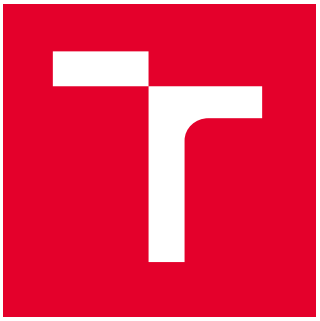


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

NÁSTROJ PRO OPTIMALIZACI NEURONOVÝCH SÍTÍ POMOCÍ ROZŠÍŘENÍ ZÁZNAMŮ BEZPEČNOSTNÍCH UDÁLOSTÍ

TOOL FOR OPTIMIZING NEURAL NETWORKS BY AUGMENTING SECURITY LOGS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — **Tereza Masárová**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Yehor Safonov**

SUPERVISOR

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Studentka: Tereza Masárová ***ID:*** 247580

Ročník: 3 ***Akademický rok:*** 2024/25

NÁZEV TÉMATU:

Nástroj pro optimalizaci neuronových sítí pomocí rozšíření záznamů bezpečnostních událostí

POKYNY PRO VYPRACOVÁNÍ:

Hlavním cílem bakalářské práce je návrh a implementace nástroje pro rozšíření logových záznamů určených k trénování neuronových sítí s cílem zlepšit jejich detekční a generalizační schopnosti. Nástroj umožní generovat a textově rozšiřovat bezpečnostní logy z klíčových technologií, jako jsou řadiče domény, databáze či nástroje typu XDR. V teoretické části práce bude provedena analýza aktuálního stavu vědy a techniky v oblasti rozšiřování logových datových sad. Součástí bude také rozbor klíčových bezpečnostních pojmů a standardů relevantních pro bezpečnostní monitoring. Zvláštní důraz bude kladen na porovnání účinnosti moderních metod textového rozšiřování a jejich využití v oblasti kybernetické bezpečnosti, a to s ohledem na současná bezpečnostní doporučení. Praktická část práce zahrnuje návrh, implementaci a experimentální ověření nástroje pro rozšiřování bezpečnostních logů. Nástroj umožní generování nových logových záznamů na základě vstupních dat a provádění různých druhů textových mutací, jako jsou syntaktické či obsahové úpravy surových logů. V rámci testování bude posouzena účinnost nástroje při trénování různých typů neuronových sítí zaměřených na syntaktickou analýzu logových záznamů.

DOPORUČENÁ LITERATURA:

[1] SHORTEN, Connor and Taghi M. KHOZADAN. Text Data Augmentation for Deep Learning. Springer International Publishing, 2021. ISBN 978-3-030-79752-0.

[2] MARTINEZ, Roberto Incident Response with Threat Intelligence. Packt Publishing, 2022. ISBN 1801070997.

Termín zadání: 10.2.2025 ***Termín odevzdání:*** 3.6.2025

Vedoucí práce: Ing. Yehor Safonov

prof. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

S rastúcou zložitosťou moderných sietí a informačných systémov sa zvyšuje aj množstvo záznamov generovaných týmito zariadeniami, ktoré sú nepretržito zhromažďované a analyzované s cieľom odhaľovať hrozby a chrániť pred kybernetickými útokmi. Cieľom tejto bakalárskej práce je návrh a implementácia nástroja na rozšírenie bezpečnostných logov, ktoré slúžia na tréning neurónových sietí a zlepšuje ich generalizačné schopnosti. Teoretická časť sa zameriava na analýzu aktuálneho stavu v oblasti bezpečnostného monitoringu, technológie logovania a využitie techník umelej inteligencie pre bezpečnostný monitoring. Rieši problematiku kategorizácie logov, moderných metód augmentácie textových údajov a ich aplikáciu na zvýšenie kvality a variability dát. Počas analýzy aktuálneho stavu bolo identifikovaných a preskúmaných viac ako 50 odborných článkov a publikácií zaoberajúcich sa textovou augmentáciou, ktoré boli neskôr realizované v rámci vývoja nástroja. Praktická časť zahŕňa návrh, implementáciu a experimentálne testovanie nástroja, ktorý umožní rozšírenie bezpečnostných logov generovaných rôznymi technológiami s cieľom zvýšiť efektivitu trénovania neurónových sietí. V rámci návrhu bola definovaná architektúra nástroja, identifikované typy dát vhodné na augmentáciu a vybrané vhodné knižnice na generovanie údajov. Nástroj umožňuje vytvárať nové logové záznamy prostredníctvom 75 generátorov využívajúcich moderných knižníc na generovanie syntetických dát a textových súborov na ktoré sa aplikuje 24 rôznych techník textovej augmentácie. Následne bol nástroj otestovaný na množine modelov neurónových sietí, kde sa sledoval vplyv augmentovaných dát na zlepšenie presnosti a generalizačných schopností.

KĽÚČOVÉ SLOVÁ

Augmentácia dát, bezpečnostné logy, bezpečnostný monitoring, generovanie syntetických dát, neurónové siete, logové formáty, textová augmentácia, SIEM

ABSTRACT

With the increasing complexity of modern networks and information systems, the volume of records generated by devices is also growing. These records are continuously collected and analyzed to detect threats and protect against cyberattacks. The aim of this bachelor's thesis is to design and implement a tool for augmenting security logs used to train neural networks, thereby improving their generalization capabilities. The theoretical part focuses on analyzing the current state of security monitoring, logging technologies, and the use of artificial intelligence techniques in security monitoring. It addresses the challenges of log categorization, modern methods of text data augmentation, and their application to enhance the quality and variability of data. During the analysis phase, more than 50 scientific articles and publications related to text augmentation were identified and studied, serving as the foundation for the tool's development. The practical part includes the design, implementation, and experimental testing of a tool that enables the augmentation of security logs generated by various technologies, with the goal of increasing the efficiency of neural network training. As part of the design, the tool's architecture was defined, data types suitable for augmentation were identified, and appropriate libraries for data generation were selected. The tool allows for the creation of new log records using 75 generators that leverage modern libraries for generating synthetic data and text files, which are further enhanced using 24 different text augmentation techniques. Subsequently, the tool was tested on a set of neural network models to evaluate the impact of augmented data on improving accuracy and generalization performance.

KEYWORDS

Data augmentation, security logs, security monitoring, synthetic data generation, neural networks, log formats, text augmentation, SIEM

MASÁROVÁ, Tereza. *Nástroj pro optimalizaci neuronových sítí pomocí rozšíření záznamů bezpečnostních událostí*. Bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2025. Vedúci práce: prof. Ing. Yehor Safonov

Vyhlásenie autora o pôvodnosti diela

Meno a priezvisko autora:	Tereza Masárová
VUT ID autora:	247580
Typ práce:	Bakalárska práca
Akademický rok:	2024/25
Téma závěrečnej práce:	Nástroj pro optimalizaci neuronových sítí pomocí rozšíření záznamů bezpečnostních událostí

Vyhlasujem, že svoju záverečnú prácu som vypracovala samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.*

Ako autorka uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušila autorské práva tretích osôb, najmä som nezasiahla nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomá následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autorky**

*Prehlasujem, že pri spracovaní tejto práce boli využité nástroje generatívnej umelej inteligencie, konkrétne ChatGPT, a to výlučne ako asistenčný nástroj na jazykovú a štylistickú úpravu textu, ako aj na návrhy spresnenia formulácií. Všetky výstupy boli kriticky zhodnotené, upravené a následne zapracované do práce.

**Autor podpisuje iba v tlačenej verzii.

POĎAKOVANIE

Veľmi rada by som vyjadrila úprimné poďakovanie vedúcemu mojej bakalárskej práce, pánovi Ing. Yehorovi Safonovovi, za jeho odborné vedenie, cenné rady a podnety, ktoré výrazne prispeli k úspešnému priebehu tejto práce. Oceňujem jeho trpezlivosť a ochotu pomôcť pri riešení výziev, ktoré sa počas realizácie objavili, ako aj pravidelné konzultácie, ktoré mi pomohli lepšie pochopiť problematiku a napredovať vo vypracovaní tejto práce. Veľká vďaka patrí aj mojej rodine a priateľom, za neutíchajúcu podporu počas celého štúdia.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

S narastajúcou komplexnosťou moderných sietí a informačných systémov rýchlo rastie aj objem záznamov generovaných týmito zariadeniami. Tie sú nepretržite zhromažďované, monitorované a analyzované v reálnom čase. Výsledkom tohto monitorovania je detekcia potenciálnych hrozieb a ochrana pred kybernetickými útokmi, čo pomáha predchádzať fatálnym škodám. Avšak so zvyšujúcim sa počtom kybernetických hrozieb a objemom zaznamenaných údajov sa tradičné monitorovacie techniky ukazujú ako nedostatočné [1].

