrect answers, such as those lacking an explicit 'no' or 'yes'. Second, we conducted a thorough manual review to confirm implicit responses and their alignment with the expected yes/no answers. In cases of the aforementioned example, we manually annotated the response with 'no' to correctly classify it. This comprehensive manual validation adds a crucial layer of scrutiny, bolstering the accuracy and reliability of the results. Our comprehensive validation process improves the accuracy of our binary classification. However, this creates a time burden for classification. Timeliness is key when using LLM-based chatbots for binary classification tasks, particularly in CTI applications. Real-time processing is vital because delays in classifying responses

```
GUIDELINES_PROMPT = (
            "Entity Definition:\n"
            "1. B-ORG: organization names.\n"
            "2. B-PRO:  product names.\n"
            "3. B-VER: Version number of products.\n"
            "4. B-VUL: Name of cybersecurity vulnerability or attack.\n"
            "5. O: Non-entity, no specific entity present.\n"
            "6 B-ID: Bulletin ID.\n"
            "\n"
            "Output Format:\n"
            "{{'B-ORG': [list of entities present]"
            "'B-PRO': [list of entities present]"
            "'B-VER': [list of entities present]"
            "'B-VUL': [list of entities present]"
            "'O': [list of entities present]"
            "'B-ID': [list of entities present]}}\n"
            "Examples:\n"
            "\n"
            "1. Sentence: rt zdnet windows users attacked via critical flash zero-day patch now urges adobe.\n"
            "Output: {{'O': ['rt', 'zdnet', 'zero-day', 'patch', 'now', 'urges', 'users', 'attacked', 'via', 'critical'], "
            "'B-PRO': ['windows', 'flash'], 'B-ORG': ['adobe']}}\n"
            "\n"
            "2. Sentence: {}\n"
            "Output: "
            "\n"

        )
```

Fig. A.1. GLP NER approach: employing a ChatGPT-4 guideline prompt template.

Table B.1
Samples from the 31 281 tweet entries in the dataset.

Timestamp	Keywords	Original tweet	Pre-processed tweet	Relevance	Entities
2018-07-24 01:00:46+00:00	oracle	RT Oracle: Learn to use and understand #Oracle's Internet Intelligence Map https://t.co/l06Nyf1FFF Dyn https://t.co/uzozFKwm97	rt oracle learn to use and understand oracle s internet intelligence map dyn	0	–
2016-12-09 19:19:38+00:00	internet explorer	Threatmeter: [dos] - Microsoft Internet Explorer 9 MSHTML - CDisp Node::Insert Sibling Node Use-After-Free MS13-0... https://t.co/gLvEwpDL9v	Threatmeter dos microsoft internet explorer 9 mshtml cdisp node::insert sibling node use-after-free ms13–0	1	O O B-ORG B-PRO I-PRO B-VER O O O O O B-VUL B-ID

can significantly affect decision-making. Therefore, balancing accuracy with the demand for prompt responses is critical.

NER Challenges. When employing LLM-based chatbots in NER tasks for cybersecurity purposes, we observed various limitations, mainly in providing precise and relevant results. While chatbots powered by pre-trained language models excel at understanding natural language and utilizing general knowledge, they frequently encounter challenges when dealing with domain-specific precise entity recognition (Yang et al., 2024). This shortcoming is attributed mainly to the intrinsic complexities of NER, which demand a profound understanding of context, domain knowledge, and syntactic intricacies.

A recurrent issue in NER is the generation of unspecific answers that fail to accurately identify precise entities in a given sentence. This often leads to generalized responses that lack the precision essential for obtaining reliable NER results. Consider this prompt as an example: *Find the name of organizations in the following sentence: 'senator calls on us government to start killing adobe flash now tripwire'. Give the shortest answer, and only use sentence segments in your response.* The ChatGPT-4 response was 'US Government, Adobe, Tripwire'. Such challenges can be traced back to factors such as the limitations inherent in the underlying language models or the absence of dedicated fine-tuning tailored to NER tasks. Another phenomenon, hallucination, as discussed in Section 2.5, also arises from these compounded challenges. As an example of precise hallucination, GPT4all mistakenly extracted '546' as a product version, the number that was not seen at all in the product version prompt, which is mentioned in Section 4.2.

