The Use of Large Language Models (LLM) for Cyber Threat Intelligence (CTI) in Cybercrime Forums

Vanessa Clairoux-Trépanier[1]*, Isa-May Beauchamp[1]*Estelle Ruellan[2], Masarah Paquet-Clouston[1,3], Serge-Olivier Paquette[2], Eric Clay[2]

White Paper, August 2024

1 School of Criminology, Université de Montréal, Montréal, Québec, Canada
2 Flare Systems, Canada
3 Complexity Science Hub, Vienna, Austria

Abstract

Large language models (LLMs) can be used to analyze cyber threat intelligence (CTI) data from cybercrime forums, which contain extensive information and key discussions about emerging cyber threats. However, to date, the level of accuracy and efficiency of LLMs for such critical tasks has yet to be thoroughly evaluated. Hence, this study assesses the accuracy of an LLM system built on the OpenAI GPT-3.5-turbo model to extract CTI information. To do so, a random sample of 500 daily conversations from three cybercrime forums—XSS, Exploit.in, and RAMP—was extracted, and the LLM system was instructed to summarize the conversations and code 10 key CTI variables, such as whether a large organization and/or a critical infrastructure is being targeted. Then, two coders reviewed each conversation and evaluated whether the information extracted by the LLM was accurate. The LLM system performed strikingly well, with an average accuracy score of 98%. Various ways to enhance the model were uncovered, such as the need to help the LLM distinguish between stories and past events, as well as being careful with verb tenses in prompts. Nevertheless, the results of this study highlight the efficiency and relevance of using LLMs for cyber threat intelligence.

1 Introduction

With the rise of large language models (LLMs), which are text-generating technologies trained on vast amounts of data, the scope and aim of artificial intelligence (AI) have changed. Now, AI can be used for a myriad of applications, such as writing stories on the fly or summarizing complex scientific content. One key application where AI can be useful is cyber threat intelligence (CTI). Indeed, there exists a vast array of cybercrime conversations in forums on both the clear and dark web. These conversations often leak key information that can be used by companies and governments to detect and sometimes even prevent cyber attacks. The question is whether and how LLMs can accurately be used for cyber threat intelligence on such forums. Indeed, can we trust such technology for CTI? Can it replace first-level threat analysts, that is, analysts who read and extract relevant information from cybercrime forums?

This study assesses the extent to which an LLM system is accurate when extracting and summarizing information from cybercrime forums. Using a random sample of 500 daily conversations from three cybercrime forums—XSS, Exploit[.]in, and RAMP—we instructed an LLM to summarize the conversations and extract specific CTI information from them. CTI information included whether a sale was conducted, whether a large organization or a critical infrastructure was discussed, whether initial access to an organization was mentioned, whether a vulnerability that is remotely exploitable or actively

*These two authors contributed equally to the study

exploited was mentioned, and whether users discussed geopolitical conflicts. Any technology mentioned in a conversation was also coded, as well as any industry, when available. Such information is key for CTI: it allows analysts to narrow down cybercrime conversations that focus, for example, on a specific technology in an industry or on specific vulnerabilities targeting large organizations.

To assess the accuracy of the LLM, two coders conducted a thorough review of each conversation and evaluated whether the information extracted by the LLM was accurate. This process 1) determined the level of accuracy of the technology, and 2) highlighted flaws that, once acknowledged, could be fixed.

In the end, the LLM system performed exceptionally well, accurately coding the ten variables, on average, 98% of the time with a minimum of 95% and a maximum of 100%. The assessment of the LLM also highlighted areas of improvement that could be useful for any researcher wishing to use LLMs to summarize forum conversations, including verb tenses in prompts and the impact of using large or vague concepts. The results highlight that large language models can be effectively used for cyber threat intelligence.

2 Methods and Data

The following sections describe the LLM system developed for this study, the CTI variables it coded, and the manual process used to verify the results.