Rastúce množstvo kybernetických hrozieb kladie dôraz na nepretržité monitorovanie a efektívne spracovanie obrovského objemu dát. Kritická potreba ochrany počítačovej infraštruktúry je obzvlášť zreteľná pri pohľade na štatistiky z roku 2024. Údaje z tretieho kvartálu odhaľujú výrazný nárast kybernetických útokov, pričom organizácie čelili v priemere 1 876 incidentom týždenne, čo predstavuje 75 % nárast oproti rovnakému obdobiu v roku 2023 [2]. Najviac zasiahnutými sektormi boli vláda a zdravotníctvo, kde počet útokov dosiahol 2 553, resp. 2 434 prípadov týždenne [2].

Vzhľadom na dynamicky sa meniacu povahu kybernetických hrozieb je nevyhnutné neustále zlepšovať metódy spracovania logových záznamov, ktoré zohrávajú kľúčovú úlohu pri detekcii, analýze a prevencii bezpečnostných incidentov. V súčasnosti predstavuje analýza týchto logov značnú výzvu, a to najmä pre ich nejednotnosť a obrovský objem dát generovaných v rámci informačných systémov [1]. Jedným z najefektívnejších spôsobov, ako skrátiť čas odozvy na incidenty a zvýšiť presnosť detekcie, je využitie techník umelej inteligencie (angl. *Artificial Intelligence*, AI) [3].

Podľa správy z roku 2024 sa podiel organizácií, ktoré integrovali AI do bezpečnostného monitorovania, zvýšil na 31 %, čo im umožnilo znížiť priemerné náklady na narušenie bezpečnosti na približne 1,72 milióna EUR [4]. Okrem toho táto integrácia pomohla skrátiť priemerný čas na identifikáciu (angl. *Mean Time to Identify*, MTTI) a priemerný čas na elimináciu (angl. *Mean Time to Contain*, MTTC) bezpečnostných incidentov o 33 % a 43 % [4]. Tieto štatistiky jasne dokazujú, že pokročilé metódy spracovania logových záznamov a implementácia AI technológií môžu výrazne zlepšiť efektivitu bezpečnostných riešení, čím organizáciám poskytujú nielen lepšiu ochranu pred útokmi, ale aj významné finančné úspory.

Cieľom tejto bakalárskej práce je navrhnutie a implementácia nástroja, ktorý umožní rozšírenie bezpečnostných logov generovaných rôznymi technológiami, ako sú EDR či XDR, s cieľom zvýšiť efektivitu trénovania neurónových sietí pri detekcii kybernetických hrozieb. Moderné techniky textovej augmentácie môžu túto situáciu zlepšiť tým, že umožnia vytváranie variabilných, ale relevantných dátových sád, ktoré prispejú k presnejšiemu detekčnému procesu.

Teoretická časť v prvej kapitole tejto práce 1 sa venuje analýze problematiky

bezpečnostného monitoringu a umelej inteligencie. Podrobne opisuje kľúčové bezpečnostné pojmy, ktoré sú dôležité pre pochopenie základov kybernetickej bezpečnosti a analyzuje kategorizáciu a formáty logov, ich spracovanie a význam v kontexte informačných systémov. Súčasťou teoretickej časti je aj prehľad technológií využívaných na bezpečnostný monitoring, ako aj metód umelej inteligencie a strojového učenia, ktoré sú aplikované pri detekcii hrozieb. Kľúčovým bodom v druhej kapitole 2 bude podrobná analýza techník textovej augmentácie a ich aplikáciu na rozšírenie logových záznamov, pričom sa zdôrazňuje ich význam pre zlepšenie kvality a variability dát.

Praktická časť zahŕňa návrh v tretej kapitole 3 a implementáciu nástroja v štvrtej kapitole 4 na rozšírenie logov s využitím moderných knižníc na generovanie syntetických dát. Táto časť práce sa sústreďuje na návrh štruktúry nástroja, vývoj čiastkových funkcií a následné testovanie jeho efektivity na rôznych typoch neurónových sietí. Experimenty sa zameriavajú na vyhodnotenie prínosu augmentácie pre presnosť a generalizačné schopnosti modelov umelej inteligencie. Výsledky a získané poznatky v piatej kapitole 5 budú slúžiť ako základ pre ďalší vývoj v oblasti zlepšovania detekcie kybernetických hrozieb.

1 Problematika bezpečnostného monitoringu a umelej inteligencie

Bezpečnostní analytici dnes čelia obrovskému množstvu údajov, v ktorých musia identifikovať skutočné hrozby v reálnom čase. Táto záťaž pramení predovšetkým z neustále pribúdajúcich logových záznamov generovaných modernými sieťami a systémami. Logové záznamy zohrávajú kľúčovú úlohu v kybernetickej bezpečnosti, pretože umožňujú identifikovať potenciálne hrozby a minimalizovať dopad kybernetických útokov. Napriek neustálemu zlepšovaniu monitorovacích systémov narážajú tradičné techniky výzvam spojeným s rastúcim objemom dát a výskytom stále sofistikovanejších hrozieb. Automatizované systémy založené na pravidlách sú síce účinné pri identifikácii známych hrozieb, no ich slabinou je nižšia efektivita pri detekcii nových, doposiaľ neznámych útokov. Z tohto dôvodu sa do popredia dostávajú metódy umelej inteligencie a strojového učenia (angl. *Machine Learning*, ML), ktoré v oblasti kybernetickej bezpečnosti umožňujú presnejšie identifikovať neobvyklé vzorce správania a rýchlejšie reagovať na nové typy hrozieb [3].

Začiatok kapitoly bude venovaný bezpečnostným pojmom a štandardom v rámci bezpečnostného monitoringu, ktoré sú dôležité pre porozumenie tejto témy. Ďalšia podkapitola sa bude zaoberať využitím technológií v rámci bezpečnostného monitoringu a možnosťami využitia umelej inteligencie pre bezpečnostný monitoring.

1.1 Kľúčové bezpečnostné pojmy

Bezpečnostný monitoring spočíva v neustálom analyzovaní a sledovaní počítačovej siete a systémov s cieľom predchádzať kybernetickým útokom. Jeho zámerom je identifikovať vyskytujúce sa zraniteľnosti v systéme a reagovať na potenciálne hrozby v reálnom čase. Opiera sa o širokú škálu pojmov a štandardov, ktoré tvoria základ pre pochopenie technológií, procesov a postupov, ktoré organizácie využívajú na ochranu svojich systémov a údajov.

- **Aktívum** (angl. *Asset*) – predstavuje akúkoľvek hodnotnú, dôležitú položku v rámci organizácie [5].
- **Bezpečnostná hrozba** (angl. *Security Threat*) – označuje subjekt, ktorý má schopnosti a motiváciu zneužiť zraniteľnosť aktíva [6].
- **Bezpečnostná udalosť** (angl. *Security event*) – situácia, ktorá môže spôsobiť alebo viesť k narušeniu informačných systémov a technológií, ako aj pravidiel stanovených na ich ochranu [7].
- **Zraniteľnosť** (angl. *Vulnerability*) – je slabina v systéme alebo v rámci zariadenia, ktorá môže umožniť útočníkovi neautorizovaný prístup k aktívu [8].

- **Zneužitie** (angl. *Exploit*) – je spôsob, akým sa útočí na zraniteľnosť [5].
- **Riziko** (angl. *Risk*) – je pravdepodobnosť, že hrozba využije zraniteľnosť [8].
- **Anomália** (angl. *Anomaly*) – je pozorovateľný jav v systéme alebo sieti, ktorý je považovaný za niečo neobvyklé [5].
- **Bezpečnostný incident** (angl. *Security Incident*) – predstavuje porušenie alebo bezprostrednú hrozbu porušenia bezpečnostných politík počítačov, pravidiel prijateľného používania alebo štandardných bezpečnostných postupov [6].
- **Záznam auditných logov** (angl. *Audit logging records*) – Uchovávanie informácií o udalostiach súvisiacich s informačnou bezpečnosťou s cieľom umožniť ich následnú kontrolu, analýzu a priebežné sledovanie bezpečnostného stavu systému [7].
- **Monitorovanie** (angl. *Monitoring*) – Zaznamenávanie informácií o udalostiach v oblasti informačnej a kybernetickej bezpečnosti za účelom ich následného preskúmania, analýzy a priebežného monitorovania. [7].