8. Conclusion

We assess the capabilities of open-source and commercial LLM-based chatbots to recognize cybersecurity-related tweets and extract pertinent information from them. Both types of chatbots can perform similarly to specialized models trained specifically for the binary classification task of identifying cybersecurity-related tweets, often achieving the same level of performance. On the contrary, the performance of the LLM-based chatbots is still very poor on named entity recognition to extract security elements from tweets. Even when training on vast datasets, these chatbots did not perform comparably to specialized models on the test data.

Our results highlight the need for further research and refinement in the application of LLM-based chatbots to extract threat indicators from open-source intelligence. Moreover, they cannot compete with specialized models on timeliness and cost. Based on our study, we have identified several possibilities for future work. Firstly, we aim to further optimize LLM-based chatbots for cost-effective real-time CTI detection on social media platforms. Secondly, we plan to improve the NER capability for the extraction of indicators of compromise. Lastly, we will investigate how cybersecurity specialists' feedback can be used to increase the efficiency and cost-effectiveness of open-source LLM-based chatbots.

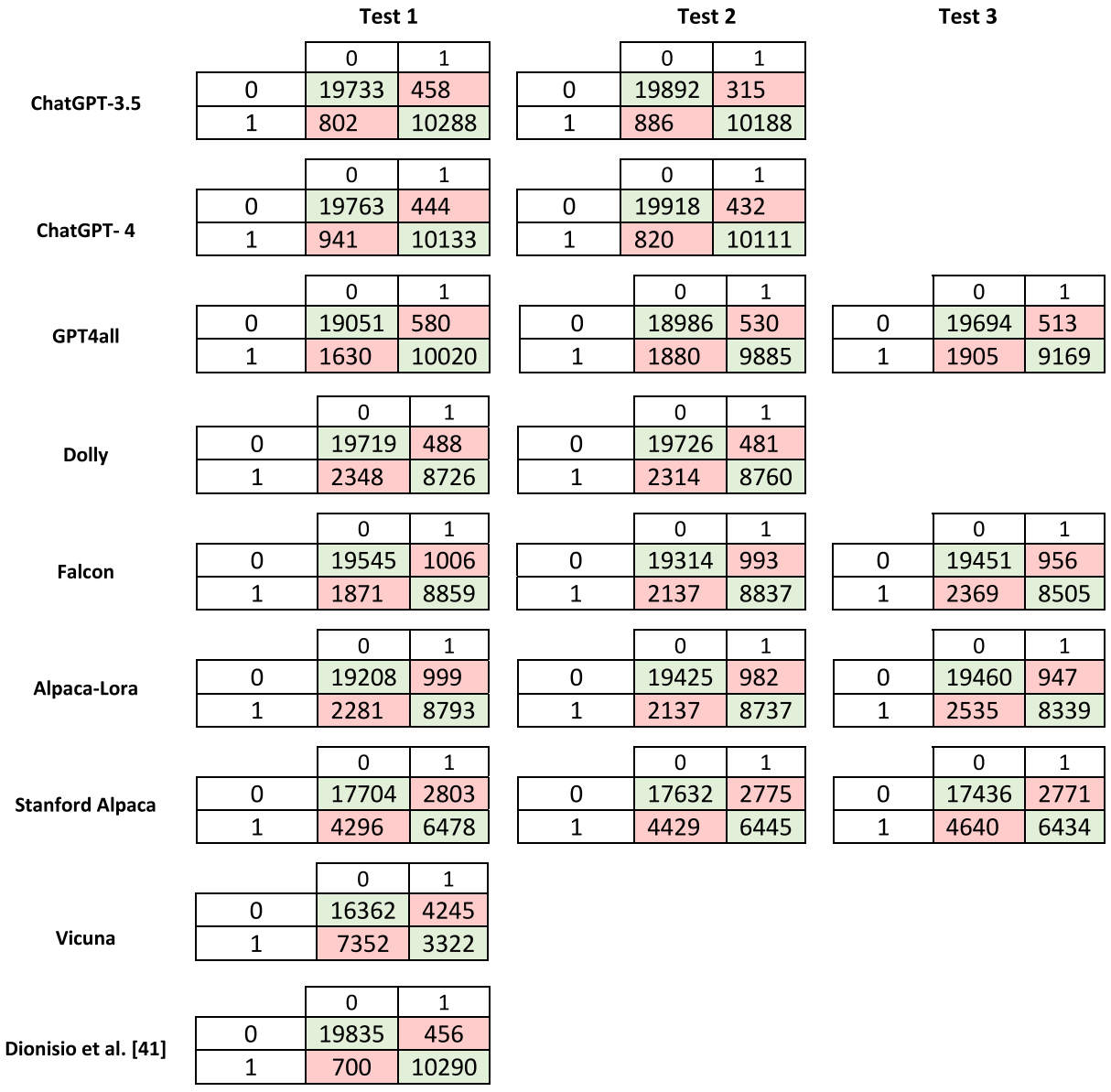

Fig. C.1. Confusion matrix of binary classification task.

CRediT authorship contribution statement

Samaneh Shafee: Methodology, Investigation, Data curation, Software, Formal analysis, Writing – original draft. **Alysson Bessani:** Conceptualization, Supervision, Writing – review & editing. **Pedro M. Ferreira:** Supervision, Methodology, Funding acquisition, Resources, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is funded by the European Commission through the SATO Project (H2020/IA/957128) and by Fundação para a Ciência e a Tecnologia (FCT) through the LASIGE Research Unit (UIDB/00408/2020 and UIDP/00408/2020).

Appendix A

The template for the guideline prompt used in the ChatGPT GLP NER approach is shown in Fig. A.1.

Appendix B

As described in Section 4.1, for each collected tweet a dataset entry is generated, including timestamp, keywords, original tweet, pre-processed tweet, cybersecurity relevance binary label, and sequence of named entities in the pre-processed tweet. Table B.1 presents two