2.1 LLM System for CTI

For this study, the LLM system, powered by the gpt-3.5-turbo-16k-0613 model, is designed to extract and summarize relevant information from cybercrime forums. The system operates through a multi-step process that involves selecting high-quality sources, summarizing conversations, and coding key variables. This section outlines the detailed methodology employed in generating unit summaries, including pseudo-code and an example of a modular prompt.

2.1.1 Data Collection and Preprocessing

The first step involves selecting and collecting data from high-quality cybercrime forums. These forums are continuously monitored, and new messages are extracted daily. The system processes both new and existing discussion threads, ensuring comprehensive coverage of relevant conversations.

2.1.2 Contextual Information Extraction

For each extracted message, the system determines the context based on whether the discussion thread is new or has been previously processed. If the thread is new, the title serves as the context. For existing threads, a summary of the prior conversation is used to provide context.

2.1.3 Prompt Design and Variable Extraction

The core of the system's functionality lies in the use of carefully designed prompts that guide the LLM to extract specific information. Prompts are formulated to capture the intent and requirements for data extraction, such as identifying transactions, potential targets, or vulnerabilities mentioned in the conversation. The prompts are designed from the perspective of a cyber threat intelligence analyst, focusing on the key elements that are critical for threat analysis and reporting. The system's output is structured as unit summaries, which include a text summary of the conversation and the extracted variables in a standardized format. This format facilitates easy analysis and integration into further processing pipelines. We explicitly instruct the LLM to output the information in the specified format.

2.1.4 Example Modular Prompt with Analyst Persona

```
"As a cyber threat intelligence analyst, your task is to review the
    conversation and identify key indicators. Please extract the following
    information: 1. An actor is selling something? 2. The conversation involves
     the sale of initial access to a corporate or organization network? 3.
    Targeted or abused products or technology names?". Output your answer in
    the following format, as an example.
```

```
{
 "summary": "The conversation focused on the sale of a new exploit targeting
     XYZ software. An actor claimed to have discovered a vulnerability and is
     offering it for sale. There was also a discussion about potential targets
     using ABC technology.",
 "variables": {
     "is_sale": true,
     "is_initial_access": true,
     "targeted_technologies": "XYZ software, ABC technology"
 }
}
```

The following pseudo-code outlines the steps taken to generate unit summaries:

```
    FOR each message in daily_batch in the thread DO
    context = daily_batch
    IF thread is new THEN
    context = extract_thread_title(message) + context
    ELSE context += retrieve_thread_summary(message) + context
    END IF
    response = LLM.generate_response(context, PROMPT)
    unit_summary = {
    "summary": extract_summary(response),
    "variables": extract_variables(response) }
    STORE unit_summary
    END FOR
```

In this pseudo-code, PROMPT represents the modular prompt guiding the LLM system, and `generate_response`, `extract_summary`, and `extract_variables` are functions that interact with the LLM model to produce the required outputs.

2.2 Key CTI Variables and Prompts

From daily cybercrime conversations, ten variables were extracted using specific prompts with the LLM. These variables were chosen because they provide key information for CTI, such as which technology or industry is targeted. Having readily accessible and easily identifiable information for CTI ensures that analysts can narrow down their focus to conversations relevant for their organization. Anyone wishing to reproduce the analysis could develop their own key variables. Table 1 presents the variables that were extracted by the LLM for this study.

2.3 Coding Process

To assess the accuracy of the LLM system, a random sample of 500 daily conversations from three cybercrime forums—XSS, Exploit.in, and RAMPs— were extracted using the Flare interface. Flare is an information technology (IT) security company that maintains a cyber threat intelligence platform by monitoring various online spaces[1].

Then, two analysts went over each of the daily conversations from cybercrime forums and assessed the accuracy of the summary coded by the LLM system. They also individually coded the ten variables according to their interpretation, then compared them to the coding performed by the LLM. Specifically, they used a binary coding scheme: 1 indicated agreement with the LLM coding and 0 indicated disagreement.

