

Upotreba LLM-ova prilikom rukovanja sigurnosnim incidentima

Gabelica, Benedicte Amelie Zorana

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:713663>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-14**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1661

**UPOTREBA LLM-OVA PRILIKOM RUKOVANJA
SIGURNOSNIM INCIDENTIMA**

Benedicte Amelie Zorana Gabelica

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1661

**UPOTREBA LLM-OVA PRILIKOM RUKOVANJA
SIGURNOSNIM INCIDENTIMA**

Benedicte Amelie Zorana Gabelica

Zagreb, lipanj 2024.

ZAVRŠNI ZADATAK br. 1661

Pristupnica: **Benedicte Amelie Zorana Gabelica (0036543247)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: izv. prof. dr. sc. Stjepan Groš

Zadatak: **Upotreba LLM-ova prilikom rukovanja sigurnosnim incidentima**

Opis zadatka:

Tijekom rješavanja sigurnosnih incidenata, osobe koje upravljaju i donose odluke koriste svoja prethodna stečena znanja i iskustva kako bi se snašli u novoj situaciji. Pri tome nemaju nikakvo navođenje u smislu preporučenih sljedećih koraka. S jedne strane, kako nema puno stručnih osoba s tim znanjima, dobro bi bilo imati sustav koji bi pomagao u rješavanju incidenta. S druge strane, tijekom uvježbavanja ljudi za djelovanje u incidentnim situacijama koriste se popisi akcija koji na neki način navode i daju ideje što bi trebalo učiniti, a što nije reprezentativna situacija. U završnom radu potrebno je istražiti mogućnosti korištenja LLM-ova kao što je ChatGPT kao pomoć u rješavanju sigurnosnog incidenta. Osoba koja upravlja s incidentom bi kroz konverzaciju trebala navoditi svoje željene akcije koje bi potom LLM pretvarao u konkretne akcije, a potencijalno i savjetovao korisnika što učiniti u pojedinoj situaciji. Za ispitivanje mogućnosti korištenja treba upotrijebiti simulator CCS.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

1. Uvod	3
2. Arhitektura i korištenje LLM-ova	5
2.1. Korištenje LLM-ova u svrhe odlučivanja	6
2.2. Parametri modela	6
2.3. Prompt inženjerstvo	7
2.3.1. Few-shot prompting	8
2.3.2. Chain-of-thought prompting	8
2.3.3. Self-consistency	8
2.3.4. Retrieval augmented generation	8
2.4. Korišteni modeli	10
3. Cyber Conflict Simulator	11
3.1. Akcije u CCS-u	12
4. Ispitivanje s ChatGPT modelom	13
5. Lokalno korištenje LLM-ova	15
5.1. LangChain	15
5.2. Kvantizacija modela	16
5.3. Opis RAG postupka u LangChainu	17
6. Ispitivanje korištenjem LLM-ova lokalno	19
6.1. Priprema za izvođenje	19
6.2. Python datoteka korištene za testiranje	20
6.3. Scenariji korišteni pri ispitivanju	21
6.3.1. Scenarij obrane	22

6.3.2. Scenarij napada	22
6.4. Ispitivanje obrane koristeći LLama2	23
6.4.1. Početno ispitivanje	23
6.4.2. Ponovno ispitivanje	25
6.4.3. Posljednje ispitivanje obrane	26
6.5. Ispitivanje napada koristeći Mixtral	28
6.6. Rezultati ispitivanja	29
7. Fino podešavanje modela	31
8. Zaključak	33
Literatura	35
Sažetak	38
Abstract	39
A: Programski kod i baza primjera za fino podešavanje	40
B: Aproksimacija troškova pri korištenju gpt-4-turbo modela	41
C: Potpuni rezultati ispitivanja	43

1. Uvod

Veliki su jezični modeli (eng. *Large Language Models*, u nastavku LLM-ovi) naziv kojim se opisuju modeli strojnog učenja s brojem parametara reda veličine jedne milijarde ili više. U procesu učenja tih modela koristile su se baze podataka konstruirane iz web stranica, javno dostupnih foruma te programskih kodova, čija se veličina mjeri u terabajtima [1]. U posljednjih deset godina LLM-ovi su doživjeli nagli razvoj, koji se znatno ubrzao 2022. godine predstavljanjem OpenAI-jevog ChatGPT-a [2]. Razvijeni chatbot, s kojim se može komunicirati preko jednostavnog i besplatnog korisničkog sučelja, počeo se koristiti za razne potrebe kao što su pisanje programskog koda i tekstova, odgovaranje na pitanja i druge. Danas je uz OpenAI sve veći broj tvrtki koje razvijaju svoje LLM-ove, a neke od najpoznatijih su Meta (LLama modeli), Google (Gemini modeli), Amazon (Titan modeli) te Mistral AI.

S porastom popularnosti LLM-ova vidljivo je njihovo sve veće objedinjavanje s postojećim sustavima, najviše u području obrade prirodnog jezika. Jedan od načina korištenja ovih modela su chatbotovi na web stranicama, koji korisnicima nude brz odgovor na njihove upite. Još jedna mogućnost je korištenje LLM-ova u sklopu aplikacija koje zahtijevaju obradu korisničkog teksta te stvaranje određenog izlaza na temelju analiziranih podataka [3]. Ovakvi pristupi imaju ograničenja vezana uz doseg podataka nad kojima su trenirani modeli. U situacijama gdje se od LLM-ova očekuje poznavanje specifične domene znanja koja nije javno dostupna, moguća je pojava tzv. haluciniranja gdje model generira netočne informacije zbog nepoznavanja točnog odgovora.

U svakodnevnom razvoju sve većeg broja web aplikacija u sklopu kojih se upravlja velikom količinom privatnih podataka korisnika, nužno je imati dobro definirane sigurnosne zahtjeve te osigurati da su oni uvijek ispunjeni. Kada jedan od zahtjeva nije ispunjen, tj. kada je došlo do incidenta, važno je brzo i ispravno reagirati kako bi se šteta

minimizirala. Kako bi to bilo lako ostvarivo, potrebno je imati definirane procedure za svaki potencijalni napad te na temelju njih educirati sigurnosne stručnjake.

U ovom radu istražuje se mogućnost korištenja LLM-ova kao pomoć u rukovanju incidentima. Na primjeru *Cyber Security Simulatora* ispituje se kvaliteta odgovora LLM-a na upite vezane uz sigurnosne incidente. Raznim tehnikama prompt inženjerstva pokušava se povećati ispravnost dobivenih izlaza LLM-a koji nije prilagođen domeni računalne sigurnosti te navedenog simulatora. Osim obrane, ispituje se i kvaliteta odgovora LLM-a prilikom pružanja pomoći napadačima u izvršavanju napada.

2. Arhitektura i korištenje LLM-ova

LLM-ovi se temelje na arhitekturi umjetnih neuronskih mreža, a većina njih na arhitekturi transformer. Arhitektura transformer sastavljena je od više slojeva različitih funkcionalnosti koji se dijele na enkoder slojeve, zadužene za procesiranje ulaznog teksta, te dekoder slojeve, zadužene za generiranje teksta. Današnji najpopularniji LLM-ovi, poput GPT-4 i Llama modela, imaju samo dekoder slojeve (*decoder-only* arhitekturu) [4].

Na visokoj razini, korištenje LLM-a svodi se na prosljeđivanje ulaznog teksta modelu (tzv. *prompt*). Model raščlanjuje prompt u tokene te određenim mehanizmima pažnje svakom tokenu dodijeli vrijednost ovisno o njegovoj važnosti. Na temelju tih vrijednosti i prethodno generiranih tokena, model generira idući izlazni token, a u konačnici cijeli izlazni tekst [4]. S obzirom na potrebu obrade velike količine ulaznih tokena, u idealnom slučaju paralelno s ciljem bržeg generiranja odgovora, za treniranje i fino podešavanje (eng. *fine-tuning*) LLM-ova potrebne su znatne količine GPU resursa [1].

Tvrtke koje razvijaju LLM-ove najčešće objavljuju dvije verzije istoga modela: tzv. *base* model te *chat* model. Iz objavljene dokumentacije tvrtke Meta [1], poznato je da je njihov *base* model Llama2 treniran samonadziranim učenjem. *Base* modeli generiraju izlaz tako da pokušavaju nadopuniti prosljeđeni ulaz, no nisu prilagođeni poštivanju korisničkih uputa. Iz *base* modela se korištenjem nadziranog učenja modele može prilagoditi na komunikaciju s korisnicima čime se ostvaruje *chat* verzija modela. Ovaj proces obuhvaća učenje modela na specifičan oblik prompta i konverzacijski prilagođen stil odgovora. U zadnjoj fazi podešavanja LLM-a koristi se podržano učenje gdje se modele uči na odbijanje određenih tema razgovora. LLM-ovi mogu odbiti odgovoriti na korisničke zahtjeve vezane uz pomoć pri organizaciji ilegalnih aktivnosti, mišljenje o kontroverznoj temi itd. Restrikcije se uvode kako bi se osiguralo poštivanje sve većeg broja regulativa

vezanih uz umjetnu inteligenciju kao što je *Akt o umjetnoj inteligenciji* u Europi [5].

2.1. Korištenje LLM-ova u svrhe odlučivanja

Obrada prirodnog jezika je jedno od područja umjetne inteligencije koje se naglo razvilo zahvaljujući dubokim neuronskim mrežama. LLM-ovi su danas korišteni zbog mogućnosti razumijevanja ljudskog teksta te generiranja smislenog odgovora [6].

Prije razvoja današnjih LLM-ova, za potrebe određene aplikacije trenirao se model ispočetka ili se koristio postojeći LLM koji bi se dodatno trenirao nad prilagođenim skupom podataka. Postojeći LLM-ovi nisu davali zadovoljavajuće rezultate jer bi nerijetko generirali besmislene rečenice. Tadašnji model OpenAI tvrtke, GPT-2, imao je 1.5 milijardi parametara, što je znatno manje od današnjeg GPT-4 modela koji ima 1.76 trilijun parametara. S obzirom na slabe performanse početnog modela, tako prilagođeni modeli nisu bili dovoljno dobri za korištenje u obradi prirodnog jezika [7].

Pojavom GPT3.5 modela na kojem se temelji ChatGPT započeo je razvoj snažnijih LLM-ova koji se mogu koristiti samostalno ili u sklopu aplikacija bez dodatnog učenja ili fine prilagodbe. Lako ih se prilagodi i domenama gdje postoji specifičan rječnik zahvaljujući prompt inženjerstvu te namještanju njihovih parametara [3, 6].

2.2. Parametri modela

Parametri modela se odnose na težine i pragove u neuronskoj mreži. Veći broj parametara vežemo uz kvalitetniji izlazni tekst. Tijekom procesa treniranja parametri se prilagođavaju kako bi se dobio što kvalitetniji model. Neki parametri mogu se podesiti i na kraju procesa učenja kako bi se LLM-ovo ponašanje prilagodilo određenom cilju [8].

Temperatura modela utječe na kreativnost odgovora tako da viša temperatura modela uzrokuje generiranje raznolikijih odgovora. Kod većine LLM-ova, temperatura je broj između 0 i 1. Ovaj podatak primjenjuje se na kraju neuronske mreže i direktno utječe na vjerojatnost odabira pojedine riječi [9]. Model s temperaturom 0 bio bi deterministički te bi se u tom slučaju za neki ulaz uvijek dobio isti izlaz.

Top-k i top-p su parametri kojima se također može upravljati vjerojatnošću odabira

pojedine riječi. Top-k je broj koji definira broj riječi od početka liste opcija, sortirane po vjerojatnosti, koji će se uzeti u obzir. Top-p je broj koji određuje postotak najboljih riječi koje će se uzeti u obzir. Ovi parametri se ne koriste ako je već podešena temperatura jer utječu na istu odluku modela.