1.2 Problematika logových záznamov

Logy predstavujú kľúčový zdroj informácií v oblasti kybernetickej bezpečnosti. Slúžia na zaznamenávanie udalostí v systémoch, aplikáciách a sieťových zariadeniach, pričom umožňujú spätnú analýzu, detekciu anomálií, ako aj reakciu na bezpečnostné incidenty. Vzhľadom na rozmanitosť zdrojov, ktoré logy generujú, a potrebu ich efektívneho spracovania, je nevyhnutné porozumieť ich základným typom a formátom [9]. Táto kapitola sa zameriava na kategorizáciu logových záznamov podľa ich pôvodu a štruktúry, čím vytvára základ pre ich ďalšie spracovanie, analýzu a využitie pri trénovaní modelov umelej inteligencie.

Definícia logových záznamov

Logovací súbor je dokumentácia udalostí, ktorá je časovo označená a automaticky generovaná konkrétnym systémom. Logy predstavujú historické záznamy všetkého, čo sa odohráva na softvérových aplikáciach a systémoch [10]. V oblasti kybernetickej bezpečnosti sú logovacie súbory texty, ktoré poskytujú mimoriadne hodnotné informácie využiteľné na monitorovanie aktivít v rámci IT infraštruktúry, identifikáciu porušení politík, odhalenie podvodných alebo neobvyklých činností a zistenie bezpečnostných incidentov [11]. Bezpečnostné logy umožňujú správcovi systému rozpoznať a odhaliť akékoľvek pokusy o neoprávnené prihlásenie. Vďaka informáciám z logov môže správca následne zaviesť prísne opatrenia na posilnenie bezpečnosti systému. Údaje v logovacích súboroch môžu byť štruktúrované, čiastočne štruktúrované alebo neštruktúrované [9].

Typy logov

Každá zložka siete generuje iný typ dát a zhromažďuje ich vo vlastných logoch. Preto existuje veľké množstvo typov logov vrátane:

- **Záznam udalostí** (angl. *Event Log*) — obsahuje záznamy o sieťovej prevádzke a aktivitách, ako sú pokusy o prihlásenie, neúspešné zadania hesla alebo udalosti aplikácií [10].
- **Serverový log** (angl. *Server Log*) — ide o textový dokument, ktorý uchováva záznamy o aktivitách na konkrétnom serveri za dané časové obdobie [10].
- **Systémové logy** (angl. *System Log*) — zaznamenáva udalosti operačného systému, ako sú správy o spustení, zmeny v systéme, neočakávané vypnutia, varovania a chyby [10].
- **Logy autorizácie a prístupu** (angl. *Authorization Logs and Access Logs*) — obsahujú zoznam ľudí alebo botov, ktorí v daný čas pristupovali k určitým aplikáciám alebo súborom. Tieto logy pomáhajú pri riešení problémov s prístupom a zmene autentifikačných politík. Taktiež zaznamenávajú dôležité bezpečnostné udalosti na účely auditu [10].
- **Logy zmien** (angl. *Change Logs*) — chronologický záznam zmien vykonaných v aplikácii alebo súbore [10].
- **Logy dostupnosti** (angl. *Availability Logs*) – sledujú výkon systému, dobu jeho prevádzky a dostupnosť [10].
- **Zdrojové logy** (angl. *Resource Logs*) — poskytujú informácie o problémoch s konektivitou a kapacitnými limitmi [10].
- **Logy hrozieb** (angl. *Threat Logs*) — zaznamenávajú informácie o prenose súborových, systémových alebo aplikačných dát, ktoré zodpovedajú preddefinovaným bezpečnostným profilom vo firewalli [10].
- **Logy webových serverov** (angl. *Web Server Logs*) – obsahujú údaje ako „kto“ navštívil stránku (IP adresa) a „ktoré“ stránky si prehliadal (URL adresy). Okrem toho môže odhaliť pasce pre spamový obsah, ktorý nahodili hackeri, nefunkčné externé odkazy, nesprávne serverové odpovede a pokusy o zneužitie [12].
- **Sieťové logy** (angl. *Network Logs*) – môžu obsahovať informácie o neúspešných pokusoch o prihlásenie, odhaľovať neoprávnené pokusy o spustenie procesov alebo prístup k zamknutým údajom a množstvo ďalších dôležitých informácií [12].
- **Aplikačné logy** (angl. *Application Logs*) – sú záznamy aktivít zaznamenaných softvérových aplikácií. Tieto súbory môžu slúžiť na diagnostiku problémov, riešenie chýb a auditovanie. Poskytujú cenné informácie o výkonnosti aplikácie, ako sú varovania o nedostatku miesta na disku, zaznamenané ope-

rácie, problémy brániace spusteniu aplikácie, úspešné prihlásenia a pokusy o prihlásenie, ktoré zlyhali [12].
- **Bezpečnostné logy** (angl. *Security Logs*) – mnohé zariadenia udržiavajú bezpečnostné logy, ktoré umožňujú sledovať informácie súvisiace so zabezpečením v počítačovom systéme [12].

Typy logov na základe ich formátu

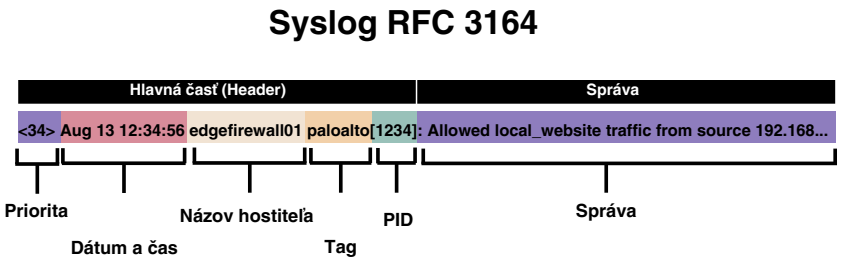
Logy sú často neštruktúrované textové údaje, čo sťažuje ich dotazovanie a hľadanie užitočných informácií. Neštruktúrované logy sú správy v obyčajnom texte, ktoré obsahujú informácie v lineárnom reťazci, avšak neštruktúrovaným spôsobom. Je náročné dotazovať a získať relevantné informácie bez analýzy lineárneho reťazca [13]. Štruktúrované logy sú namiesto reťazcov tvorené objektmi. Objekty môžu zahŕňať premenné, dátové štruktúry, metódy a funkcie. Napríklad objekt, ktorý je súčasťou správy logu, môže obsahovať podrobnosti o platforme alebo aplikácii [14]. Štruktúrovaný formát zabezpečuje, že logy sú strojovo čitateľné a ľahko spracovateľné, čo je jednou zo základných požiadaviek na softvér pre správu logov [15]. Medzi štruktúrované logy patria napríklad formáty JSON, XML, Syslog, CEF, KVP (angl. *key-value pair*), Logfmt a CSV, ktoré umožňujú efektívne spracovanie a analýzu logovacích údajov.

Syslog

System logging protocol (Syslog) je typ logovania, ktorý umožňuje administrátorovi systému sledovať a spravovať logy z rôznych častí systému. Tento protokol slúži na sledovanie udalostí a chýb a poskytuje aj informácie o výkone systému. Syslog možno využívať na Unixových systémoch, Windows a ďalších operačných systémoch. Syslog je štandard na zhromažďovanie a uchovávanie logovacích informácií, ktorý možno využiť aj na ich odoslanie na ďalšiu analýzu. Pokiaľ ide o samotný protokol syslog, tento využíva na prenos logovacích správ protokoly UDP (angl. *User Datagram Protocol*) a TCP (angl. *Transmission Control Protocol*). Správy cez UDP sa spravidla posielajú na port 514, zatiaľ čo pri TCP je to port 601, aj keď tieto porty možno prispôsobiť podľa konkrétnych potrieb systému. Syslog umožňuje do každej správy vložiť dodatočné údaje, napríklad ID procesu (PID), časovú pečiatku a názov hostiteľa, čo uľahčuje určenie miesta a času, kedy došlo k udalosti [16].