Once the individual coding was completed, the two analysts performed an inter-coder agreement to obtain a common decision for each unit summary. Such inter-coder agreement allowed them to pinpoint discrepancies, but also find areas of improvement to fine-tune the LLM system. In the end, the inter-coder agreement was high, with an average of 98.2%, a minimum of 96.2% and a maximum of 99.8%. A merged database was created resulting from the agreement. This merged database was used to determine the accuracy of the LLM system in coding each of the ten CTI variables from the 500 daily conversations on cybercrime forums. The results are presented below.

3 Assessing the accuracy of the LLM System for CTI

The LLM system performed strikingly well in coding variables that represented valuable CTI information from cybercrime forums. Indeed, on average, the LLM was accurate in 97.96% of cases, with a minimum

[1] https://flare.io/

Variable Name	Prompt
summary	Generate concise extraction summaries with ample information for effective similarity searches, including conversation outcomes and all technical details that could be used by an analyst to investigate the threat.
is_sale	An actor is selling something.
is_initial_access	Involves the sale of initial access to corporate or organization network.
is_targeting_mainstream	A mainstream software or hardware or a product utilized by many businesses and organizations for its features and security capabilities is being targeted or compromised.
is_targeting_large_organization	A large organization is being targeted by the actors.
is_targeting_critical_infrastructure	A critical infrastructure provider is targeted by the threat.
is_remotely_exploitable	A mentioned vulnerability is remotely exploitable.
is_actively_exploitable	A mentioned vulnerability is being actively exploited.
is_geopolitics	The discussion involves geopolitical issues.
targeted_technologies	Targeted or abused products or technology names.
industries	Names of the industries relevant to the content, including those targeted by threat actors. Choose from the following options, each accompanied by a specific definition: *Finance*: Involving banking, cryptocurrencies, investment, insurance, real estate, stock market, money mule, embezzlement, money laundering, insider trading, shell companies, and other financial services. *Technology and Software*: Involving software development, cryptocurrencies, blockchain, IT services, hardware manufacturing, electronics, and related fields. *Critical Infrastructure*: Covering essential systems and facilities such as energy, oil and gas sector, transportation, water supply, telecommunications, internet providers, military, governments, harbor, airport. *Healthcare*: Involving medical services, hospitals, clinics, pharmaceuticals, biotechnology, emergency centers, and healthcare IT. *Other*: Any industry not explicitly mentioned above. *All*: Indicates that the content may be relevant to all industries.

Table 1: Key information from cybercrime conversations and associated prompts

accuracy of 95% for the variable `is_targeting_critical_infrastructure` and a maximum of 100% for the variable `industries`. The percentage accuracy for each variable is presented in Table 2 below.

For the `summary` variable, the coders ensured that the summary provided by the LLM system was accurate, and it was the case for 98.8% (N=500) of the conversation flows analyzed. However, it must be acknowledged that sometimes, the coders did notice that some key CTI information was missing from the summary. This is because "summarizing" a conversation means cutting text elements, which sometimes may be interpreted as important by some and not by others. The variable `is_sale` was also well coded by the LLM system, with an accuracy of 97.2%. Such variables captured when a user was conducting or announcing a sale. Most cases in which the LLM was wrong related to individuals expressing interest in sales or discussing sales, while not conducting or announcing any. The LLM system also performed really well when coding the variable `is_initial_access`, flagging messages involving the sale of initial access to a corporate or organizational network, with an accuracy of 99%. Hence, almost all posts that involve a sale and/or the sale of an initial access could be narrowed down using these two variables. The variable`is_targeting_large_organization` had an accuracy of 96.2% and `is_targeting_critical_infrastructure` had the lowest accuracy of 95%. For these two variables, the LLM

Variable Name	N	Accuracy
summary	500	98.8%
is_sale	500	97.2%
is_initial_access	500	99.0%
is_targeting_large_organization	500	96.2%
is_targeting_critical_infrastructure	500	95.0%
is_remotely_exploitable	500	98.2%
is_actively_exploitable	500	99.4%
is_geopolitics	500	96.0%
targeted_technologies	115	99.1%
industries	273	100%

Table 2: LLM System Accuracy for each CTI Variable

coding errors related mainly to definitions, specifically, defining what is a "large" organization or what is a "critical infrastructure". Analysts could still use these two variables to assess which large organization and/or critical infrastructure is discussed in cybercrime forums.