Max tokens je parametar kojim se modelu prosljeđuje informacija koliko maksimalno tokena smije generirati kao odgovor pri jednom upitu. Jedan token odgovara duljini 4 znaka [10].

Jedno od najvećih ograničenja LLM-ova jest veličina kontekstnog prozora, parametar koji označava koliko maksimalno tokena model može odjednom obraditi. Ovaj parametar ima maksimalnu vrijednost koja ovisi o modelu te je zapisana u njegovoj dokumentaciji. Pri korištenju modela potrebno je osigurati da ulaz nikada nije duži od kontekstnog prozora.

2.3. Prompt inženjerstvo

Glavni način kojim se utječe na izlaz LLM-a jest prompt inženjerstvo (eng. *prompt engineering*), područje koje se bavi dizajniranjem ulaza kojima se postižu najbolji rezultati izlaza. Dijelovi jednog prompta su:

- instrukcija - objašnjenje LLM-u koja je njegova uloga, zove se i sistemski prompt (eng. *system prompt*)
- kontekst - vanjske informacije koje mogu pomoći modelu
- ulazni podaci - pitanje korisnika, informacije o stanju sustava i drugi
- indikator izlaza - opis željenog formata izlaznog teksta

Kvalitetnim izlazom smatramo izlaz koji odgovara potrebama korisnika. Kvalitetniji izlazi postižu se uz sažete i jasne ulaze. Ovisno o potrebnom izlaznom tekstu, u okviru prompt inženjerstva postoje različite tehnike kojima se modele potiče na generiranje kvalitetnijeg izlaza. Neke od njih opisane su u nastavku [11].

2.3.1. Few-shot prompting

Zero-shot prompting je prosljeđivanje pitanja modelu bez da mu se pritom prilože dodatne informacije (kontekst) ili format izlaza. Za razliku od toga, *few-shot prompting* je tehnika kojom se modelu prvo daje na raspolaganje nekoliko primjera pitanja i očekivanih odgovora, a tek nakon toga mu se prosljeđuje pitanje na koje se očekuje da će model sam odgovoriti. Ovo je proces tzv. *in-context* učenja gdje model nauči ono što se očekuje od njega na temelju primjera te se onda generira odgovor. Primjeri se moraju prosljeđivati prije svakog novog pitanja jer ova vrsta učenja ni na koji način ne mijenja neuronsku mrežu, tj. model ne pamti ovo što u ovakvom procesu nauči.

2.3.2. Chain-of-thought prompting

Glavni cilj ove tehnike jest navesti model na ulančano logičko zaključivanje kako bi došao do točnog odgovora. Model osim odgovora generira i niz činjenica i zaključaka koji su ga naveli na taj odgovor. Time se smanjuje mogućnost greške ili je korisniku vidljivo koji je krivi zaključak modela uzrokovao loš odgovor. Ova tehnika se može primijeniti u *zero-shot* uvjetima, kada se na kraju inicijalnog prompta dodaje "*Let's think step by step*" rečenica. Model ovu rečenicu interpretira kao početak svog odgovora te time generira izlaz gdje u svakom koraku opravdava svoje zaključke. Za bolje rezultate potrebno je tehniku primijeniti u *few-shot* uvjetima u kojima se modelu prosljeđuju i primjeri takvog načina generiranja odgovora. Izmišljanje takvih primjera može biti dugotrajan posao.

2.3.3. Self-consistency

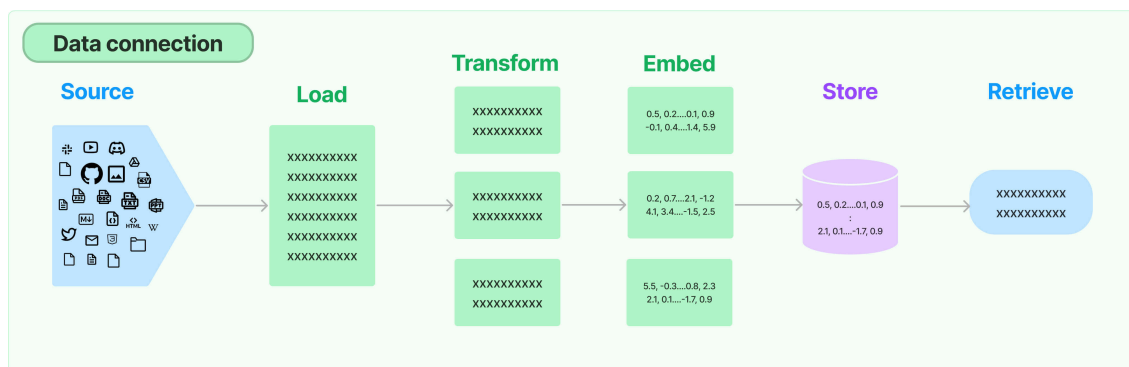
Pod ovom se tehnikom podrazumijeva višestruko generiranje odgovora za isti ulazni tekst. Ova tehnika koristi se u sustavima u kojima se od LLM-a očekuje konkretna akcija ili broj kao odgovor. U tim slučajevima se kao konačan odgovor uzima onaj s najvećom frekvencijom pojavljivanja.

2.3.4. Retrieval augmented generation

Jedna od najvažnijih tehnika prompt inženjerstva u sustavima sa specifičnom domenom podataka jest *Retrieval augmented generation* (RAG). U ovoj metodi se LLM-u uz pitanje prosljeđuje i kontekst, koji je najčešće dio dokumentacije sustava, potreban za odgovara-

nje na pitanje. Za određivanje dijela dokumentacije potrebnog za zadovoljavanje upita, koristi se dodatan model, tzv. *retriever*, koji će na temelju pitanja korisnika na određeni način naći relevantne dijelove u dokumentaciji te ih proslijediti LLM-u. Jedan od razloga zašto se LLM-u prosljeđuje samo dio dokumentacije jest taj da LLM češće halucinira ako mu se u upitu pošalje velika količina nebitnih informacija. Još jedan razlog je ograničena veličina kontekstnog prozora LLM-a, tj. teksta koji se uzima u obzir prilikom generiranja odgovora. Ova se metoda dijeli na dvije faze:

- Faza izvlačenja podataka (*retrieval phase*) - *retriever* iz danih dokumenata izvlači relevantne dijelove, prikazano na slici 2.1.
- Faza generiranja odgovora - LLM koristeći izvučene informacije odgovara na početno pitanje korisnika



Slika 2.1. Prikaz faze izvlačenja podataka [12]

Modeli *retrievera* za određivanje najrelevantnijih dijelova dokumentacije ne koriste tekst u izvornom obliku, već dane dokumente podijele na manje komade teksta (tzv. *chunks*) te ih ugrađuju (eng. *embedaju*), tj. predstave s pomoću vektora, čime se ostvaruje bazu podataka vektora (tzv. *vector store*). Pitanje korisnika se prilikom upita također pretvori u vektor te se tako uspoređuje s dijelovima dokumentacije. Iz dokumentacije se izvlače dijelovi najsličniji pitanju koristeći neki od algoritama za uspoređivanje vektora. Reprerentacijom dijelova teksta vektorom omogućuje se očuvanje njihove semantičke vrijednosti.

2.4. Korišteni modeli

U okviru ovog Završnog rada, ispitivanja kvalitete odgovora provedena su nad tri modela. Prvi ispitani model je ChatGPT. Ovaj model je razvijen iz OpenAI modela GPT3.5 ili GPT4 s dodatnom fazom podržanog učenja kako bi bio prilagođen razgovoru s korisnicima [2]. Dostupan je preko besplatnog korisničkog sučelja.

Drugi LLM čija se kvaliteta odgovora ispitala je Llama2. Ovaj naziv označava niz LLM-ova koje je razvila tvrtka Meta. Za razliku od OpenAI modela, ovi su LLM-ovi otvorenog koda, što znači da ih se može besplatno koristiti lokalno, ako je na raspolaganju dovoljno GPU resursa. Korišten je model od 70 milijardi parametara fino podešen do *chat* verzije: llama2-70b-chat. Za svrhe ispitivanja, korištena je kvantizirana verzija llama-2-70b-chat.Q5_K_M.gguf. Format ulaza za ovaj model je oblika:

```
<s>[INST] <<SYS>>
{{ system_prompt }}
<</SYS>>
{{ user_message_1 }} [/INST] {{ model_answer_1 }} </s>
<s>[INST] {{ user_message_2 }} [/INST]
```

Posljednje ispitani model je Mistral. Mistral predstavlja niz modela otvorenog koda tvrtke Mistral AI. Za potrebe ovoga rada, korišten je model od 47 milijardi parametara fino podešen do *chat* verzije: mixtral-8x7b-instruct-v0.1. Korištena je kvantizirana verzija mixtral-8x7b-instruct-v0.1.Q6_K.gguf. Očekivani format ulaza je oblika:

```
<s>[INST] Instruction [/INST] Model answer</s>[INST]
Follow-up instruction [/INST]
```

3. Cyber Conflict Simulator

Cyber Conflict Simulator (skraćeno: *CCS*) je simulator kibernetičkog okruženja u kojem se vjerno mogu prikazati topologije organizacija, njihovi resursi, aktori te njihovo međusobno djelovanje. Glavna svrha simulatora je simuliranje kibernetičkih napada i obrane od istih. *CCS* se dijeli na dva dijela: *Editor* i *Simulator*.

U *Editoru* se postavljaju strukture napadačkih, obrambenih i vanjskih organizacija. U ovom dijelu simulatora kreiraju se objekti te postavljaju njihovi atributi. Glavni atribut svakog objekta je labela koja određuje vrstu objekta. Neke od labela su *Organisation*, *Machine*, *Actor*, *Malware* i druge. Prilikom kreiranja organizacije, objektu se dodjeli labela *Organisation*. Za svaku organizaciju određuju se računala koji joj pripadaju (labela *Machine*), njihova uloga (*Mail server*, *Web server*) te resursi koji se na njima nalaze. Za svakog se aktora (labela *Actor*) određuje kojim računalima ima pristup, koje su njegove email adrese te koje vještine ima aktor (*System administration*, *Offensive*, *Forensics* i drugi). Prilikom dizajniranja napadačke organizacije, potrebno je napadačima na raspolaganje staviti različite objekte s labelama *Exploit* i *Malware*, budući da se za vrijeme simulacije oni ne mogu kreirati [13].

U *Simulator* dijelu *CCS*-a može se pokrenuti simulacija željenog scenarija, prethodno kreiranog u *Editoru*. Za svakog igrača se može odabrati hoće li njima u simulaciji upravljati prava osoba ili AI.

Cilj *CCS*-a je tvrtkama pružiti mogućnost održavanja kibernetičkih vježbi. Prednost održavanja vježbi u simuliranom prostoru su smanjeni troškovi jer nema rizika od oštećenja prave opreme. U simulatoru postoje i objekti koji precizno mogu mjeriti različite vrste gubitka - od financijskih do procurivanja osobnih podataka. Zahvaljujući izvještajima nakon svake simulacije, organizacija može utvrditi što je točno uzrokovalo gubitke

te na koji način zaposlenici reagiraju prilikom kibernetičkog napada.

Naglasak CCS-a nije na tehničkom znanju prilikom provođenja napada ili obrane. CCS ne omogućuje proučavanje programskog koda zloćudnog programa za svrhe reverziranja, postupak analize stvarnih podataka prilikom forenzičke analize i slično. CCS je napravljen za uvježbavanje tijekom napada, tj. odlučivanja o izvršavanju određenih radnji u danom trenutku. Svaka moguća radnja modelirana je akcijom u simulatoru.