examples of dataset entries. In the relevance column, '1' denotes an entry considered relevant for cybersecurity, a '0' means otherwise. The last column shows the tags used to label the different NER entities.

Appendix C

The confusion matrices are provided (See Fig. C.1) for the test and chatbot combinations considered, excluding the 7B parameter variants in Table 1 which achieved the worse results in the respective group. The rows in the matrices correspond to the actual expected result, whereas the columns show the predicted results. In addition to the confusion matrices, we also report the F_1 score for each combination of test and chatbot in Table 1, which provides a balanced measure of precision and recall. The F_1 score is given by

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{C.1}$$

where Precision is the ratio of correctly predicted positive observations to the total predicted positives given by

$$\text{Precision} = \frac{TP}{TP + FP},$$

and Recall measures the ratio of correctly predicted positive observations to all observations in the positive class:

$$\text{Recall} = \frac{TP}{TP + FN}.$$

References

Agrawal, S. (2023). Are LLMs the master of all trades?: Exploring domain-agnostic reasoning skills of LLMs. Preprint arXiv:2303.12810.

Akyash, M., & Kamali, H. M. (2024). Self-HWDebug: Automation of LLM self-instructing for hardware security verification. arXiv preprint arXiv:2405.12347.

Akyash, M., & M Kamali, H. (2024). Evolutionary large language models for hardware security: A comparative survey. In *Proceedings of the great lakes symposium on VLSI 2024* (pp. 496–501).

Al-Hawawreh, M., Aljuhani, A., & Jararweh, Y. (2023). Chatgpt for cybersecurity: practical applications, challenges, and future directions. *Cluster Computing*, *26*(6), 3421–3436.

Altalhi, S., & Gutub, A. (2021). A survey on predictions of cyber-attacks utilizing real-time twitter tracing recognition. *Journal of Ambient Intelligence and Humanized Computing*, 1–13.

Alves, F., Andongabo, A., Gashi, I., Ferreira, P. M., & Bessani, A. (2020). Follow the blue bird: a study on threat data published on twitter. In *European symposium on research in computer security* (pp. 217–236). Springer.