The variable `is_remotely_exploitable`,flagged conversations in which a vulnerability that is remotely exploitable was mentioned. This variable was accurately coded 98.2% of the time. The LLM system also performed well (score of 99.4%) when coding the variable `is_actively_exploitable`, which aimed at flagging conversations in which a vulnerability actively exploited was mentioned in cybercrime conversations. These two variables combined can narrow down conversations on relevant vulnerabilities that organizations should prioritize in fixing.

The variable `is_geopolitics` aimed at flagging any conversations that related to geopolitical issues. It was accurately coded 96% of the time by the LLM. For this variable, the LLM tended to code the variable as `TRUE` when a country was mentioned, particularly if the country was associated with a tense geopolitical context. Hence, analysts could use this variable to narrow down conversations that may target specific countries or geopolitical conflicts.

For the variable `targeted_technologies`, the coders made sure that the name of a technology mentioned in the message matched the technology indicated by the LLM. For all messages that had a technology (N=115), the accuracy was 99.1%, indicating that the LLM missed a technology really rarely. Given the high accuracy of the variable, analysts could use this variable to assess whether their technologies are discussed in cybercrime forums.

Finally, the variable `industries` was accurately coded 100% of the time, the highest accuracy score. This variable was coded when an industry was mentioned in a message (N=273). Such a high score might be due to this variable being well detailed, which seemed to contribute to the LLM's effectiveness in correctly identifying the appropriate industry categories indicated in the messages. With this variable, analysts -and even policy makers- can narrow down conversations in cybercrime forums that target specific industries, thus facilitating their risks assessments.

4 Coders' Insights to Fine-Tune the LLM System

Given the results of the analysis, there is no doubt that LLM systems can be useful in extracting key CTI information from cybercrime forums. Indeed, the summaries generated from daily conversations focused on relevant information, even when a large number of messages were posted. Such relevant information was sometimes even missed by the analysts. The LLM also demonstrated an exceptional ability to code key CTI variables, as evidenced by the results obtained: it had an accuracy of 98% on average. Such high results were unexpected and showcase interesting future research avenues.

Reviewing the same conversations allowed the coders to discuss and identify areas for improvement in coding the CTI variables. These areas are useful for any researcher or analyst wishing to use LLMs to code variables based on text input. They are presented below.

4.1 Difficulties in Detecting Stories and Past Events

Across variables, the small number of observations that were miscoded often involved stories or past events mentioned by a user. Indeed, the LLM sometimes encountered difficulties when processing past

events reported by users. For example, the LLM system coded the variable `is_sale` as `TRUE` for a conversation flow that was summarized as follows:

> "A police officer was caught selling fake certificates through a darknet market. The price per certificate was $9,000, and during a bulk sale of $80,000 worth of certificates, the officer attempted to deliver them personally, leading to his arrest." (382)

However, in this message, the user was reporting a past event. Consequently, the LLM incorrectly categorized this example as a case of a user selling certificates rather than understanding it as a narrative of a past event involving a third party. Similarly, the LLM coded the variable `is_sale` as `TRUE` for a conversation that was summarized as follows:

> "[An actor] reported that Russian national Evgeny Doroshenko has been charged in the USA for working as an initial access broker. Doroshenko is suspected of hacking at least one company in New Jersey and has been providing similar services since 2019. He set the starting price for access to the compromised company at $3,000, with an auction increment of $500 or an instant sale price of $6,000. His preferred attack method is brute-forcing Remote Desktop Protocol services. Doroshenko's personal information, including phone numbers and home address in Astrakhan, was found linked to his Telegram account. Following the news of the charges, he attempted to contact moderators on the Exploit forum." (275)

Again, in this conversation, a past event was reported, and therefore, the LLM should not have coded it as an active sale. When the LLM incorrectly coded such messages as `TRUE`, it over-represents the number of sales in the dataset. Hence, it is important to enhance the LLM's ability to distinguish historical narratives from current facts to ensure a more accurate and contextually appropriate analysis of the information provided by users.