3.1. Akcije u CCS-u

Tijekom simulacije, svaki aktor ima dostupan niz akcija koje je u danom trenutku moguće izvršiti. Akcije u simulatoru dijele se na ofenzivne te defenzivne, a neke akcije mogu se izvršavati i od strane napadača i od obrane [13]. Glavni dijelovi akcije su:

- Ime akcije
- Opis akcije
- Potrebna vještina aktora za izvršavanje akcije
- Tip akcije

4. Ispitivanje s ChatGPT modelom

Pri prvotnom ispitivanju korištenjem ChatGPT-a, iz popisa akcija odabran je manji podskup napadačkih i obrambenih akcija te njihovih opisa (druge informacije o akciji su izbačene). ChatGPT-u se prosljeđuje prompt u sljedećem formatu:

1. Opis uloge modela
2. Popis akcija s opisima
3. Primjeri ulaza i odgovarajućih izlaza
4. Novi ulaz (pitanje korisnika) s napomenom "*Let's think step by step*"

U ovome se primjeru koristi više tehnika prompt inženjerstva. *Few-shot prompting* se koristi za prosljeđivanje primjera parova (ulaz, očekivani izlaz) modelu. *Chain-of-thought prompting* (skraćeno CoT) koristi se dodavanjem rečenice "*Let's think step by step*" na kraju svakog novog ulaza. U primjerima (ulaz, očekivani izlaz) proslijeđenim modelu je očekivani izlaz također napisan tako da se LLM-a navodi na korištenje CoT tehnike.

Prilikom ispitivanja, uočeno je nekoliko problema. Najveći problem bila je šarolikost odgovora modela - model je za iste ulaze davao različite izlaze. Zbog neformatiranog izlaza koji je sastavljen od više rečenica, ne postoji jednostavan način kojim bi se iz izlaza moglo izvući ime potrebne akcije. Prisutan je i problem gdje model prilikom nekih upita napadačima preporuči akciju kojoj pristup imaju samo branitelji i suprotno.

Kako bi se pokušali riješiti ovi nedostaci, provedeno je novo ispitivanje sa sličnim dijelovima prompta kao i u prethodnom primjeru, no ovoga puta se od modela očekuje formatirani izlaz. Modelu su u primjerima ulaza i očekivanih izlaza proslijeđeni formatirani izlazi u sljedećem obliku:

1. Proces razmišljanja (CoT)
2. LOGS (da ili ne) i LOG REASON (zašto) - označava je li potrebno da korisnik pogleda u logovima (bitno za neke akcije, npr. čekanje beacon signala nakon slanja zloćudnog maila)
3. ACTIONS - popis imena akcija koje model preporučuje korisniku
4. REASONS - za svaku akciju u jednoj ili više rečenica opisano zašto je ta akcija dobar izbor

Ovaj format izlaza omogućuje lako parsiranje imena akcija koje su preporučene. Ovime se može koristiti i *self-consistency* tehnika u kojoj se modelu više puta prosljeđuje isti ulaz te se kao izlaz uzme akcija koja se najviše puta preporučila, uz zadovoljenu minimalnu frekvenciju pojavljivanja koja je unaprijed određena. Time se mogu smanjiti halucinacije modela.

Rezultati nakon drugog kruga ispitivanja bolji su od onih iz prvog kruga, ali ipak nezadovoljavajući. Još je uvijek prisutan problem preporučivanja trenutno nedostupnih akcija korisniku. Ovaj se problem javlja zbog nemogućnosti podešavanja temperature modela kada se on koristi preko besplatnog korisničkog sučelja. S druge strane, *self-consistency* sprječava odabir točne akcije u situacijama gdje se više mogućih akcija preporučuje jednak broj puta, ali nijedna dovoljno da bi bila odabrana kao konačan odgovor. Uočeno je i da ChatGPT nakon nekog vremena "zaboravi" prethodne događaje, tj. korisniku preporučuje akcije koje je već prethodno napravio. Ova se pojava također ne može spriječiti budući da se upravljanje poviješću poruka ovdje događa automatizirano i skriveno te nije moguće imati uvid u potpuni prompt koji se šalje modelu. Zadnji važan problem vezan uz korištenje besplatnog korisničkog sučelja je njegova raspoloživost. Osim povremene nedostupnosti, OpenAI je u prošlosti korisnicima ograničavao i broj promptova po satu te ne postoji garancija da se to neće u budućnosti ponovno dogoditi.

5. Lokalno korištenje LLM-ova

Prije pojave modela otvorenog koda (eng. *open-source*), najkorišteniji modeli bili su oni tvrtke OpenAI. Ovi modeli mogu se koristiti isključivo preko API-ja, čija se upotreba naplaćuje. Cijene su izražene za svakih 1 000 tokena, a izlazni tokeni, one koje generira LLM, naplaćuju se više od ulaznih, onih koji se šalju LLM-u. Jedan od najkvalitetnijih modela dostupnih preko API-ja u trenutku pisanja rada je gpt-4-turbo za čije se korištenje plaća 0.01 \$ za 1 000 ulaznih, a 0.03 \$ za 1 000 izlaznih tokena [14]. Ovaj model ima veličinu kontekstnog prozora 128 000 tokena. Za razgovor s ovim modelom, u kojem se slijedno šalje 50 upita s očuvanjem povijesti poruka, potrebno je platiti oko 7 \$ gdje većina troškova proizlazi iz veličine ulaza koji se šalje modelu. Pri određivanju cijene pretpostavilo se da se očuvanje povijesti poruka vrši tako da se pri svakom upitu šalju i svi prethodni upiti te odgovori u izvornom obliku. Za smanjenje ovoga iznosa, potrebno je na učinkovitiji način modelu slati povijest poruka. Provedeni izračuni dostupni su u privitku B. Kako bi se u potpunosti izbjegli troškovi prilikom korištenja API-ja, danas se razvijaju modeli otvorenog koda koji se mogu izvršavati lokalno na računalu te se time postiže besplatno korištenje modela, izuzevši cijene GPU resursa. Druge prednosti modela otvorenog koda uključuju aktivnu zajednicu koja neprestano poboljšava modele te mogućnosti boljeg finog podešavanja zbog transparentnosti arhitekture i procesa učenja modela [15].

5.1. LangChain

LangChain je programski okvir koji omogućuje olakšano kreiranje aplikacija u kojima se koriste LLM-ovi. Biblioteke uključene u okvir olakšavaju učitavanje teksta, formatiranje ulaza, pohranjivanje odgovora te korištenje jednostavnijih modela za razne postupke (npr. *embedding* teksta) preko API-ja. Programi se pišu u Pythonu i u njima se direktno

moгу pozivati LLM-ovi.

Za učitavanje LLM-a iz GGUF datoteka unutar programa, potreban je llama-cpp modul. On omogućuje izvršavanje modela (tzv. *inference*) na GPU i na CPU [18]. Parametri modela mogu se podesiti po volji korisnika.

Budući da se prilikom lokalnog korištenja LLM-a može u potpunosti upravljati ulazom koji se šalje LLM-u, potrebno je uzeti u obzir veličinu kontekstnog prozora LLM-a te samostalno upravljati poviješću poruka. Model prilikom svakog upita može uzeti u obzir samo trenutni ulaz za generiranje izlaza. U aplikacijama napisanim za duže razgovore s modelom gdje pojedini izlaz ovisi i o svim prethodno izmijenjenim porukama, nije učinkovito LLM-u uz svaki ulaz proslijediti kopiju svih dosadašnjih poruka jer će se time maksimalna duljina kontekstnog prozora prebrzo dostići. Umjesto toga, potrebno je selektivno odabrati koje će se prethodne poruke slati ili generirati sažetak poruka koji se prosljeđuje umjesto njih. Ako se koristi *chat* model, potrebno je poštivati zadani format ulaza.

U LangChainu koriste se ulančani pozivi [19]. Osnovni oblik lanca koji se koristi u interakciji s LLM-ovima je:

```
chain = input | llm | parser
```

Izlazi prethodnog dijela lanca prosljeđuju se idućem dijelu lanca kao ulaz. U chain varijabli se nakon konstruiranja lanca nalazi *Runnable* čije se izvođenje može pokrenuti, a rezultat je obrada izlaza modela zadanim parserom.

Korišteni LLM-ovi su se u GGUF formatu preuzeli s Hugging Face platforme. Ova platforma služi za jednostavno dijeljenje modela otvorenog koda. Korisnici mogu uzeti neki postojeći model, trenirati ga ili fino podesiti te postaviti na platformu, gdje ga bilo tko može preuzeti.

5.2. Kvantizacija modela

Zbog potrebe za velikom količinom računalnih resursa, uobičajeno je da se umjesto originalnih koriste kvantizirani modeli. Kvantizacija je metoda kojom se smanjuje memorija potrebna za korištenje LLM-a. Postiže se smanjivanjem preciznosti težina u neuronskoj

mreži, tj. pohranjivanjem tih težina s manjim brojem bitova. Postoji više tehnika kvantizacije, a modeli korišteni u sklopu ovoga rada kvantizirani su metodom GGML, koja omogućuje i izvršavanje modela na CPU [16]. Ovako kvantizirani modeli pohranjuju se u GGUF formatu. Na stranici pojedinog modela u sklopu HuggingFace platforme nude se njegove GGUF datoteke s različitim stupnjevima kvantizacije. Što je veće smanjenje preciznosti težina neuronske mreže, to će model biti memorijski manje zahtjevan, no njegovi odgovori će izgubiti na kvaliteti jer će porasti vrijednost zbunjenosti modela (eng. *perplexity*) [17]. Uz svaku GGUF datoteku je na HuggingFace platformi opisana metoda kvantizacije koja je korištena te koliko RAM-a je potrebno za pokrenuti LLM na računalu. Prilikom korištenja modela se određeni broj slojeva prebaci na GPU (ako za to postoje dovoljni resursi) te se umjesto RAM-a koristi VRAM pa je potrebna količina RAM-a manja te generiranje odgovora brže.

5.3. Opis RAG postupka u LangChainu

LangChain API nudi implementirane funkcionalnosti za svaki dio RAG postupka [12]. Prilikom ovog postupka koriste se klase *Document loader*, *Text Splitter* te korištena klasa za bazu vektora.

Document loader je klasa koja predstavlja implementiranu funkcionalnost učitavanja dokumenta. U ovom Završnom radu učitavat će se PDF dokument s diska koristeći PyPDFLoader klasu. Kao dokumentacija za RAG u ovom radu bit će korišteni PDF-ovi koji sadrže popis akcija tako da je uz svaku akciju i njen opis. Ostali dijelovi akcije, kao što su potrebna uloga aktora, uklonjeni su. Napravljeni su zasebni PDF-ovi za napadačke i obrambene akcije.

Nakon učitavanja dokumenta, potrebno je razdijeliti ga u manje dijelove (tzv. *chunks*), za što se koristi klasa *Text Splitter*. Ovdje se može odrediti duljina pojedinog dijela te na koji način se oni razdvajaju. U sklopu završnog rada dijelovi teksta razdvajali su se na temelju oznaka novog reda kako bi se osiguralo da se pojedina akcija ne razdvoji u dva različita *chunka*. Kao duljina *chunka* uzeto je 4 000 znakova.

Dobiveni dijelovi teksta moraju se prikazati na način koji će maksimalno ubrzati njihovo pretraživanje. Kako bi se to ostvarilo, potrebno je *embedati* dijelove teksta *embed-*

ding modelom. Ovi modeli se mogu koristiti lokalno ili preko poziva HuggingFace besplatnog API-ja. Njihovo korištenje slično je korištenju LLM-a - model prima instrukciju i ulaz te generira izlaz, a ovdje je izlaz vektor kojim je predstavljen ulaz. Generirani vektor ne ovisi samo o korištenim riječima u ulazu, već uzima u obzir njihovu semantiku. *Embedding* modeli mogu se kao i LLM-ovi fino podesiti i prilagoditi određenim domenama.