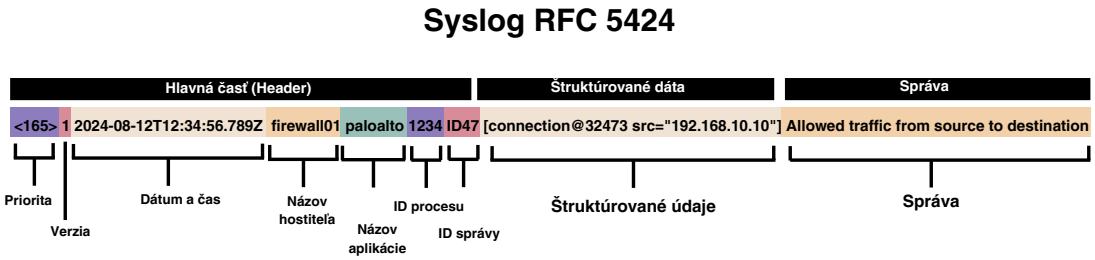
Syslog zároveň klasifikuje správy podľa úrovne závažnosti. Táto klasifikácia zahŕňa stupne od núdzovej situácie (angl. *emergency*) až po debugovacie správy (angl. *debug*), čo umožňuje filtrovať logy podľa dôležitosti a efektívne reagovať na kritické udalosti [16]. Syslog používa dva hlavné formáty:

- **RFC3164** – je jednoduchý a efektívny formát, ktorý sa používa na prenos logovacích správ. Obsahuje základné informácie ako časovú pečiatku, názov hostiteľa a úroveň závažnosti správy. Tento formát je široko podporovaný a používa sa v mnohých aplikáciách a zariadeniach. Ukážka formátu tohto typu logu je znázornená na obrázku 1.1 [17].



Obr. 1.1: Ukážka Syslog formátu RFC 3164 podľa [18].

- **RFC5424** – je novší a komplexnejší formát, ktorý poskytuje viac možností pre štruktúrovanie a prenos logovacích správ. Obsahuje podrobnejšie informácie o správe, ako sú metadáta, identifikátory udalostí a ďalšie kontextové informácie. Tento formát je vhodný pre moderné aplikácie a systémy, ktoré vyžadujú pokročilé možnosti správy logov. Príklad štruktúry tohto typu logu je znázornený na obrázku 1.2 [17].



Obr. 1.2: Ukážka Syslog formátu RFC 5424 podľa [18].

Správy zo zariadení alebo aplikácií môžu byť smerované na tzv. Syslog server, špecializovaný komponent, ktorý zhromažďuje, uchováva a spracováva logovacie dáta z rôznych zdrojov. Takýto server sa zvyčajne skladá z viacerých častí:

- **Syslog Listener** (angl. *Syslog Listener*) – prijíma prichádzajúce správy v reálnom čase.
- **Databáza** (angl. *Database*) – uchováva veľké množstvo logovacích záznamov pre účely vyhľadávania a archivácie.
- **Softvér na správu a filtrovanie** (angl. *Management and Filtering Software*) – zabezpečuje triedenie, koreláciu, notifikácie a reporting podľa nastavených pravidiel [15].

Z pohľadu architektúry funguje Syslog vo vrstvenom modeli pozostávajúcom z troch hlavných častí:

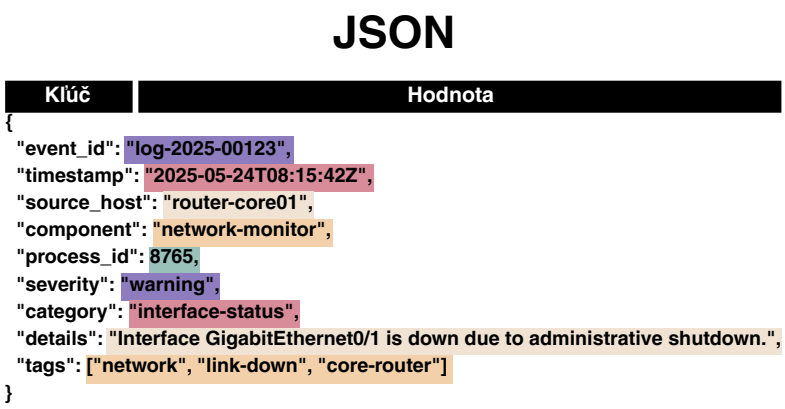
- **Aplikačná vrstva** (angl. *Application layer*) – zahŕňa zdrojové aplikácie, služby alebo systémové komponenty, ktoré generujú správy. Každá správa je vytvorená v súlade so štandardom Syslog, pričom definuje typ udalosti, úroveň závažnosti a ďalšie metadáta.
- **Prenosová vrstva** (angl. *Transport layer*) – zabezpečuje transport správ od zdrojov k cieľovým serverom prostredníctvom protokolov UDP alebo TCP. Dôležitá je spoľahlivosť a latencia prenosu, najmä pri kritických udalostiach.
- **Kolekčná vrstva** (angl. *Collection layer*) – predstavuje prijímaciu infraštruktúru, kde sa správy zbierajú, analyzujú a ďalej spracúvajú. Medzi jej hlavné úlohy patrí ukladanie správ do úložísk (súbory, databázy), ich smerovanie, generovanie upozornení a zabezpečenie centralizovaného prístupu pre analytikov [19].

JSON LOG FORMAT

JavaScript Object Notation (JSON) je moderný formát pre štruktúrované logovanie, v ktorom sa údaje ukladajú ako objekty v pároch kľúč-hodnota. Vďaka svojej jednoduchej čitateľnosti pre ľudí a zároveň efektívnej spracovateľnosti strojmi sa stal jedným z najpoužívanejších formátov v oblasti logovania [15]. V prostredí distribuovaných systémov, mikroservisov či cloudových platforiem je JSON štandardom pre transparentné a škálovateľné sledovanie udalostí. Prehľadnú ukážku takéhoto logu možno vidieť na obrázku 1.3. Typický JSON log obsahuje základné polia ako:

- **timestamp** – čas vzniku udalosti, väčšinou vo formáte ISO 8601,
- **level** – úroveň závažnosti udalosti (napr. *info*, *warning*, *error*),
- **message** – hlavný popis udalosti v textovej forme,
- **context/metadáta** – doplnkové informácie ako ID požiadavky, meno služby, IP adresa a podobne [20].

JSON logy môžu byť zapisované priamo do súborov alebo odosielané pomocou nástrojov ako Fluent Bit, Logstash, či cez HTTP API do systémov pre zber a analýzu logov (napr. Elasticsearch, Loki alebo OpenObserve) [20]. Táto flexibilita umožňuje jednoduchú integráciu do CI/CD pipeline-ov, kontajnerových platforiem (napr. Kubernetes) a cloudových služieb. Výhodou JSON formátu je schopnosť zachytiť detailný kontext udalosti, ktorý môže zahŕňať technické aj obchodné metadáta. Vďaka štruktúrovanosti sa dá s logmi efektívne pracovať, ľahko ich indexovať, filtrovať a analyzovať. Formát tak ponúka vysokú interoperabilitu medzi systémami a zjednodušuje diagnostiku, monitoring a auditovanie aplikácií v modernom softvérovom prostredí [20].



Obr. 1.3: Ukážka JSON formátu.

CEF FORMAT

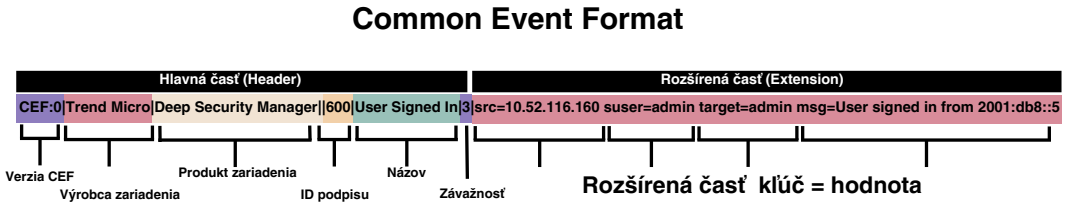
Common Event Format (CEF) je štruktúrovaný textový formát určený na logovanie a audit udalostí. Vyvinutý spoločnosťou ArcSight, tento formát umožňuje efektívne zdieľanie bezpečnostných informácií medzi rôznymi systémami, ako sú sieťové zariadenia, aplikácie či bezpečnostné nástroje. CEF je navrhnutý tak, aby zjednodušil centralizovaný zber logovacích údajov a podporoval integráciu s analytickými nástrojmi typu SIEM (napr. ArcSight, Splunk) [21].