4.2 Verb Tenses in Prompts

The coders also noticed that, given that the prompts were written in the present tense and the unit summaries were written in the past tense, the LLM sometimes had difficulties in coding accurately. For example, the variable `is_sale` was not coded as `TRUE` for the conversation flow related to this summary:

> "Two actors participated in a conversation. [the first actor] posted a message that was likely truncated and is not informative on its own. Actor [2] indicated that a previous discussion or sale has been closed and the item in question has been sold. No further details are provided." (60)

Most likely, the verb tense in the prompt, "an actor is selling something," was the reason why the LLM did not code this conversation as involving a sale. However, upon reflection, it could be considered a sale because the item in question has been sold. In the end, if the goal is to code all sales, past or present, it might be necessary to revisit the verb tenses in the prompts given to the LLM.

4.3 The Importance of Data Chunking

During the coding process, the coders went through the `summary` variables and then the whole conversation on the cybercrime forum to validate that no information was missing and subsequently assess the accuracy of the CTI variables. When reading the complete conversation, the coders noticed that the way the data was chunked influenced the results. For example, given this summary:

> "No actionable intelligence was extracted from the new message by [an actor] dated [xx-xx-xx] as it contained no specific information related to the concepts of interest for threat intelligence." (165)

Almost all variables were coded as `FALSE`. However, when reading the complete conversation, one could quickly see that the discussion revolved around selling a database containing various Telegram chats and channels. The main user made various updates on the database's availability and scope, including traffic sources and promotional offers. However, the posts spanned many days. Hence, although the LLM's result was accurate, the way the data was chunked led to the conclusion that the conversation did not revolve around a sale, although it did.

It is important to consider that how data is chunked and fed to the LLM may lead to an under-representation of some variables, as in the example above, or an over-representation if multiple "pieces" of the same thread exposing one single sale, for example, were coded in several places. One might wonder whether taking the whole thread would be more efficient than daily conversations. Such a decision may depend on the length of the threads, as threads that span many days may lead to various discussion avenues. In the end, such observations have to be taken into account when coding the LLM system.

4.4 About Coding General or Vague Concepts

When coding CTI information, some variables relate to specific well-defined concepts, such as the variable `industries`. Indeed, the prompt for `industries` was well defined and included clear descriptions of industries, such as finance, technology, or critical infrastructure. On the other hand, other variables referred to general or vague concepts. For example, the coders noticed that some organizations that could be considered "large", and therefore coded as TRUE for the variable `is_targeting_large_organization`, were not coded as such by the LLM system. Since no definition was provided for what constitutes a "large organization," this variable was subject to the LLM's interpretation. This means that only organizations recognized as such by the LLM were identified, or —as the coders noticed— only without-a-doubt large organizations were flagged (e.g., Apple, Netflix, Microsoft). Large organizations that were less known or required additional searches were not identified.

This is not ideal, but also not completely problematic. By leaving the concept to be interpreted by the LLM, at least well-known organizations were identified. Hence, there is a certain interest in leaving some concepts subject to the LLM's interpretation, especially given the high accuracy rate for all variables. Still, further research on the topic is needed.