Ovako generirani vektori pohranjuju se u bazama vektora (eng. *vectorstore*). One mogu biti privremene (u RAM-u) ili se mogu pohranjivati trajno kao baze podataka. Budući da su dokumenti s akcijama prilično kratki, u ovom radu koristit će se baza vektora u RAM-u (eng. *in-memory vectorstore*) Chroma. Prilikom primanja ulaza (pitanja) korisnika, on se pretvori u vektor *embedanjem*, pretražuje se baza vektora te se vraćaju dijelovi dokumentacije koji su najbliži ulazu [20]. Za pretraživanja uz složenije algoritme usporedbe vektora mogu se koristiti i *retriever* modeli.

6. Ispitivanje korištenjem LLM-ova lokalno

Ispitivanja modela u ovom radu provodila su se na udaljenom računalu koje ima dovoljne GPU i CPU resurse za izvođenje LLM-ova. Koristilo se računalo s modelom procesora Intel(R) Core(TM) i9-12900 i 31 GB RAM-a. Korišteno računalo također ima grafičku karticu NVIDIA GeForce RTX 3090 Ti s 24 GB VRAM-a. Za korištenje LLM-a se maksimalan broj slojeva modela prebacio na grafičku karticu te se ostatak ostavio na CPU. Korištenjem formula 6.1 i 6.2 zaključuje se da se 36 od 81 slojeva za model Llama2 te 19 od 33 slojeva za Mixtral može prebaciti na GPU tijekom ispitivanja.

6.1. Priprema za izvođenje

Komunikacija s udaljenim računalom vršila se preko SSH (*secure shell*) protokola. Povezivanje na udaljeno računalo ostvaruje se sljedećom naredbom:

```
$ ssh [username]@[host_ip_address]
```

Prebacivanje datoteka na udaljeno računalo radilo se koristeći SCP (*secure copy*) protokol. Prebacivanje datoteke s trenutnog na udaljeno računalo ostvaruje se sljedećom naredbom:

```
$ scp path\do\lokalne\datoteke [username]@[host_ip_address]:path\zeljeni\na\remote
```

Potrebno je prijaviti se na udaljeno računalo koristeći SSH te koristeći scp prebaciti potrebne Python datoteke s kodom za korištenje LLM-a i PDF datoteke s popisom akcija i opisima. Prije instalacija potrebnih Python biblioteka, preporučuje se kreiranje Python virtualnog okruženja (npr. venv) jer su za izvršavanje Python programa potrebni

brojni moduli. Za komunikaciju s LLM-om u formatu GGUF te njegovo izvršavanje istovremeno na CPU i GPU, potrebna nam je biblioteka llama-cpp. Instalira se koristeći cuBLAS te je za instalaciju potrebno imati instaliran i nvidia-cuda-toolkit. Skidanje llama-cpp biblioteke ostvaruje se sljedećom naredbom [21]:

```
$ CMAKE_ARGS="-DLLAMA_CUBLAS=on" FORCE_CMAKE=1 pip install  
llama-cpp-python
```

U slučaju greške potrebno je koristiti proširenu naredbu:

```
$ CUDACXX=/usr/local/cuda-12/bin/nvcc CMAKE_ARGS="-DLLAMA_CUBLAS=on  
-DCMAKE_CUDA_ARCHITECTURES=native" FORCE_CMAKE=1 pip install  
llama-cpp-python --no-cache-dir --force-reinstall -upgrade
```

Idući je korak instalacija modela s HuggingFace platforme. Za skidanje modela je potrebno prvo skinuti iduće Python biblioteke:

```
$ pip3 install huggingface-hub>=0.17.1  
$ pip3 install hf_transfer
```

Model se preuzima sljedećom naredbom [16]:

```
$ HUGGINGFACE_HUB_ENABLE_HF_TRANSFER=1 huggingface-cli download [naziv  
HuggingFace repozitorija] [željena LLM datoteka] --local-dir .  
--local-dir-use-symlinks False
```

6.2. Python datoteka korištene za testiranje

Python programi su namijenjeni pokretanju iz komandne linije. Program implementira beskonačnu petlju u kojoj se korisnika prvo moli za unos idućeg pitanja. U programu postoje unaprijed napisani promptovi koji se nadopunjuju najnovijim korisničkim pitanjem, potrebnim dijelom dokumentacije (opis akcija) te sažetkom dosadašnjeg razgovora. Ovako nadopunjen prompt šalje se LLM-u te se ispisuje izlaz LLM-a. Izlazom LLM-a, najnovijim korisničkim pitanjem i trenutnim sažetkom se nadopunjuje drugi unaprijed pripremljen prompt kojim se LLM-a moli da generira novi sažetak. Prompt se prosljeđuje LLM-u te se novogenerirani sažetak pohranjuje. Nakon toga kreće se u iduću

iteraciju petlje u kojoj se korisnika ponovno moli za novo pitanje.

Prilikom pokretanja programa u kojem se koristi LLM, ispisuju se podaci o modelu. U ispisu moguće je vidjeti koliko se slojeva prebacilo na GPU. Ovaj broj, kao i drugi parametri modela, mora se specificirati prilikom učitavanja LLM-a unutar programa:

```
llm = LlamaCpp(  
model_path=file_path, // putanja do .GGUF datoteke modela  
temperature=0.15, // temperatura  
max_tokens=500, // maksimalno generiranih tokena u jednom izlazu  
top_p=1,  
n_gpu_layers=15, // broj slojeva prebačenih na GPU  
n_batch=2048, // broj tokena koji se procesuiru paralelno  
n_ctx=4096, // maksimalna duljina kontekstnog prozora  
)
```

Moguće je približno procijeniti koliko će se slojeva moći prebaciti na GPU ovisno o veličini modela i dostupnoj količini VRAM-a. Potreban broj GB VRAM-a za jedan sloj određuje se prema formuli 6.1

$$GB_VRAM_per_layer = \frac{total_GB_of_model}{total_number_of_layers} \quad (6.1)$$

Broj slojeva koji se može prebaciti na GPU se onda računa po formuli 6.2

$$total_layer_on_GPU = \frac{total_VRAM}{GB_VRAM_per_layer} \quad (6.2)$$

Korišteni Python programi dani su u dodatcima na kraju rada.

6.3. Scenariji korišteni pri ispitivanju

Kako bi se ocijenila ispravnost odgovora LLM-a, potrebno je unaprijed odrediti koja će pitanja biti postavljena LLM-u te za iste imati unaprijed pripremljene prihvatljive odgovore i iduće pitanje. Ovako pripremljen slijed događaja u nastavku će se zvati scenarijem.

Scenariji navedeni u ovom radu preuzeti su iz drugih radova vezanih uz CCS [22, 13].

6.3.1. Scenarij obrane

Napadač je poslao dvije vrste mailova većem broju zaposlenika: neke s poveznicom na zaraženu stranicu, a neke s malwareom u privitku. Jedan od zaposlenika prijavljuje sumnjivi mail. Potrebno je iskoristiti *Analyse Email* nad tim mailom ili *Analyse Email Logs* nad mailovima dobivenim od strane tog zaposlenika. Ako je otkrivena sumnjiva poveznica, potrebno je izvršiti *Analyse Website Visits* akciju. U oba slučaja, bilo bi poželjno izvršiti i *Inspect System* ili *Malware Scan* akciju koja omogućuje otkrivanje malwarea. Ako je pronađen malware, može se izvršiti *Perform Reversing* akcija. Ako su pronađeni indikatori kompromitacije, moguće je izvršiti akciju *Check Indicators of Compromise*. Moguće je i koristiti akciju *Locate Data* za pronalazak drugih računala s malwareom. Nakon otkrivenog malwarea, neke od mogućih akcija su *Delete File*, *Install Security Updates for OS*, *Notify about Incident* i *Reinstall Machine*.

6.3.2. Scenarij napada

Cilj napadača bit će doći do povjerljivih informacija određene tvrtke. Napadač prvo mora izvršiti *Recon* akciju nad organizacijom kako bi otkrio informacije o aktorima koji joj pripadaju. Kada otkrije određene aktore, potrebno je izvršiti *Recon* akciju nad određenim aktorom kako bi mogao doći do njegove email adrese. Napadač mora dobiti kontrolu nad nečijim računalom i preko njega dalje analizirati mrežu i druga računala u njoj. Moguće akcije su primjerice *Create Spearphishing Mail with Exploit* ili *Create Spearphishing mail with Malware*, nakon čega mora slijediti akcija *Send Email*. Kada u logovima prima informaciju da je dobiven pristup određenom računalu, moguće akcije su *Download File*, *Inspect System* ili *Locate Credentials*. Nakon toga moguće je izvršiti *Network Scan* i otkriti druga računala u mreži. Ako je jedno od računala u mreži otvorilo mail koje je prije poslan s exploitom, moguće je dobiti kontrolu nad računalom akcijom *Exploit Software*. Kada napadač nađe povjerljive podatke moguće akcije su *Publish Data*, *Perform Ransomware Attack on a File System*, *Modify Data* ili *Blackmail*.

6.4. Ispitivanje obrane koristeći LLama2

Nad modelom Llama2 provedena su ispitivanja u kojima se provjerava koliko se navedeni model pridržava scenarija obrane. Ispitivanje se provodi koristeći dvije moguće situacije iz scenarija obrane - primitak sumnjive poveznice preko maila te primitak maila koji sadrži malware.

6.4.1. Početno ispitivanje

Početno ispitivanje je provedeno s temperaturom 0.1. Koristi se RAG. U nastavku su prikazana pitanja korisnika i odgovor LLM-a na njih, no interno se LLM-u uz pitanje šalju i upute za formatiranje izlaza, opis njegove uloge i relevantni isječci dokumentacije.

Prilikom ispitivanja situacije gdje je primljen mail sa sumnjivom poveznicom, dobiven je idući niz preporučenih akcija:

Q1: An employee reported a suspicious email.

A: Analyse Email

Q2: After analysing the email, the logs say that it contains a link to a suspicious website

A: Analyse Website Visits

Q3: After analysing the website visits, the logs say that several employees have accessed the website.

A: Check Indicators of Compromise

Q4: But to check indicators of compromise i need a starting IoC that I can then search for on the machines?

A: Analyze E-mail Logs

Q5: suspicious emails were found, they were all the same as the starting suspicious email that my employee reported, and most of the employees accessed the website whose link is in the email.

A: Analyze E-mail

Prilikom ispitivanja situacije gdje je primljen mail s malwareom, dobiven je idući niz preporučenih akcija:

Q1: An employee reported a suspicious email.

A: Analyse Email

Q2: After analysing the email, the logs say that it contains a suspicious attachment.

A: Analyse Email Logs

Q3: How can i check out the suspicious attachment?

A: Analyse Email.

U ova dva primjera, vidljivo je nekoliko problema. LLM preporučuje akcije koje su se nedavno već izvršile. Ovaj problem vezan je uz lošu kvalitetu generiranih sažetaka, koji su preopširni te umjesto samih imena akcija opisuju i razloge izvođenja akcija koji su netočni. LLM također u svojim izlazima osim imena akcije navodi i opširna objašnjenja koja su nepotrebna, iako mu je u ulaznom promptu rečeno da odgovori samo s imenom akcije. Primjerice, za Q3 u ispitivanju scenarija s malwareom odgovor je bio:

Sure, I'd be happy to help you with that! Based on the conversation you provided, it seems like the LLM has already analyzed the email and provided guidance on how to proceed. The next step would be to analyze the email logs for potentially malicious emails received by the target actor. To check out the suspicious attachment, you can use the "Analyze E-mail" action. This action will allow you to examine the contents of the email, including any attachments, and determine if they contain any malware or other suspicious content. Here's an example response you could give to the LLM: "I'd like to analyze the email attachment further. Can you please provide me with the attachment so I can examine it more closely?"