Každý záznam v CEF pozostáva z dvoch častí: povinnej hlavičky a voliteľnej prílohy. Hlavička obsahuje základné údaje o udalosti, zatiaľ čo príloha poskytuje detailnejšie informácie vo forme párov kľúč-hodnota (napr. `src=192.168.0.1 dst=10.0.0.5 spt=443`). Polia sú v rámci správy oddelené znakom „|“, čo uľahčuje ich spracovanie. Hlavička obsahuje nasledujúce polia (schematicky znázornené na obrázku 1.4):

- **Version** – verzia CEF formátu (napr. 0).
- **Device Vendor, Product a Version** – identifikácia zdrojového zariadenia alebo softvéru.
- **Device Event Class ID** – identifikátor typu udalosti (napr. ID šablóny alebo typu chyby).
- **Name** – krátky, zrozumiteľný popis udalosti.
- **Severity** – stupeň závažnosti udalosti (napr. *Low*, *High*, *Very-High*) [21].

Na prenos CEF správ sa najčastejšie používa protokol Syslog, čo umožňuje kompatibilitu s existujúcimi logovacími infraštruktúrami. Výhodou CEF je jeho čitateľnosť, rozšíriteľnosť a široká podpora medzi bezpečnostnými technológiami. V praxi sa používa najmä v bezpečnostných auditoch, pri analýze incidentov a v rámci centralizovaných monitorovacích riešení. Medzi hlavné výhody CEF formátu patrí jeho

štandardizácia, jednoduchá analýza pomocou nástrojov typu SIEM, ako aj schopnosť efektívne prenášať štruktúrované informácie o udalostiach medzi rôznymi systémami.



Obr. 1.4: Ukážka CEF log formátu podľa [18].

1.3 Prehľad technológií pre bezpečnostný monitoring

S rastúcim množstvom kybernetických hrozieb a narastajúcou zložitosťou IT infraštruktúr sa potreba pokročilých technológií pre bezpečnostný monitoring stáva nevyhnutnosťou. Tieto technológie umožňujú detekciu, analýzu a reakciu na bezpečnostné incidenty v reálnom čase, čím pomáhajú chrániť kritické dáta a systémy. Využívajú sa na identifikáciu neobvyklých vzorcov správania, predchádzanie útokom a znižovanie rizík súvisiacich s narušením bezpečnosti. Táto podkapitola sa zameriava na význam týchto technológií, ich kľúčové úlohy v modernej kybernetickej bezpečnosti a spôsob, akým ich využitie formuje nástroje pre automatizovanú analýzu a detekciu hrozieb.

1.3.1 Problematika systému Log Manager

Log Manager je softvérové riešenie určené na centralizované zhromažďovanie, ukladanie, spracovanie a analýzu logových záznamov z rôznych IT systémov, aplikácií, zariadení a infraštruktúry. Logy predstavujú dôležitý zdroj informácií o dianí v systémoch, vrátane bezpečnostných incidentov, výkonnostných problémov a auditných udalostí. Správne spravované logy napomáhajú zvyšovaniu bezpečnosti, zefektívňovaniu IT operácií a zabezpečeniu súladu s legislatívnymi normami a auditnými požiadavkami [22].

Funkcionalita Log Managera

Log Manager poskytuje súbor funkcií zameraných na efektívne spracovanie, ukladanie a vyhodnocovanie logových údajov z rôznych zdrojov v rámci organizácie. Tieto funkcionality umožňujú IT a bezpečnostným tímom monitorovať stav infraštruktúry,

identifikovať problémy v reálnom čase, reagovať na incidenty a zabezpečiť dodržiavanie legislatívnych požiadaviek. Kľúčovým prvkom je schopnosť centralizovať logy, transformovať ich do použiteľného formátu a uľahčiť ich analýzu a vizualizáciu.

- **Zbieranie logov** (angl. *Log collection*) – Log Manager agreguje údaje z rôznych zdrojov, ako sú operačné systémy, servery, databázy, aplikácie, sieťové zariadenia a bezpečnostné systémy. Zbieranie prebieha v reálnom čase alebo na základe definovaných intervalov.
- **Monitorovanie udalostí** (angl. *Event monitoring*) – Sleduje aktivitu v systéme a identifikuje anomálie alebo podozrivé správanie na základe preddefinovaných pravidiel a metód detekcie.
- **Analýza logov** (angl. *Log analysis*) – Umožňuje vyhodnocovanie a koreláciu zozbieraných dát s cieľom identifikovať technické chyby, bezpečnostné incidenty a výkonové problémy. Využíva sa pri troubleshooting-u aj pri vyšetrovaní incidentov.
- **Uchovávanie údajov** (angl. *Log retention*) – Definuje pravidlá pre uchovávanie logových záznamov podľa legislatívnych požiadaviek a interných politík. Uchovávanie môže byť prispôsobené podľa typu údajov alebo závažnosti udalostí.
- **Indexovanie a vyhľadávanie** (angl. *Indexing and search*) – Log Manager umožňuje rýchle a efektívne vyhľadávanie v rozsiahlych objemoch logových dát pomocou indexovacích mechanizmov, filtrov a dotazovacích jazykov.
- **Reportovanie** (angl. *Reporting*) – Automatizované reporty a vizualizácie z logov poskytujú prehľad o stave IT prostredia, výkonnostných ukazovateľoch a bezpečnostných udalostiach. Slúžia aj ako podklady pre audity alebo regulačné orgány [23].

1.3.2 Problematika systému SIEM

Security Information and Event Management (SIEM) je nástroj, ktorý centralizuje zber a spracovanie logov zo všetkých bezpečnostných a sieťových zariadení. Tento systém zhromažďuje a agreguje údaje o udalostiach generovaných bezpečnostnými zariadeniami, sieťovou infraštruktúrou, IT systémami a aplikáciami, aby ich bolo možné využiť na rýchle vytváranie automatizovaných aj manuálnych bezpečnostných reakcií. Okrem toho, SIEM umožňuje analýzu týchto údajov v reálnom čase, čo pomáha identifikovať potenciálne hrozby a urýchliť reakciu na incidenty. Vďaka pokročilým analytickým funkciám a korelácii udalostí z rôznych zdrojov môžu organizácie efektívnejšie monitorovať svoju bezpečnostnú situáciu a znižovať riziko útokov. SIEM riešenia z pohľadu bezpečnosti poskytujú komplexný prehľad o celej sieti. Tieto systémy sa ľahko integrujú s rôznymi koncovými bodmi, sieťovými zaria-

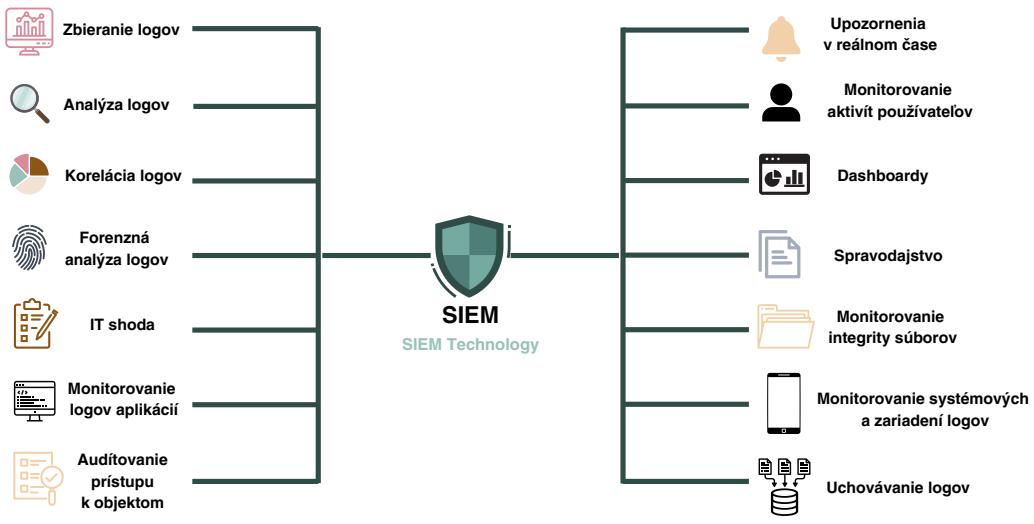
deniami a aplikáciami. Všetky tieto koncové body generujú logy s hodnotnými informáciami, ktoré pomáhajú odhaľovať bezpečnostné zraniteľnosti a uľahčujú reakciu na incidenty. SIEM umožňuje bezpečnostným tímom zachytávať logy a vykonávať ich analýzu v reálnom čase počas monitorovania [24].