4.5 Links between Variables with Similar Concepts

During the analyses, the coders noticed some disparities between interrelated variables such as `industries`, `is_targeting_critical_infrastructure`, and `is_targeting_large_organization`. For example, consider this summary:

> "A user announced the sale of an SQL Injection vulnerability in the Peru Military Document Management System. This vulnerability allows access to documents, users, passwords, and more, with the server holding over 50 databases under the domain 'mil.pe'." (259)

In this example, the LLM coded `FALSE` for `is_targeting_critical_infrastructure` and `TRUE` for `is_targeting_large_organization`. In the `industries` variable, the LLM indicated ['Critical Infrastructure', 'Military']. This discrepancy raises questions about the definitions of these three variables, as discussed above, and how the LLM understood them. It also showcases that some variables may be used to code others. For example, a technology mentioned in the summary was not always associated with an organization, despite the obvious connection between the two. In this excerpt:

> "A user expressed interest in purchasing generated iCloud accounts, offering a rate of $0.5 per account" (116)

The LLM indicated "iCloud" in the `targeted_technologies` variable but did not indicate "Apple" in the `targeted_organization` variable. Linking the two variables could enhance the information coded.

4.6 The Title as Key

As a reminder, the LLM system codes variables based on the conversation flows and is given some context. Such context is either the title or, if part of the conversation was already coded by the LLM, the summary of the previous conversation. The coders noticed that often, the title provided key information that, if given to the LLM, would have avoided some coding errors. For example, here is an LLM summary:

> "An actor expressed interest in purchasing data, with a priority on information from Israel and less emphasis on data from Iran." (312)

From this conversation, the LLM coded `is_targeting_critical_infrastructure` as `FALSE`, which is accurate given the summary. However, the thread's title is "Buy GOV access." Hence, with

this title, the LLM should have coded `is_targeting_critical_infrastructure` as `TRUE` because the access the user wishes to purchase belongs to a government. Since the title was not included in the LLM's results, it missed this important detail. Potentially, adding the title with the previous summary in the context of the prompt could reduce coding errors. This relates, as well, to how the data is chunked and subsequently interpreted, as discussed above.

4.7 The LLM System is Imperfect, just like Humans

One last element that needs to be considered is that there were instances where the coders simply did not understand the logic behind the LLM's coding. For example, given this summary:

> "A user advertised a product called 'Red Node HVNC', which is a hidden VNC (Virtual Network Computing) tool that allows remote control over a target machine via a web browser. [The actor] claims the tool is fully undetected (FUD) by antiviruses, including Windows Defender, and does not require a crypter for distribution. A demonstration video and a feature comparison chart were provided. Another user criticized the product, suggesting it is overpriced and questioning the lack of moderation for such products. [The principal actor] defended their product, stating it is privately coded, FUD, fast, and the first HVNC to work over a web browser, unlike others cloned from 'tinynuke'." (160)

The LLM coded the variable `is_initial_access` as `FALSE` even though the actor sells a tool that allows remote control over a target machine via a web browser. Hence, in some cases, the LLM missed the information. The LLM system was therefore not flawless, just like humans. Although rare, it made coding errors that could not be explained.

5 Conclusion

Cyber threat intelligence has traditionally been plagued by a lack of actionability and challenges around making effective use of analyst time. Our research indicates that there is significant potential for leveraging LLMs to conduct an initial review of original source data to categorize events and prioritize those that may be most relevant to CTI teams. This presents a substantial opportunity to use analyst time more effectively by narrowing searches and automatically alerting on relevant findings.

When language models go wrong, they tend to do so in predictable ways. Providing the LLM with the context needed to categorize the event was important, and it did sometimes miscategorize events, particularly when the prompt left room for interpretation. Another potential error was introduced when conversations spanned multiple days, so the summarization may have missed crucial context, thus resulting in an incorrect label. Our analysis also showed that the model struggled with differentiating between stories and original events occurring.

Despite these issues, OpenAI's GPT-3.5-turbo model was able to produce exceptional results across our study of 500 events. This indicates language models are currently a scalable and cost-effective solution for identifying relevant CTI data and could have exceptional potential when paired with state-of-the-art (SOTA) models for reporting data and fine-tuning results. Leveraging SOTA models, such as Claude 3.5 Sonnet or GPT-40, has the potential to further improve results or allow for categorization of even more complex data. This represents a significant opportunity for further study.

Acknowledgments

This study was supported by a mini-grant from the Human-Centric Cybersecurity Partnership (HC2P).