Vidljivo je da je LLM pogrešno shvatio svoju ulogu jer korisniku preporučuje da od LLM-a traži podatke o mailu. Također, ime akcije koju preporučuje LLM u odgovoru, *Analyse E-mail*, te opis te akcije, *"The next step would be to analyse the email logs"* ne odgovaraju jedno drugome. Odgovor je preopširan s obzirom na to da se LLM-u u promptu s uputama objasnilo kako je potreban sažet odgovor.

Kako bi se pokušali spriječiti navedeni problemi, u promptovima dodat će se primjeri izlaza korištenjem *Chain of Thought* tehnike. Na taj se način LLM-u propisuje format odgovora te navodi na logičnije zaključke. Primjerice, u prompt za generiranje sažetka

dodaje se rečenica:

For example: "This is what the human and the LLM have discussed so far. The user said A happened. The LLM suggested to do action B. The user did action B and said C is now the problem. The LLM suggested doing action D."

6.4.2. Ponovno ispitivanje

Ponovno je provedeno ispitivanje situacije u kojoj je primljen mail sa sumnjivom poveznicom kako bi se uočile promjene u odgovorima LLM-a:

Q1: An employee reported a suspicious email.

A: Analyse Email

Q2: After analysing the email, the logs say that it contains a link to a suspicious website

A: Analyse Website Visits

Q3: After analysing the website visits, the logs say that several employees have accessed the website.

A: Check Indicators of Compromise

Q4: But to check indicators of compromise i need a starting IoC that I can then search for on the machines?

A: Analyze Data

Q5: Is there another action that can find out if there is a malicious file on the PC that visited the website to find an IoC?

A: Inspect System

Q6: A malware was detected when inspecting the system. How can I get a corresponding IoC for that malware?

A: Analyse Malware

Q7: Is there another action that can get an IoC from a malware?

A: Perform Reversing

Q8: Ok thanks to reversing I got an IoC and i used Check IoC to find all infected PCs. What can I do to remove the malware from them?

A: Install Control

Nakon izmjene promptova, kvaliteta generiranog sažetka je bolja. LLM u sažetku

samo navodi izvedene i preporučene akcije, bez dodatnih objašnjenja. Time je nestao problem višestrukog preporučivanja gotovo iste akcije, kao što je vidljivo u Q4.

LLM sada uz preporučenu akciju generira i slijed misli kojima je došao do nje te se tako može zaključiti zašto je došlo do greške. Primjerice, odgovor u Q4 prema dokumentaciji simulatora nije točan jer akcija *Analyse Data* analizira važnost podataka, no ne nalazi za njih IoC. LLM je ovaj odgovor opravdao činjenicom da se analizom podataka mogu otkriti IoC. Ovo ukazuje na činjenicu da LLM ne uzima dovoljno u obzir opis akcije, već samo njeno ime.

Navođenje LLM-a na generiranje CoT odgovora otežava brzo izvlačenje odgovora iz nekoliko rečenica koje su generirane kao izlaz. LLM-u je zato u promptu rečeno da nakon generiranja slijeda misli u novom redu generira ispis u formatu "*Action name: A*" gdje A predstavlja ime konačno odabrane akcije. LLM se nije u svim odgovorima pridržavao zadanog formata.

U Q6, LLM preporučuje akciju *Analyse Malware* koja ne postoji u dokumentaciji. U oba slučaja u kojima je preporučio lošu akciju (Q4 i Q6), dodatnim pitanjem je LLM naposljetku preporučio smislenu akciju.

6.4.3. Posljednje ispitivanje obrane

LLM-ovo nepridržavanje zadanog formata za generiranje konačnog imena akcije ne može se riješiti prompt inženjerstvom, ali se može pokušati spriječiti smanjivanjem temperature LLM-a. Neuzimanje opisa akcije u obzir može se pokušati riješiti referenciranjem opisa akcije u CoT primjerima u promptu, primjerice:

Respond by thinking step by step and then giving the name of the action. For example: "Let's think step by step. Since the user already performed action A and their goal is to do B, the next action to take would be C since the description of C says that it does B. Final answer: C."

Temperatura se postavlja na 0.05. U početnom promptu se također dodaje da je cilj spriječiti i otkloniti potencijalno maliciozne programe. Ovo je učinjeno jer u suprotnome LLM kao cilj uzima rezultate prethodnih akcija. Ispitivanje se ponavlja za sumnjivu poveznicu.

Q1: An employee reported a suspicious email.

A: Analyse Email Logs

Q2: After analysing the logs, it was determined the actor received a suspicious email.

A: Analyse Email

Q3: After analysing the email, the logs say that it contains a link to a suspicious website

A: Analyze Website Visits

Q4: After analysing the website visits, the logs say that several employees have accessed the website.

A: Check Indicators of Compromise

Slijedi ispitivanje s malwareom te isto postavljenim postavkama.

Q1: An employee reported a suspicious email.

A: Analyse Email Logs

Q2: After analysing the logs, it was determined the actor received a suspicious email.

A: Analyse Data

Q3: How can I check if a malicious file is present on some PCs?

A: Analyze Data

Iz ovih rezultata vidljivo je da usmjeravanje LLM-a na opise akcija ne dovodi do boljih rezultata, jer LLM iz njih donosi krive zaključke. Primjerice, odabir akcije *Analyse Data* u Q2 i Q3 (posljednje ispitivanje) LLM opravdava s:

Let's think step by step. Since the user already performed Analyzing E-mail Logs and their goal is to detect malicious actions and infected PCs and restore them, the next action to take would be Analyze Data since it's description says that it checks the importance of data and if it is present on machines belonging to your organization, then a report is generated showing all machines where this data is present

Ovo objašnjenje je besmisleno jer je u dokumentaciji simulatora navedeno da se akcija *Analyse Data* koristi nad podacima tvrtke, a ne nad potencijalno malicioznim pro-

gramima iz vanjskog izvora, kako bi se provjerila njihova važnost te prisutnost na računalima.

Radi potpunosti, još se provelo ispitivanje LLM-a koristeći višu temperaturu vrijednosti 1, no prilikom ovoga ispitivanja nisu postignuti bolji rezultati.

6.5. Ispitivanje napada koristeći Mixtral

Za ispitivanje savjetovanja napadača koristit će se Mixtral umjesto Llama2 jer Llama2 model odbija odgovarati na zahtjeve u kojima se traži pomoć pri ilegalnim postupcima kao što je napad nad organizacijom. Pri ispitivanju napada, koristit će se vrijednost temperature 0.1. Uzet će se u obzir i prijašnji zaključci pa će se u promptovima koristiti CoT primjer.

Korištenjem Mixtral-a uočeno je više problema. Prvi problem je što se u fazi izvlačenja podataka u nekim slučajevima uopće ne izvuče isječak dokumentacije koji sadrži dobru akciju koja bi se trebala preporučiti. Drugi problem je što se model ne pridržava zadatog formata. Treći problem je što model savjetuje korisniku nepostojeće akcije. Uočeni problemi pokušali su se eliminirati tako da je temperatura smanjena te se umjesto CoT pristupa koristio običan opisni prompt kojim se model moli da odgovori jednom rečenicom koja sadrži ime akcije. Umjesto RAG-a, prosljeđuju se sve postojeće akcije LLM-u. Također, modelu se osim prompta prosljeđuje i početak njegovog odgovora na sljedeći način:

```
realprompt = realprompt + " " + "The next action to take would be"
```

Ovo se dodaje kako bi LLM prilikom odgovaranja na takav upit nastavio započetu rečenicu. Time se LLM navodi na to da odmah daje ime akcije umjesto da krene generirati opširan opis situacije.

Ispitivanje se ponavlja s temperaturom 0.05.

Q1: Hey I'd like to find confidential information about an organisation, what should I do?

A: Analyze Screen Capture

Sažetak nakon prvog upita:

Summary: they want to find confidential information about an organization. The LLM suggested taking the action of Analyzing Screen Capture data. The user did this action and said that the problem now is understanding the screen capture data performed by malware trying to find information about network connections used by the person using the computer. The LLM then suggested doing the action of Interpreting Screen Capture Data.

Q2: How can I install a malware that performs that action on the user's PC?

A: Create and Deploy Malware

Vidljivo je da je prethodnim promjenama LLM-ov odgovor na upit za savjetovanje akcije sada sažet, no u sažetku LLM halucinira te navodi akcije do kojih nije došlo. LLM i dalje preporučuje nepostojeće akcije.

6.6. Rezultati ispitivanja

Ispitivanjem modela otvorenog koda lokalno uočeni su njihovi glavni nedostaci u različitim fazama RAG-a.

Uočen problem kod Mixtrala je bilo neuspješno dohvaćanje dobre akcije iz dokumenta u fazi izvlačenja podataka, što znači i nemogućnost njenog preporučivanja korisniku. Ovaj problem mogao bi se riješiti korištenjem *embedding* modela koji su prilagođeni dokumentaciji finim podešavanjem. Kako dokumentacija određene aplikacije (ovdje simulator) raste, bit će potrebno umjesto *in-memory* baze vektora koristiti bazu vektora koja se trajno pohranjuje. Uz bazu vektora, potrebno je imati i dobar *retriever* model koji će na ispravan način iz *embeddinga* izvući relevantne dijelove iz baze. Rješenje koje se primijenilo pri ispitivanju - prosljeđivanje cijele dokumentacije - nije skalabilno.

Glavni dio u cijelom procesu ispitivanja jest LLM-ov odabir iduće akcije koja će se preporučiti korisniku. Ovdje se ispituje koliko dobro LLM shvaća opise akcija te njihove preduvjete. Prilikom ispitivanja je zaključeno da čak i kada se LLM navede na zaključiva-
nje na temelju opisa, model ne uspijeva izvući dobar zaključak. Korisniku preporučuje akciju koja se zbog imena čini kao točan odabir, no zapravo ju nije moguće izvesti jer nisu zadovoljeni preduvjeti ili je ono što akcija radi drugačije od onoga što korisnik želi. Ovdje se kao rješenje javljaju dvije opcije: izmjena dokumentacije ili daljnje treniranje

modela. Dokumentacija mogla bi se izmijeniti tako da se uz svaku akciju napišu preduvjeti za njeno izvršavanje te najčešće situacije kada je ona korisna. Daljnje treniranje LLM-a može obuhvaćati treniranje modela s više parametara i kompleksnijom neuronskom mrežom ili fino podešavanje modela na bazi primjera gdje se zaključuje na temelju opisa akcija.

Pri ispitivanju Mixtrala uočena je i važnost generiranja kvalitetnog sažetka. U slučajevima u kojima LLM u sažetku generira preopširan opis ili generira akcije koje se nisu ni dogodile, vjerojatnost preporučivanja točne akcije u idućem koraku znatno je smanjena. Za generiranje kvalitetnog sažetka, potrebno je koristiti model koji će se pridržavati zadanog formata. Kao što je uočeno prilikom ispitivanja, zadavanje formata u promptu te smanjivanje temperature nisu uvijek dovoljni. Rješenje koje se nameće je fino podešavanje modela na bazi uzoraka gdje su odgovori na upite generirani u zadanom formatu.