Alves, F., Bettini, A., Ferreira, P. M., & Bessani, A. (2021). Processing tweets for cybersecurity threat awareness. *Information Systems*, *95*, Article 101586.

Anand, Y., Nussbaum, Z., Duderstadt, B., Schmidt, B., & Mulyar, A. (2023). GPT4All: Training an assistant-style chatbot with large scale data distillation from GPT-3.5-turbo.

Arora, A., Arora, A., & McIntyre, J. (2023). Developing chatbots for cyber security: Assessing threats through sentiment analysis on social media. *Sustainability*, *15*(17), 13178.

Arora, D., Singh, H. G., et al. (2023). Have LLMs advanced enough? A challenging problem solving benchmark for large language models. In *The 2023 conference on empirical methods in natural language processing*.

Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., et al. (2024). Deepseek llm: Scaling open-source language models with longtermism. Preprint arXiv:2401.02954.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Chen, T., Xu, B., Zhang, C., & Guestrin, C. (2016). Training deep nets with sublinear memory cost. Preprint arXiv:1604.06174.

Chen, L., Zaharia, M., & Zou, J. (2023). How is ChatGPT's behavior changing over time? Preprint arXiv:2307.09009.

Cheshkov, A., Zadorozhny, P., & Levichev, R. (2023). Evaluation of ChatGPT model for vulnerability detection. Preprint arXiv:2304.07232.

Chiang, W., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., et al. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, mar. 2023.

Choi, S. R., & Lee, M. (2023). Transformer architecture and attention mechanisms in genome data analysis: a comprehensive review. *Biology*, *12*(7), 1033.

Conover, M., Hayes, M., Mathur, A., Meng, X., Xie, J., Wan, J., et al. (2023). Free dolly: Introducing the world's first open and commercially viable instruction-tuned LLM - The databricks blog. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297.

Dao, T., Fu, D., Ermon, S., Rudra, A., & Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, *35*, 16344–16359.

Databricks (2023). Databricks/dolly-v2-12b · Hugging Face. URL https://huggingface.co/databricks/dolly-v2-12b.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. (pp. 4171–4186).

Ding, B., Qin, C., Liu, L., Bing, L., Joty, S., & Li, B. (2023). Is GPT-3 a good data annotator?

Dionísio, N., Alves, F., Ferreira, P. M., & Bessani, A. (2019). Cyberthreat detection from twitter using deep neural networks. In *2019 international joint conference on neural networks* (pp. 1–8). IEEE.

Dionisio, N., Alves, F., Ferreira, P. M., & Bessani, A. (2020). Towards end-to-end cyberthreat detection from Twitter using multi-task learning. In *2020 international joint conference on neural networks* (pp. 1–8). IEEE.

Eleutherai. (2023). URL https://www.eleuther.ai.

Falcon LLM. (2023). URL https://falconllm.tii.ae/falcon.html.

Farooq, A., Awais, M., Ahmed, S., & Kittler, J. (2021). Global interaction modelling in vision transformer via super tokens. Preprint arXiv:2111.13156.

Franco, M. F., Rodrigues, B., Scheid, E. J., Jacobs, A., Killer, C., Granville, L. Z., et al. (2020). SecBot: A business-driven conversational agent for cybersecurity planning and management. In *2020 16th international conference on network and service management* (pp. 1–7). IEEE.

Gilardi, F., Alizadeh, M., & Kubli, M. (2023). ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, *120*(30), Article e2305016120.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., et al. (2022). Training compute-optimal large language models. Preprint arXiv:2203.15556.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., et al. (2021). Lora: Low-rank adaptation of large language models.

Jehangir, B., Radhakrishnan, S., & Agarwal, R. (2023). A survey on named entity recognition—datasets, tools, and methodologies. *Natural Language Processing Journal*, *3*, Article 100017.

Kim, Y. (2015). Convolutional neural networks for sentence classification.

Kocoń, J., Cichecki, I., Kaszyca, O., Kochanek, M., Szydło, D., Baran, J., et al. (2023). ChatGPT: Jack of all trades, master of none. *Information Fusion*, *99*, Article 101861.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, *35*, 22199–22213.