Funkcionalita SIEM

SIEM softvér začína spravovaním a konsolidáciou údajov z logových záznamov, aby poskytol komplexný prehľad o celom prostredí. Rôzne zdroje dát posielajú protokoly, udalosti a kontextové informácie do SIEM systému. Agregačný modul tohto systému prijíma tieto informácie od zdrojov. Po ich prijatí ich odovzdáva normalizačnému modulu, ktorý prevádza dáta z rôznych formátov do formátu JSON. Týmto spôsobom sa zabezpečuje, že ostatné moduly navrhovaného SIEM systému môžu s týmito dátami efektívne manipulovať. Proces normalizácie využíva paralelizáciu, aby sa zvýšila rýchlosť spracovania, a následne sa dáta ukladajú do databázy prostredníctvom úložného modulu. Korelačný modul potom vykonáva analýzu normalizovaných dát a ak tieto údaje vyhovujú určitému korelačnému pravidlu, vytvorí sa upozornenie. To je následne zobrazené na vizualizačnom module (GUI) prostredníctvom modulu reálneho upozornenia najvhodnejším spôsobom [25]. Jednotlivé úlohy SIEM sú znázornené na 1.5.

- **Agregačný panel** (angl. *Aggregate dashboard*) – SIEM platforma poskytuje vizualizáciu udalostí z rôznych bezpečnostných zariadení. Tento panel zhromažďuje a zobrazuje údaje o udalostiach zo zdrojov, ako sú aplikácie, databázy, servery, firewally a iné systémy, čím poskytuje bezpečnostnému tímu komplexný pohľad na sieťovú a systémovú aktivitu. Takto môžu bezpečnostné tímy jednoducho sledovať potenciálne hrozby a rýchlo reagovať na incidenty [26].
- **Správa logov** (angl. *Log Management*) – SIEM zhormažďuje a agreguje logové záznamy, ktoré sú získavane z rôznych aplikácií, zariadení, sietí, infraštruktúry a systémov. Následne sú normalizované do spoločného formátu, aby sa zjednodušila analýza. Surové logy sú zložité na pochopenie pre bezpečnostných analitikov preto prebehne proces parsovania, kedy logy sú obohatené o kontextové informácie a umožňujú ľahšiu analýzu. Následne sú tieto logy uchovávane, to umožňuje efektívnejšiu analýzu, tvorbu reportov a forenzné vyšetrovanie. Správa logov pomáha bezpečnostným tímom identifikovať vzorce správania, odhaliť zraniteľnosti a rýchlo reagovať na bezpečnostné incidenty[26].
- **Detekcia hrozieb** (angl. *Threat detection*) – SIEM systémy môžu byť integrované s nástrojmi na hľadanie a detekciu hrozieb, čím sa zvyšuje viditeľnosť potenciálnych hrozieb a zraniteľností.

- **Alerty** (angl. *Alerting*) – SIEM systémy využívajú preddefinované pravidlá, agregovanú inteligenciu hrozieb, monitorovanie a strojové učenie na filtrovanie a prioritizáciu udalostí, čím generujú vysoko kvalitné upozornenia len pre tie problémy, ktoré sú pre organizáciu najdôležitejšie [26].
- **Reakcia na incidenty** (angl. *Incident response*) – Pokročilá analýza v SIEM riešeniach umožňuje bezpečnostným tímom lepšie pochopiť údaje, spolupracovať na riešení prípadov a efektívne reagovať na udalosti. Plnohodnotné SIEM systémy je možné integrovať s technológiou SOAR, čo umožňuje automatizovanú reakciu na hrozby [26].
- **Automatizácia** (angl. *Automation*) – SIEM softvér môže byť integrovaný s ďalšími bezpečnostnými nástrojmi, ako sú SOAR riešenia, čím umožňuje automatizáciu pracovných postupov a reakčných plánov na incidenty [26].
- **Súlad s reguláciami** (angl. *Compliance*) – SIEM produkty podporujú plnenie regulačných požiadaviek automatizáciou úloh, ako je monitorovanie údajov, udržiavanie záznamov pre účely auditu a vytváranie správ o zhode [26].
- **Integrácia** (angl. *Integration*) – SIEM riešenia dokážu spolupracovať s rôznymi bezpečnostnými systémami a nástrojmi. Pokročilé SIEM produkty poskytujú integráciu s Externými zdrojmi hrozieb, Cloudovými službami, SOAR nástrojmi, Analýzou správania používateľov a entít [26].



Obr. 1.5: Úlohy vykonávané systémami SIEM podľa [27].

1.3.3 Problematika systému EDR

Endpoint Detection and Response (EDR) je riešenie zamerané výhradne na ochranu koncových zariadení, ako sú pracovné stanice, notebooky alebo servery. Cieľom EDR je detegovať podozrivú aktivitu, reagovať na incidenty a uchovávať forenzné dáta pre neskoršiu analýzu. EDR poskytuje bezpečnostným analytikom nástroje na detailnú kontrolu nad tým, čo sa deje priamo na zariadeniach v sieti [28].

Funkcionalita EDR

EDR systémy zohrávajú kľúčovú úlohu pri ochrane koncových zariadení pred pokročilými hrozbami. Tieto riešenia kombinujú kontinuálne monitorovanie, zber dát, analýzu hrozieb a automatizovanú reakciu, čím poskytujú ucelený prístup k ochrane pred kybernetickými útokmi. Vďaka EDR môžu bezpečnostné tímy nielen rýchlo odhaliť podozrivú aktivitu, ale aj spätne analyzovať incidenty a prijímať nápravné opatrenia.

- **Monitorovanie koncových bodov** (angl. *Endpoint monitoring*) – EDR nepretržite sleduje činnosť zariadení, procesy, registruje zmeny súborov, vytváranie nových procesov a správanie používateľa.
- **Zber a ukladanie údajov** (angl. *Data collection and storage*) – Zachytáva a uchováva telemetriu zo zariadení pre účely analýzy, čo umožňuje spätné vyšetrovanie incidentov.
- **Detekcia hrozieb** (angl. *Threat detection*) – Identifikuje podozrivú aktivitu pomocou indikátorov kompromitácie (IoC) a behaviorálnej analýzy založenej na AI.
- **Alertovanie** (angl. *Alerting*) – Po detekcii anomálie EDR generuje upozornenie a poskytuje relevantné informácie pre bezpečnostný tím.
- **Reakcia na incidenty** (angl. *Incident response*) – EDR umožňuje bezpečnostným tímom vykonávať nápravné opatrenia, ako je izolácia zariadenia, ukončenie procesu alebo odstránenie malvéru.
- **Forenzná analýza** (angl. *Forensic analysis*) – Umožňuje detailné sledovanie priebehu incidentu a spätne analyzovať správanie útočníka alebo šírenie malvéru v sieti [29].

1.3.4 Problematika systému XDR

Extended Detection and Response (XDR) predstavuje moderný bezpečnostný systém, ktorý konsoliduje a automatizuje detekciu hrozieb, vyšetrovanie incidentov a reakciu naprieč rôznymi bezpečnostnými vrstvami vrátane koncových zariadení, sietí,

serverov a cloudového prostredia. Na rozdiel od tradičných nástrojov, ktoré fungujú izolovane XDR umožňuje širší kontext pri detekcii a zároveň zvyšuje efektivitu bezpečnostných tímov znižovaním falošných poplachov a zrýchlením reakcie na incidenty. Kombinuje technológie ako EDR, NDR a e-mailová bezpečnosť do jedného integrovaného riešenia [30].