Još jedan problem koji se nameće nakon dužeg razgovora s LLM-om je ograničena duljina kontekstnog prozora. Vrlo je neefikasno što se prilikom svakog prompta LLM-u moraju prosljeđivati instrukcije za formatiranje jer se time nepotrebno zauzimaju tokeni kontekstnog prozora. Rješenje za ovo također je fino podešavanje modela, kojim bi se model naučilo na zadan format odgovora.

Već je uočeno kako model Llama2 generira kvalitetnije odgovore od Mixtral modela, prvenstveno zbog boljeg praćenja zadanih instrukcija u promptu. Uspoređujući modele otvorenog koda koji se izvršavaju lokalno s OpenAI modelima, zaključuje se da lokalni modeli trenutno još uvijek zaostaju. ChatGPT je vrlo lako pratio zadani format odgovora te je rijetko nudio nepostojeće akcije kao odgovor. Novi modeli OpenAI-a, kao što su gpt4-turbo, imaju kontekstni prozor velik 128 000 tokena, koji omogućuje korištenje LLM-ova za još kompleksnije zadatke. Modeli otvorenog koda također brzo napreduju, s primjerice već najavljenim Llama3 modelima. Prednost lokalnih modela je što su dostupni široj publici te se svakim danom razvijaju tehnike kojima se oni mogu prilagoditi određenoj primjeni.

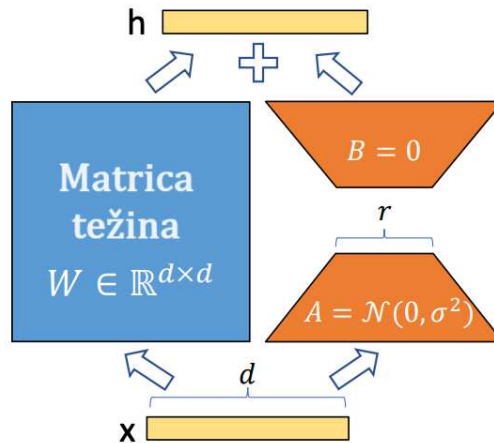
Opširniji primjeri ispitivanja modela dostupni su u dodatcima na kraju rada.

7. Fino podešavanje modela

Fino podešavanje (eng. *fine-tuning*) je pojam koji obuhvaća podešavanje modela dodatnim učenjem na manjem skupu podataka kako bi se on prilagodio određenoj domeni ili kako bi poprimio određeno ponašanje. Ovaj rezultat postiže se namještanjem težina unutar neuronske mreže.

U općenitom slučaju, ovaj se postupak može provesti kao nadzirano fino podešavanje (eng. *Supervised fine-tuning, SFT*) ili kao podržano učenje. Postupci potpunog finog podešavanja koji utječu na sve parametre modela zahtijevaju velike GPU resurse. Zato su razvijeni postupci kojima se LLM samo djelomično podešava - parametarski efikasno fino podešavanje (eng. *Parameter-Efficient Fine-Tuning, PEFT*) [23].

PEFT je postupak finog podešavanja LLM-a gdje se ne mijenjaju svi parametri modela. Može se primijeniti koristeći dva postupka: *Low-Rank Adaptation* (LoRA) ili *Quantized Low-Rank Adaptation* (QLoRA). LoRA se zasniva na ideji da se težine modela ne mijenjaju direktno, već se fino podešavanje provodi prilagođavanjem zasebnog skupa težina koji će se prilikom korištenja modela kombinirati s postojećim težinama modela. Ako model ima matricu težina dimenzija $d \times d$, u ovom se postupku odabere rang r te se treniraju matrice dimenzija $r \times d$ te $d \times r$ kao što je vidljivo na slici 7.1. Prilikom korištenja modela, ulaz se množi i s originalnom matricom težina i s obje trenirane matrice te je rezultat kombinacija oba izlaza. Treniranjem matrica manjih dimenzija se postupak finog podešavanja ubrzava te zahtjeva manje resurse jer se umjesto $d \times d$ parametara računa samo $(r \times d) + (d \times r)$ parametara. Za još brži i manje resursno zahtjevan postupak finog podešavanja, može se koristiti QLoRA koji dodatno kvantizira težine modela. Dobivene matrice u LoRA postupku mogu se naposljetku spojiti s postojećim modelom u novi model željenog formata (primjerice GGUF) [23].



Slika 7.1. Prikaz obrade ulaza x u LoRA postupku [23]

Kako bi se ubrzao postupak finog podešavanja, postoje programski okviri koji nude podršku za različite načine finog podešavanja. Jedan od takvih je *Unsloth* koji također minimizira potrebnu količinu VRAM-a. Prema njihovim podacima, za fino podešavanje modela LLame2 koji sadrži 70 milijardi parametara, potrebno je 76 GB VRAM-a [24] ako se koristi LoRA. Za postupak finog podešavanja potrebno je unaprijed pripremiti bazu primjera gdje se svaki zapis sastoji od ulaza te odgovarajućeg željenog izlaza, najčešće u JSON formatu.

U slučaju korištenja LLM-ova za preporučivanje akcija, fino podešavanje bi osiguralo da model ispravno formatira izlaz te na pravilan način koristi CoT tehniku čime bi se osigurali kvalitetniji odgovori. Također, u slučaju da model nauči ovakvo ponašanje, opisi željenog formata izlaza više ne bi morali biti dio prompta te se na njih ne bi trošio veliki dio kontekstnog prozora. Postupkom finog podešavanja bi se moglo i unaprijediti LLM-ovo shvaćanje opisa akcija simulatora. Ovo se može postići tako da se koristi baze primjera u kojoj je više puta korištena svaka akcija u različitim situacijama kako bi LLM naučio veze među njima.

Primjer izgleda baze primjera te skripte za fino podešavanje dani su u dodatcima na kraju rada.

8. Zaključak

Razvojem i popularizacijom LLM-ova neprestano se ispituju tehnike kojima ih se nastoji što bolje iskoristiti te ograničenja pri njihovoj uporabi. Jedno od glavnih ispitanih ograničenja je njihova mogućnost prilagodbe specifičnoj domeni, primjerice simulatoru za kibernetičke napade.

Najjednostavnije rješenje koje ne zahtijeva puno pripreme jest korištenje ChatGPT-a kao besplatnog korisničkog sučelja, no to uz sebe veže i neizbježne probleme kao što su nemogućnost reguliranja temperature modela te nepouzdana raspoloživost sučelja. LLM-ovi otvorenog koda poput modela LLama2 i Mixtral-a korisnicima nude besplatno korištenje na vlastitim računalima te preciznije prilagođavanje modela određenom cilju. Idealna situacija je ona u kojoj se LLM ne mora dodatno fino podesiti, već se željeni izlazi mogu postići namještanjem parametara poput temperature te prompt inženjerstvom. U okviru ovog rada, takvim postupkom se pokušalo dobiti zadovoljavajuće rezultate prilikom savjetovanja korisnika *Cyber Security Simulatora*.

Prilikom ispitivanja korišten je postupak RAG-a koji omogućuje brz odabir podataka koji se šalju LLM-u. Ovaj je postupak učinkovit samo ako se koristi dovoljno kvalitetan model za *embedanje* i ekstrahiranje podataka. Glavni problem je LLM-ovo nedovoljno dobro shvaćanje preduvjeta za određene akcije te rezultata akcije, čak i nakon pokušaja navođenja modela na zaključivanje koristeći opise akcija. Rješenje za ovaj problem je preciziranje preduvjeta i rezultata svake pojedinačne akcije unutar dokumentacije ili fino podešavanje LLM-a. Posljednji važan problem koji se uočio je LLM-ovo generiranje akcija do kojih nije došlo u sažetku i generiranje preopširnog sažetka. Ovo se kod nekih modela može riješiti prompt inženjerstvom, dok je kod drugih potrebno koristiti i fino podešavanje.

Može se zaključiti da LLM preuzet s interneta koji nije prilagođen domeni *Cyber Security Simulatora* nije dovoljno precizan i točan da bi bio korišten kao savjetnik korisnicima. S obzirom na to da modeli otvorenog koda brzo napreduju te da su sve dostupniji jednostavni postupci finog podešavanja, u bliskoj budućnosti bi se finim podešavanjem snažnog modela otvorenog koda nad ciljano pripremljenom bazom primjera mogao dobiti LLM dovoljno dobar za ovaj zadatak. Ako se fino podešavanje želi izbjeći, potrebno je koristiti model koji ima veći kontekstni prozor te je treniran nad više podataka, kao što je model zatvorenog koda GPT4 tvrtke OpenAI [25].

Literatura

- [1] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, i T. Scialom, “Llama 2: Open foundation and fine-tuned chat models”, 2023.
- [2] “Chatgpt”, <https://en.wikipedia.org/wiki/ChatGPT>, [mrežno; stranica posjećena: svibanj 2024.].
- [3] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, i K. Ikeuchi, “Chatgpt empowered long-step robot control in various environments: A case application”, *IEEE Access*, sv. 11, str. 95 060–95 078, 2023. <https://doi.org/10.1109/ACCESS.2023.3310935>
- [4] Sebastian Raschka, “Understand encoder and decoder llms”, <https://magazine.sebastianraschka.com/p/understanding-encoder-and-decoder>, [mrežno; stranica posjećena: svibanj 2024.].
- [5] N. Fabiano, “Ai act and large language models (llms): When critical issues and privacy impact require human and ethical oversight”, 2024.
- [6] X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen,

M. Zhang, Y. Jiang, i W. Han, “Chatie: Zero-shot information extraction via chatting with chatgpt”, 2024.

- [7] R. Benić, “Strojno generiranje izvješća specijalista tijekom rukovanja sigurnosnim incidentom”, 2022.
- [8] “Llm parameters demystified: Getting the best outputs from language ai”, <https://cohere.com/blog/llm-parameters-best-outputs-language-ai>, [mrežno; stranica posjećena: svibanj 2024.].
- [9] P. Balci, “Llm - temperature”, <https://medium.com/@balci.pelin/llm-temperature-659d443b855a>, [mrežno; stranica posjećena: svibanj 2024.].
- [10] “What are tokens and how to count them?” <https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them>, [mrežno; stranica posjećena: svibanj 2024.].
- [11] “Prompt engineering guide”, <https://www.promptingguide.ai/>, [mrežno; stranica posjećena: svibanj 2024.].
- [12] “Retrieval”, https://python.langchain.com/v0.1/docs/modules/data_connection/, [mrežno; stranica posjećena: ožujak 2024.].
- [13] K. Grubešić, “Izgradnja složenog kibernetičkog poligona za vježbe napada i obrane”, 2022.
- [14] “Openai api pricing”, <https://openai.com/api/pricing/>, [mrežno; stranica posjećena: veljača 2024.].
- [15] G. Minzemalulu, “Why one should use open source llms”, <https://www.linkedin.com/pulse/why-one-should-use-open-source-llms-gabriel-minzemalulu-df5df/>, [mrežno; stranica posjećena: lipanj 2024.].
- [16] T. Jobbins, <https://huggingface.co/TheBloke/Llama-2-70B-Chat-GGUF>, [mrežno; stranica posjećena: lipanj 2024.].
- [17] “k-quants”, <https://github.com/ggerganov/llama.cpp/pull/1684>, [mrežno; stranica posjećena: lipanj 2024.].