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., et al. (2022). A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology*, *13*(2), 1–41.

Liao, X., et al. (2016). Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In *Proceedings of the 23rd ACM cCS*.

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. Elsevier,

Lipton, Z. C., Elkan, C., & Naryanaswamy, B. (2014). Thresholding classifiers to maximize F1 score. *Stat*, *1050*, 14.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, *55*(9), 1–35.

López Espejel, J., Ettifouri, E. H., Yahaya Alassan, M. S., Chouham, E. M., & Dahhane, W. (2023). GPT-3.5, GPT-4, or BARD? Evaluating LLMs reasoning ability in zero-shot setting and performance boosting through prompts. *Natural Language Processing Journal*, *5*, Article 100032.

McKee, F., & Noever, D. (2023). Chatbots in a honeypot world. Preprint arXiv:2301.03771.

Medsker, L. R., & Jain, L. (2001). Recurrent neural networks. *Design and Applications*, *5*(64–67), 2.

Megahed, F. M., Chen, Y.-J., Ferris, J. A., Knoth, S., & Jones-Farmer, L. A. (2023). How generative ai models such as chatgpt can be (mis) used in spc practice, education, and research? an exploratory study. *Quality Engineering*, 1–29.

Microsoft (2023). Microsoft security copilot. URL https://www.microsoft.com/en-us/security/business/ai-machine-learning/microsoft-security-copilot.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., et al. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, *56*(2), 1–40.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning–based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, *54*(3).

Noever, D., & Williams, K. (2023). Chatbots as fluent polyglots: Revisiting breakthrough code snippets. arXiv preprint arXiv:2301.03373.

Okey, O. D., Udo, E. U., Rosa, R. L., Rodríguez, D. Z., & Kleinschmidt, J. H. (2023). Investigating ChatGPT and cybersecurity: A perspective on topic modeling and sentiment analysis. *Computers & Security*, *135*, Article 103476.

Openai platform. (2023). URL https://platform.openai.com.

Qammar, A., Wang, H., Ding, J., Naouri, A., Daneshmand, M., & Ning, H. (2023). Chatbots to ChatGPT in a cybersecurity space: Evolution, vulnerabilities, attacks, challenges, and future recommendations. Preprint arXiv:2306.09255.

Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., & Yang, D. (2023). Is ChatGPT a general-purpose natural language processing task solver?

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Rasa (2024). Conversational AI Platform | Superior Customer Experiences Start Here. URL https://rasa.com/. Last visited: 13 February 2024.

Ritter, A., et al. (2015). Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th international conference on world wide web*.

Sabottke, C., et al. (2015). Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits. In *Proceedings of the 24th USeNIX security symp.*.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv:2402.07927.

Sanford, C., Hsu, D. J., & Telgarsky, M. (2024). Representational strengths and limitations of transformers. *Advances in Neural Information Processing Systems*, *36*.

Shazeer, N. (2019). Fast transformer decoding: One write-head is all you need. Preprint arXiv:1911.02150.

Sun, X., Dong, L., Li, X., Wan, Z., Wang, S., Zhang, T., et al. (2023). Pushing the limits of ChatGPT on NLP tasks. Preprint arXiv:2306.09719.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., et al. (2023). *Alpaca: A Strong, Replicable Instruction-Following Model*. Stanford Center for Research on Foundation Models, https://crfm.stanford.edu/2023/03/13/alpaca.html.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., et al. (2023). Llama: Open and efficient foundation language models. Preprint arXiv:2302.13971.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wang, E. J. (2024). Tloen/alpaca-lora. URL https://github.com/tloen/alpaca-lora. original-date: 2023-03-13T21:52:36Z.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., et al. (2023). Self-instruct: Aligning language model with self generated instructions.

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., et al. (2024). Harnessing the power of LLMs in practice: A survey on ChatGPT and beyond. *ACM Transactions on Knowledge Discovery from Data*.

Zhang, L., Zhang, Y., Ren, K., Li, D., & Yang, Y. (2023). MLCopilot: Unleashing the power of large language models in solving machine learning tasks. Preprint arXiv:2304.14979.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., et al. (2024). Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, *36*.