Funkcionalita XDR

XDR predstavuje komplexný prístup k detekcii a reakcii na bezpečnostné incidenty naprieč rôznymi bezpečnostnými vrstvami. Jeho hlavným cieľom je odstrániť izolované systémy a poskytovať zjednotený pohľad na bezpečnostné dáta z viacerých zdrojov. Vďaka pokročilej analýze, automatizácii a centralizovanému spracovaniu incidentov pomáha XDR organizáciám zrýchliť odhaľovanie útokov, znížiť falošné pozitíva a efektívne reagovať na hrozby v reálnom čase.

- **Zjednotený prehľad o bezpečnosti** (angl. *Unified security visibility*) – XDR integruje dáta z viacerých bezpečnostných vrstiev (endpoint, sieť, cloud, e-mail), čím vytvára jednotný prehľad o bezpečnostných udalostiach naprieč celou organizáciou.
- **Pokročilá analýza hrozieb** (angl. *Advanced threat analytics*) – Využíva umelú inteligenciu a strojové učenie na rozpoznávanie hrozieb, ktoré by iné systémy mohli prehliadnuť, a koreluje dáta medzi rôznymi zdrojmi.
- **Automatizované reakcie** (angl. *Automated response*) – XDR dokáže automaticky reagovať na incidenty napríklad izoláciou zariadení, blokovaním IP adries alebo upravovaním firewallových pravidiel.
- **Zníženie falošných pozitív** (angl. *False positive reduction*) – Vďaka korelácii dát z rôznych zdrojov a pokročilým detekčným modelom XDR minimalizuje počet falošných upozornení a zvyšuje presnosť detekcie.
- **Centralizované vyšetrovanie incidentov** (angl. *Incident investigation hub*) – Poskytuje jedno centralizované rozhranie pre analýzu útokov, ich kontext a prepojenie jednotlivých fáz útoku.
- **Integrácia s ďalšími riešeniami** (angl. *Integration*) – XDR sa jednoducho integruje so SIEM, SOAR, EDR alebo NDR nástrojmi, čím zvyšuje efektivitu celého bezpečnostného ekosystému [30].

1.3.5 Problematika systému SOAR

Security Orchestration, Automation and Response (SOAR) je súbor technológií, ktoré organizáciám umožňujú efektívne zhromažďovanie, koordináciu a automatizáciu bezpečnostných operácií. Vďaka integráciám s rôznymi bezpečnostnými nástrojmi a systémami umožňuje SOAR zjednotené riadenie reakcie na incidenty podľa

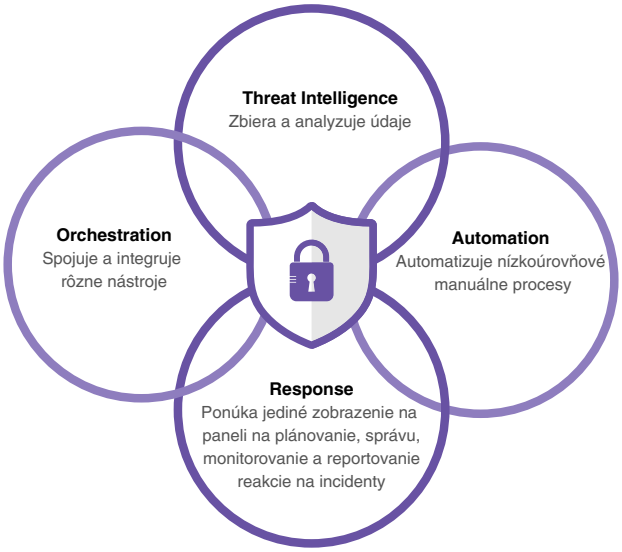
vopred definovaných pravidiel a postupov, čím znižuje záťaž na bezpečnostný tím a zvyšuje rýchlosť odozvy [31].

SOAR platformy pomáhajú bezpečnostným tímom efektívnejšie riešiť bezpečnostné incidenty, a to znížením potreby manuálnych zásahov, zlepšením konzistencie reakcií a zjednodušením forenzného vyšetrovania. Ich prínos spočíva aj v možnosti vizualizácie priebehu incidentu, centralizovanej dokumentácie a v analytike, ktorá pomáha odhaľovať opakujúce sa vzory hrozieb [32].

Funkcionalita SOAR

SOAR predstavuje komplexný systém na riadenie, automatizáciu a koordináciu reakcií na bezpečnostné incidenty. Umožňuje bezpečnostným tímom zrýchliť a zefektívniť proces riešenia hrozieb prostredníctvom centralizácie informácií, automatizovaných reakcií a jednotného rozhrania pre správu incidentov. Funkčnosť SOAR systémov možno rozdeliť do troch hlavných komponentov, ktoré sú úzko prepojené a spoločne tvoria základ pre moderné riadenie kybernetickej bezpečnosti. Jednotlivé úlohy SOAR sú znázornené na 1.6.

- **Bezpečnostná orchestrácia** (angl. *Security orchestration*) – Zabezpečuje prepojenie rôznych bezpečnostných nástrojov a zdrojov údajov prostredníctvom API integrácií. Pomocou tohto modulu môžu byť dáta centralizované a sprístupnené naprieč celým systémom, čo umožňuje ich efektívnu koordináciu pri riešení incidentov. Orchestrácia poskytuje SOC tímom možnosť centralizovaného zberu a prepojenia údajov z IDS/IPS, firewallov, EDR, SIEM či externých *threat intelligence* nástrojov [32, 33].
- **Bezpečnostná automatizácia** (angl. *Security automation*) – Umožňuje vykonávať rutinné a opakujúce sa úlohy bez potreby manuálneho zásahu. Typickými príkladmi sú filtrovanie alertov, otvorenie tiketov, overenie reputácie IP adries alebo vykonanie základnej analýzy logov. Automatizácia využíva playbooky, definované scenáre reakcií a zároveň môže vďaka umelej inteligencii a strojovému učeniu poskytovať odporúčania na ďalšie kroky [32, 33].
- **Reakcia na bezpečnostné incidenty** (angl. *Security response*) – Poskytuje jednotné rozhranie pre správu incidentov od ich identifikácie až po uzatvorenie a reporting. Tento komponent umožňuje rýchle vyšetrovanie, manažment úloh, zber dôkazov a zdieľanie informácií medzi členmi tímu. Takisto podporuje spätnú analýzu a tvorbu auditných záznamov, čo zvyšuje odolnosť voči budúcim hrozbám [32, 33].



Obr. 1.6: Úlohy vykonávané systémom SOAR podľa [34].

1.4 Využitie umelej inteligencie pre bezpečnostný monitoring

Umelá inteligencia (AI) sa stáva neoddeliteľnou súčasťou moderných bezpečnostných riešení, pričom umožňuje organizáciám automatizovať detekciu hrozieb, prevenciu a reakciu na incidenty. Využitím strojového učenia a hlbokého učenia dokáže AI analyzovať obrovské množstvo dát, identifikovať vzory a odhaliť anomálie, ktoré by mohli signalizovať potenciálne kybernetické hrozby. Takýto prístup nielenže zvyšuje efektivitu bezpečnostných tímov, ale tiež skracuje čas potrebný na identifikáciu a zvládnutie bezpečnostných incidentov. Správa z roku 2023 uvádza, že organizácie, ktoré zaviedli AI a automatizáciu v oblasti bezpečnosti, dokázali zvládnuť narušenia dát v priemere o 108 dní rýchlejšie a ušetrili približne 1,76 milióna USD na nákladoch spojených s reakciou na incidenty [35].

1.4.1 Spôsoby nasadenia umelej inteligencie

Pokročilé algoritmy strojového učenia a spracovania veľkých dát umožňujú AI efektívne rozpoznávať hrozby, reagovať na incidenty v reálnom čase a zvyšovať kybernetickú odolnosť systémov. Medzi hlavné oblasti využitia AI v bezpečnostnom monitoringu patrí:

- **Detekcia hrozieb v reálnom čase** (angl. *Real-time Threat Detection*) – AI analyzuje veľké množstvo dát z rôznych zdrojov (sieťová prevádzka, systémové

logy, používateľské správanie) a okamžite identifikuje podozrivé aktivity ako malware, phishing alebo neoprávnený prístup [36]. V prostredí SIEM systémov sa využíva na analýzu rozsiahlych bezpečnostných dát, pričom v simulovaných scenároch dosiahla AI úspešnosť detekcie známych hrozieb až 95 % a pri neznámych útokoch 87 % [37].