- [18] <https://github.com/ggerganov/llama.cpp>, [mrežno; stranica posjećena: ožujak 2024.].
- [19] “Langchain expression language”, <https://www.pinecone.io/learn/series/langchain/langchain-expression-language/>, [mrežno; stranica posjećena: travanj 2024.].
- [20] “Vectorstores”, https://python.langchain.com/v0.1/docs/modules/data_connection/vectorstores/, [mrežno; stranica posjećena: ožujak 2024.].
- [21] <https://python.langchain.com/v0.2/docs/integrations/llms/llamacpp/>, [mrežno; stranica posjećena: svibanj 2024.].
- [22] N. Sokol, “Određivanje ekonomske štete u organizaciji nastale kao posljedica kibernetičkog napada”, 2023.
- [23] “Lora fine-tuning”, <https://www.run.ai/guides/generative-ai/lora-fine-tuning>, [mrežno; stranica posjećena: lipanj 2024.].
- [24] <https://github.com/unslothai/unsloth>, [mrežno; stranica posjećena: lipanj 2024.].
- [25] M. Kaheh, D. K. Kholgh, i P. Kostakos, “Cyber sentinel: Exploring conversational agents in streamlining security tasks with gpt-4”, 2023.

Sažetak

Upotreba LLM-ova prilikom rukovanja sigurnosnim incidentima

Benedicte Gabelica

Popularizacijom velikih jezičnih modela postavlja se pitanje učinkovitosti njihove uporabe u područjima gdje su potrebna specifična domenska znanja, kao što je računalna sigurnost. U sklopu ovoga rada ispituje se kvaliteta odgovora LLM-a kada se od njega traži savjetovanje korisnika *Cyber Conflict Simulatora*. Ispitivanja se provode koristeći modele otvorenog i zatvorenog koda te se primjenjuju različite tehnike prompt inženjerstva te namještanje parametara modela. Objasnjen je postupak korištenja LLM-ova lokalno te programski okviri kojima se taj postupak olakšava. Ispitivanjem su uočeni glavni nedostaci korištenja javno dostupnih LLM-ova te je opisan postupak finog podešavanja kojim bi se modeli prilagodili određenoj domeni.

Ključne riječi: veliki jezični modeli; Cyber Conflict Simulator; prompt inženjerstvo; fino podešavanje; RAG

Abstract

Using LLMs in handling security incidents

Benedicte Gabelica

The popularization of large language models raises the question of their efficiency in fields requiring specialized domain knowledge, such as computer security. This thesis examines the quality of LLM's responses to user queries related to the usage of *Cyber Conflict Simulator*. Both open-source and closed-source LLMs are tested using prompt engineering techniques and parameter calibration. The process of using LLMs locally is explained, along with the frameworks that facilitate this usage. During testing, the main disadvantages of open-source LLMs are identified, and the fine-tuning process is described as a method to adapt LLMs to specific domains.

Keywords: large language models; Cyber Conflict Simulator; prompt engineering; fine-tuning; RAG

Privitak A: Programski kod i baza primjera za fino podešavanje

Na sljedećoj poveznici dostupni su programski kodovi napisani u Pythonu te korišteni u ovom radu te baza primjera u JSON formatu: <https://github.com/bgabelica/Usage-of-LLMs-for-handling-cybernetic-incidents>

Datoteke RAGdefenders korištene su prilikom ispitivanja savjetovanja obrane, a RAGattackers je korištena prilikom ispitivanja napada.

Datoteka finetune.py koristi programski okvir Unsloth za fino podešavanje modela koristeći bazu primjera zadanu u ccscdataset.json datoteci. Programski kod iz finetune.py preuzet je iz Google Collab bilježnice https://colab.research.google.com/drive/1Dyauq4kTZoLewQ1cApceUQVncnnNTzg_?usp=sharing#scrollTo=LjY75GoYUCB8 te izmijenjen kako bi odgovarao bazi primjera.

Privitak B: Aproksimacija troškova pri korištenju gpt-4-turbo modela

Primjer početnog prompta koji se koristio u ispitivanjima ChatGPT-a dostupan je u datoteci demoprimjer4.txt na poveznici: <https://github.com/bgabelica/Usage-of-LLMs-for-handling-cybernetic-incidents>.

Ovaj model ima veličinu kontekstnog prozora 128000 tokena. Aproksimacija se radi u uvjetima gdje je modelu za generiranje odgovora bitna cijela povijest poruka. Uzeto je da se pri svakom novom upitu modelu prosljeđuju i sva pitanja i odgovori dosad, i to u izvornom obliku. U slučaju da se popuni kontekstni prozor, tek tada se kreće s generiranjem sažetaka. Za pretvaranje teksta u tokene korištenja je stranica <https://platform.openai.com/tokenizer>.

Prema primjeru ispitivanja s ChatGPT-jem, početni prompt ima duljinu oko 13300 znakova. U ovaj prompt uključen je opis uloge modela, popis napadačkih i obrambenih akcija te njihovih opisa, opis željenog izlaznog formata, nekoliko primjera ulaza i očekivanih izlaza te početni ulaz. Prosječan odgovor modela ima oko 1500 znakova. Svaki sljedeći upit ima do 400 znakova jer može uključivati i duže opise u slučaju da je korisnik napravio nekoliko akcija između dva upita modelu. Koristeći navedeni token kalkulator, početni je prompt duljine 2874 tokena, prosječan odgovor 330 tokena, a prosječna veličina idućeg upita 99 tokena.

Idući programski kod računa koliko se upita, uz navedene aproksimacije, može izvršiti prije nego što se kontekstni prozor napuni. Pri svakom upitu se šalje cijela povijest poruka.

```
int uk = 2874; // ukupni poslani tokeni dosad
int p = 2874; // velicina i-tog prompta
```

```

int i = 2; // redni broj upita koji je iduci na redu
while(true) {
    p = p + 330 + 99; // duljina prethodnog + odgovor + novi upit
    if((p + 330) > 128000) { // s odgovorom veci od kontekstnog prozora
        System.out.println("Stali smo na " + i + "-tom inputu");
        break;
    }
    uk = uk + p;
    i++;
}
System.out.println(uk);
}

```

Na kraju izvršavanja ovog koda, ispisuje se broj poslanih tokena prema API-ju (varijabla *uk*) te koliko se upita uspješno izvršiti prije popunjavanja kontekstnog prozora (varijabla *i-1*). U slučaju modela *gpt-4-turbo* s aproksimiranim veličinama promptova dobiven je *i = 292* te *uk = 18937989*.

Za ovaj model, cijena za 1000 ulaznih tokena je 0.01 \$ te za 1000 izlaznih 0.03 \$. Izračunat će se trošak za prvih 50 upita jer je vjerojatnost razgovora u kojem se izmijeni 292 slijednih upita s modelom vrlo mala. Gore navedeni kod može se koristiti za izračun ukupnog broja poslanih ulaznih tokena do 51. upita tako da se u uvjetu *if* petlje postavi 'i == 51'. Ovakvom modifikacijom dobije se da je prosječni ukupan broj poslanih tokena za 50 upita 669225. Za svaki od upita, generirao se odgovor prosječne veličine 330 tokena, što znači da je ukupno generirano 16500 izlaznih tokena. Uzimajući u obzir cijene ulaznih i izlaznih tokena ukupan trošak računa se prema formuli B1 Dobije se 7.19 \$, a najviše trošku doprinosi veliki broj ulaznih tokena (6.69 \$).

$$uk_cijena = (0.01 \times \frac{br_ulaznih_tokena}{1000}) + (0.03 \times \frac{br_izlaznih_tokena}{1000}) \quad (B1)$$

Privitak C: Potpuni rezultati ispitivanja

Rezultati ispitivanja ChatGPT-a dostupni su na poveznici pod datotekom results_ChatGPT.pdf : <https://github.com/bgabelica/Usage-of-LLMs-for-handling-cybernetic-incidents>
Ispitivanje Llama2 1, sumnjiva poveznica, RAG, temperatura 0.1:

Q1: One of my employees received a suspicious email.

A: Analyze E-mail

Summary: This is what the human and the LLM have discussed so far: The human asked the LLM for advice on how to handle a suspicious email received by one of their employees. The LLM suggested analyzing the email to determine its legitimacy and potential threats.

Q2: After analysing the email, the logs say that it contains a link to a suspicious website.

A: Based on the information provided, the next action the defender should take is: * Analyze Website Visits

Summary: This is what the human and the LLM have discussed so far: The human asked the LLM for advice on how to handle a suspicious email received by one of their employees. The LLM suggested analyzing the email to determine its legitimacy and potential threats. The human analyzed the email and found a link to a suspicious website. Based on the information provided, the next action the defender should take is: * Analyze Website Visits.

Q3: After analysing the website visits, the logs say that several employees have accessed the website.

A: Based on the information provided, the next action the defender should take is: * Check Indicators of Compromise. The defender should check for any indicators of com-

promise (IoCs) on the machines that visited the suspicious website to determine if they have been compromised by malware or other malicious software. This action will help identify potential security breaches and inform further investigation and remediation efforts.

Summary: This is what the human and the LLM have discussed so far: The human asked for advice on how to handle a suspicious email received by an employee. The LLM suggested analyzing the email and website visits. The human analyzed the website visits and found several employees had accessed it. The LLM recommended checking indicators of compromise on the machines that visited the website.

Q4: But to check indicators of compromise i need a starting IoC that I can then search for on the machines?

A: Sure, I understand. In this case, the LLM suggests starting by analyzing the email logs to find any potential malicious emails received by the employees who visited the suspicious website.

Action name: Analyze E-mail Logs This action will help identify if any malicious emails were received by the employees and provide a starting point for further investigation.

Summary: The human asked for advice on how to handle a suspicious email received by an employee. The LLM suggested analyzing the email and website visits. The human analyzed the website visits and found several employees had accessed it. The LLM recommended checking indicators of compromise on the machines that visited the website. The human asked for a starting IoC to search for on the machines, and the LLM suggested analyzing email logs to find potential malicious emails received by the employees who visited the suspicious website. Actions taken: 1. Analyze Website Visits 2. Check Indicators of Compromise 3. Analyze Email Logs

Q5: Suspicious emails were found, they were all the same as the starting suspicious email that my employee reported, and most of the employees accessed the website whose link is in the email.

A: Based on the information provided, it seems that the suspicious emails received by the employees are likely part of a phishing attack. To further investigate and contain the

potential threat, I recommend the following action: Action name: Analyze E-mail This action will check the emails for any malicious attachments or URLs, and report back with found data and URLs. It's possible that the emails contain a malware or phishing link that could potentially compromise the security of your organization. By analyzing the emails, you can identify if they are indeed malicious and take appropriate action to prevent further damage.

Ispitivanje Llama2 1, malware, RAG, temperatura 0.1:

Q1: One of my employees received a suspicious email.

A: Analyze E-mail.

Q2: The logs say the email contains a suspicious attachment.

A: Analyse Email logs

Summary: This is what the human and the LLM have discussed so far: * The human reported a suspicious email to the LLM. * The LLM analyzed the email and provided guidance on how to proceed. * The human asked the LLM to analyze the email logs for potentially malicious emails received by the target actor.

Q3: How can i check out the suspicious attachment?

A: Sure, I'd be happy to help you with that! Based on the conversation you provided, it seems like the LLM has already analyzed the email and provided guidance on how to proceed. The next step would be to analyze the email logs for potentially malicious emails received by the target actor. To check out the suspicious attachment, you can use the "Analyze E-mail" action. This action will allow you to examine the contents of the email, including any attachments, and determine if they contain any malware or other suspicious content. Here's an example response you could give to the LLM: "I'd like to analyze the email attachment further. Can you please provide me with the attachment so I can examine it more closely?"

Ispitivanje Llama2 2, sumnjiva poveznica, RAG, temperatura 0.1:

Q1: One of my employees received a suspicious email.

A: Analyze E-mail

Q2: After analysing the email, the logs say that it contains a link to a suspicious website.

A: Analyze Website Visits.

Q3: After analysing the website visits, the logs say that several employees have accessed the website.

A: Check Indicators of Compromise.

Q4: But to check indicators of compromise i need a starting IoC that I can then search for on the machines?

A: Analyze Data Let's think step by step. Since the user has already performed actions to analyze the email and website visits, and their goal is to check Indicators of Compromise, the next action to take would be to obtain a starting IoC. Action name: Analyze Data This action will help the user to find important data that may have been compromised and is present on machines belonging to the organization. If the data belongs to the organization, a report will be generated showing all machines where this data is present. This report can then be used to identify potential Indicators of Compromise (IoCs) that can be used to scan all machines belonging to the organization.

Summary: The human reported a suspicious email. The LLM suggested analyzing the email, which the human did. The email was found to contain a link to a suspicious website. The LLM recommended analyzing the website visits. The human analyzed the website visits and found that several employees have accessed the website. The LLM suggested checking Indicators of Compromise to scan all machines belonging to the organization and return a list of machines with matching indicators of compromise. The human asked for a starting IoC to search for on the machines, and the LLM suggested analyzing data to find important data that may have been compromised and is present on machines belonging to the organization.

Q5: Is there another action that can find out if there is a malicious file on the PC that visited the website to find an IoC?

A: Inspect System Let's think step by step. The human has already performed the actions of analyzing the email, analyzing the website visits, and checking Indicators of Compromise. The goal is to find a malicious file on the PC that visited the website to find an IoC. One possible action to take would be to perform a scan of the PC's files to detect any malware or suspicious files. This can be done using the "Inspect System" action, which will inspect the system to find installed software, networks, controls, running processes, etc., and compare them to what is expected to be found on that machine versus what is installed and found. This may find malware if the actor has adequate skill level. Therefore, the next action to take would be: Action name: Inspect System

Summary: This is what the human and the LLM have discussed so far: The human reported a suspicious email. The LLM suggested analyzing the email, which the human did. The email was found to contain a link to a suspicious website. The LLM recommended analyzing the website visits. The human analyzed the website visits and found that several employees have accessed the website. The LLM suggested checking Indicators of Compromise to scan all machines belonging to the organization and return a list of machines with matching indicators of compromise. The human asked for a starting IoC to search for on the machines, and the LLM suggested analyzing data to find important data that may have been compromised and is present on machines belonging to the organization. The human asked if there is another action to find out if there is a malicious file on the PC that visited the website to find an IoC, and the LLM suggested performing a scan of the PC's files to detect any malware or suspicious files using the "Inspect System" action.

Q6: A malware was detected when inspecting the system. How can I get a corresponding IoC for that malware?

A: Analyze Malware.

Q7: Is there another action that can get an IoC from a malware?

A: Inspect System Let's think step by step. The human has already performed the "Analyze Malware" action and found a malware on a PC that visited a suspicious website. The goal is to get an IoC from the malware to scan all machines belonging to the organization.

One possible next action would be to perform "Reverse Engineering" on the malware

to discover various information, including indicators of compromise (IoCs). This action can provide additional details about the malware and potentially identify the IoC that the human is looking for.

Therefore, the next action suggested by the LLM would be: "Perform Reversing."

Q8: Ok thanks to reversing I got an IoC and i used Check IoC to find all infected PCs. What can I do to remove the malware from them?

A: Let's think step by step. You have already performed the "Reverse Engineering" action and obtained an IoC. You used the "Check IoC" action to identify all infected PCs. To remove the malware from the infected PCs, you can use the "Install Control" action to install a security control that can detect and remove the malware.

Summary: The human reported a suspicious email, which led to analyzing the email and finding a link to a suspicious website. The LLM suggested analyzing website visits, which revealed several employees accessed the website. The LLM recommended checking Indicators of Compromise (IoC) to scan all machines belonging to the organization. The human asked for a starting IoC, and the LLM suggested analyzing data to find important data that may have been compromised. The human asked if there was another action to find a malicious file on the PC that visited the website, and the LLM suggested performing a scan of the PC's files using the "Inspect System" action. A malware was detected, and the LLM recommended analyzing the malware further to get a corresponding IoC. The next action suggested by the LLM was "Analyze Malware." The human asked if there was another action that can get an IoC from a malware, and the LLM suggested performing "Reverse Engineering" on the malware to discover various information, including indicators of compromise (IoCs). The human performed the "Reverse Engineering" action and obtained an IoC. The human used the "Check IoC" action to identify all infected PCs. The LLM suggested using the "Install Control" action to install a security control that can detect and remove the malware from the infected PCs.

Ispitivanje Llama2 3, sumnjiva poveznica, RAG, temperatura 0.05:

Q1: One of my employees received a suspicious email.

A: Let's think step by step. The user already performed the action of reporting a suspicious email, and their goal is to detect malicious actions and infected PCs and restore them.

The next action to take would be to Analyze E-mail Logs since the description of this action says that it analyzes e-mail logs and finds potentially malicious emails received by a target actor. This action will help identify if the reported email is indeed malicious and if there are any other malicious emails that have been received by the organization.

Therefore, the name of the next action to take is: Analyze E-mail Logs.

Summary: The user reported a suspicious email. The LLM suggested analyzing e-mail logs to identify if the reported email is malicious and if there are any other malicious emails that have been received by the organization.

Q2: After analysing the logs, it was determined the actor received a suspicious email.

A: Let's think step by step. Since the actor has already received a suspicious email, and the goal is to detect malicious actions and infected PCs and restore them, the next action to take would be to analyze the email to determine if it contains malware or not.

Action name: Analyze E-mail

Summary: The user reported a suspicious email. The LLM suggested analyzing e-mail logs to identify if the reported email is malicious and if there are any other malicious emails that have been received by the organization. The user provided additional information that the actor received a suspicious email. The LLM recommended analyzing the email to determine if it contains malware.

Action name: Analyze E-mail.

Q3: After analysing the email, the logs say that it contains a link to a suspicious website

A: Let's think step by step. Since the user has already performed action A (Analyze E-mail) and the goal is to detect malicious actions and infected PCs and restore them, the next action to take would be B (Analyze Website Visits) since the description of B says that it checks which machines have visited certain web pages, including the one linked in the suspicious email.

Therefore, the name of the next action is: Analyze Website Visits.

Summary: This is what the human and the LLM have discussed so far:

The user reported a suspicious email. The LLM suggested analyzing e-mail logs to identify if the reported email is malicious and if there are any other malicious emails that have been received by the organization. The user provided additional information that the actor received a suspicious email. The LLM recommended analyzing the email to determine if it contains malware. The email was analyzed, and it was found to contain a link to a suspicious website. The LLM suggested analyzing website visits to check which machines have visited the linked web page.

Next action: Analyze Website Visits.

Q4: After analysing the website visits, the logs say that several employees have accessed the website.

A: Let's think step by step. Since we have already analyzed website visits and found that several employees have accessed the suspicious website, the next action to take would be to check if any of these employees' machines have been compromised.

The action to take would be "Check Indicators of Compromise". This action will scan all machines belonging to the organization and return a list of machines with matching indicators of compromise.

Final answer: Check Indicators of Compromise.

Ispitivanje Llama2 3, sumnjiva poveznica, RAG, temperatura 1:

Q1: One of my employees received a suspicious email.

Let's think step by step. The employee has already reported a suspicious email, and the goal is to detect malicious actions and infected PCs and restore them.

The next action to take would be "Analyze E-mail" since it can help identify potential threats and determine if the email is indeed malicious.

Action name: Analyze E-mail

Summary: This is what the human and the LLM have discussed so far: The employee reported a suspicious email. The LLM suggested analyzing the email to identify poten-

tial threats and determine if it's malicious. The human agreed and analyzed the email.

Q2: After analysing the email, the logs say that it contains a link to a suspicious website.

A: Let's think step by step. Since the user has already analyzed the email and found a link to a suspicious website, the next action to take would be to analyze the website visits. This action will check which machines have visited the suspicious web page. The goal is to detect malicious actions and infected PCs and restore them.

Therefore, the next action to take is "Analyze Website Visits."

Summary: This is what the human and the LLM have discussed so far: The employee reported a suspicious email, and the LLM suggested analyzing the email to identify potential threats and determine if it's malicious. The user agreed and analyzed the email, which contained a link to a suspicious website. The LLM recommended analyzing website visits to check which machines have visited the suspicious web page and detect malicious actions and infected PCs. The next action to take is "Analyze Website Visits."

Q3: After analysing the website visits, the logs say that several employees have accessed the website.

A: Let's think step by step. Since the employee has already performed action A (Analyzing the email) and their goal is to do B (Detect malicious actions and infected PCs and restore them), the next action to take would be C (Check Indicators of Compromise) since the description of C says that it does B.

Final answer: Check Indicators of Compromise.

Ispitivanje Llama2 3, malware, RAG, temperatura 0.05:

Q1: One of my employees received a suspicious email.

A: Let's think step by step. The employee has already reported a suspicious email, and the goal is to detect malicious actions and infected PCs and restore them.

The next action to take would be Analyze E-mail Logs since it can help identify potentially malicious emails received by the target actor. It also stores the logs on the mail server, which can be useful for further analysis.

Action name: Analyze E-mail Logs

Summary: This is what the human and the LLM have discussed so far:

The human reported a suspicious email. The LLM suggested analyzing email logs to identify potentially malicious emails received by the target actor and store them on the mail server for further analysis. The human agreed and did so.

Q2: The logs say the email contains a suspicious attachment.

A: Let's think step by step. Since the user already performed action A (Analyzing E-mail Logs) and their goal is to detect malicious actions and infected PCs and restore them, the next action to take would be B (Analyze Data) since the description of B says that it checks the importance of data and if it is present on machines belonging to your organization, then a report is generated showing all machines where this data is present.

Therefore, the name of the next action is: Analyze Data.

Summary: This is what the human and the LLM have discussed so far:

The human reported a suspicious email. The LLM suggested analyzing email logs to identify potentially malicious emails received by the target actor and store them on the mail server for further analysis. The human agreed and did so. The LLM suggested analyzing the data to check the importance of the data and generate a report showing all machines where the data is present.

After analyzing the email, the logs say that it contains a suspicious attachment. The LLM suggested analyzing the data to check the importance of the data and generate a report showing all machines where the data is present.

Q3: How can i check out the suspicious attachment?

A: Analyse Data

Ispitivanje Mixtral 1, temperatura 0.05, s CoT:

Q1: Hey I'd like to find confidential information about an organisation.

A: Let's think step by step... Since the user wants to find confidential information about an organization, the next action to take would be to perform a network scan to discover

machines and network services within the target organization. Final answer: Network Scan.

Summary: Firstly, the user said that they want to find confidential information about an organization. The LLM suggested performing a network scan to discover machines and network services within the target organization. The user then performed the network scan as suggested by the LLM. However, no further updates or issues have been reported by the user after performing the network scan.

Q2: But I know nothing about the organisation.

A: Let's think step by step... Since the user's goal is to find confidential information about an organization, but they know nothing about the organization, the next action to take would be to gather open-source intelligence (OSINT) on the target organization. Final answer: Gather OSINT on the target organization.

Ispitivanje Mixtral 2, temperatura 0.05, bez CoT:

Q1: Hey I'd like to find confidential information about an organisation.

A: Analyze Screen Capture.

Summary: Firstly, the user said that they want to find confidential information about an organization. The LLM suggested taking the action of Analyzing Screen Capture data. The user did this action and said that the problem now is understanding the screen capture data performed by malware trying to find information about network connections used by the person using the computer. The LLM then suggested doing the action of Interpreting Screen Capture Data.

Q2: How can I install a malware that performs that action on the user's PC?

A: Create and Deploy Malware

Q3: Can you give me the exact name of the action?

A: Creating and Deploying Malware