- **Anomálna detekcia** (angl. *Anomaly Detection*) – Vďaka schopnosti učiť sa z predchádzajúcich vzorcov správania AI rozpoznáva neštandardné správanie a odhaľuje nové typy útokov, ktoré by tradičné systémy nezachytili. V cloudových bezpečnostných operačných centrách (*Cloud-SOC*) zabezpečuje AI v reálnom čase detekciu malware s presnosťou 89 % a *insider threats* s úspešnosťou 85 %, pričom znižuje falošné poplachy až o 40 % [37].
- **Rýchla incidentná reakcia** (angl. *Rapid Incident Response*) – Automatizované systémy umožňujú rýchle rozhodovanie a reakciu na bezpečnostné incidenty, čím skracujú čas potrebný na zásah a minimalizujú škody. V rámci technológie SOAR skracuje AI reakčný čas až o 60 %. Umožňuje automatické kroky ako karanténovanie infikovaných zariadení, blokovanie škodlivých IP adries či informovanie tímu. Pri DDoS útokoch takto zachovala 95 % dostupnosť systémov a pri phishingových útokoch dosiahla detekčnú úspešnosť 92 % [37].
- **Riadenie zraniteľností** (angl. *Vulnerability Management*) – AI identifikuje slabé miesta v softvéri a infraštruktúre, analyzuje ich rizikovosť a pomáha ich prioritne riešiť, čo výrazne zefektívňuje *patch management* a eliminuje časové oneskorenie reakcie [38].
- **Overenie totožnosti** (angl. *Identity Verification*) – AI zlepšuje autentifikáciu pomocou biometrických údajov (rozpoznávanie tváre, hlasu) a viacfaktorovej autentifikácie. V kombinácii s behaviorálnou analýzou umožňuje identifikovať podozrivé prístupy aj pri kompromitovaných prihlasovacích údajoch [39].
- **Bezpečnostná analytika** (angl. *Security Analytics*) – AI umožňuje pokročilú analýzu bezpečnostných incidentov, identifikáciu trendov a tvorbu prediktívnych modelov, čím pomáha analytikom sústrediť sa na hrozby s najvyššou prioritou [38].
- **Spravodajstvo o hrozbách** (angl. *Threat Intelligence*) – Pomocou AI je možné analyzovať dáta z rôznych zdrojov (napr. sociálne siete, dark web, logy z cloudových platforiem) a identifikovať nové typy útokov alebo taktiky útočníkov ešte pred ich realizáciou [38].
- **Detekcia podvodov** (angl. *Fraud Detection*) – V odvetviach ako bankovníctvo alebo zdravotníctvo AI monitoruje transakcie a záznamy, aby odhalila podozrivú činnosť vrátane poistných podvodov alebo neoprávneného prístupu k citlivým údajom [38].
- **Nákladová efektivita a škálovateľnosť** (angl. *Cost Efficiency and Scala-*

bility) – Automatizácia procesov znižuje nároky na ľudské zdroje a zvyšuje možnosti nasadenia bezpečnostných systémov v rozsiahlejších sieťach, čo je obzvlášť dôležité pre cloudové prostredia [38].

- **Prediktívna bezpečnosť** (angl. *Predictive Security*) – Na základe historických dát AI predpovedá možné hrozby a umožňuje včasné prijatie preventívnych opatrení, čím organizáciám pomáha byť o krok pred útočníkmi [38].
- **Sémantická analýza logov a rozpoznávanie pomenovaných entít (NER)** (angl. *Log Analysis and Named Entity Recognition*) – NER je metóda spracovania prirodzeného jazyka (NLP), ktorá extrahuje a kategorizuje kľúčové informácie z textu, ako sú mená osôb, názvy organizácií, lokality, časové údaje, finančné sumy a percentá. V oblasti kybernetickej bezpečnosti sa NER využíva na analýzu systémových logov, identifikáciu relevantných entít a vytváranie kontextuálnych súvislostí medzi udalosťami. Týmto spôsobom pomáha pri detekcii pokročilých pretrvávajúcich hrozieb (APT) a korelácii incidentov naprieč rôznymi systémami. [40]

1.4.2 Základy strojového učenia

Strojové učenie predstavuje podmnožinu umelej inteligencie (AI), ktorá sa sústreďuje na schopnosť učiť počítače z údajov a zlepšovať sa na základe získaných skúseností, namiesto toho, aby boli explicitne programované [41]. Je vo všeobecnosti definovaná ako schopnosť stroja imitovať inteligentné ľudské správanie. V rámci strojového učenia sa algoritmy trénujú na identifikovanie vzorov a korelácií vo veľkých objemoch dát, pričom na základe tejto analýzy prijímajú najlepšie rozhodnutia a prognózy. V situáciách, kde je k dispozícii množstvo potenciálne správnych odpovedí, jedným z možných prístupov je označiť niektoré z týchto odpovedí ako platné a použiť ich ako tréningové dáta. Označovanie dát zahŕňa pridávanie informatívnych štítkov k neoznačeným informáciám. Tento proces zlepšuje celkovú účinnosť algoritmu a zvyšuje jeho schopnosť poskytovať čo najpresnejšie odpovede. Strojové učenie je veľmi užitočné v oblastiach, kde by manuálne programovanie bolo neefektívne alebo ťažko realizovateľné. Taktiež má blízko k štatistike, dolovaniu dát a lineárnej algebre, najmä vektory a matice [42].

Metódy strojového učenia

- **Učenie s učiteľom** (angl. *Supervised machine learning*) – využíva označené dátové sady na trénovanie algoritmov, aby mohli presne klasifikovať údaje alebo predpovedať výsledky. Keď sú vstupné údaje posielané do modelu, ten upravuje svoje váhy, kým nie je správne nastavený. Tento proces sa realizuje v rámci krížovej validácie, aby sa predišlo problémom ako preučovanie (angl. *overfitting*) alebo nedostatočné učenie (angl. *underfitting*) [43].

- **Učenie bez učiteľa** (angl. *Unsupervised machine learning*) – využíva algoritmy strojového učenia na analýzu a zoskupovanie neoznačených dátových súborov. Systému nie je povedaný správny výstup. Tieto algoritmy dokážu odhaliť skryté vzory a skupiny dát bez potreby zásahu človeka [43].
- **Spätno-väzbové učenie** (angl. *Semi-supervised learning*) – ponúka strednú cestu medzi supervised a unsupervised učením. Počas tréningového procesu využíva menšiu množinu označených dát na usmernenie klasifikácie a extrakcie vlastností z väčšej množiny neoznačených dát [43].

1.4.3 Princíp fungovania neurónových sietí

Neurónová sieť, známa aj ako umelá neurónová sieť, je výpočtová architektúra inšpirovaná spôsobom fungovania ľudského mozgu. Tieto siete sa skladajú z množstva spracovateľských jednotiek, nazývaných uzly, ktoré si navzájom odosielajú informácie, podobne ako neuróny v mozgu, ktoré prenášajú elektrické impulzy.

Neurónové siete sú základom strojového učenia, čo je oblasť, v ktorej počítačové programy získavajú schopnosti učiť sa bez potreby explicitných inštrukcií. Konkrétne v oblasti hlbokého učenia, čo je pokročilá forma strojového učenia, môžu tieto siete spracovávať neoznačené dáta a dospieť k záverom bez ľudskej pomoci. Napríklad hlboký učebný model založený na neurónovej sieti, ktorý je trénovaný na veľkom množstve dát, môže rozpoznať objekty na fotografii, ktoré nikdy predtým nevidel. Neurónové siete pozostávajú zo zbierky uzlov, ktoré sú vidieť na 1.7. Uzly sú rozmiestnené v najmenej troch vrstvách [44] . Tieto tri vrstvy sú [45]:

- **Vstupná vrstva** (angl. *Input layer*),
- **Skrytá vrstva** (angl. *Hidden layer*),
- **Výstupná vrstva** (angl.*Output layer*).

Vstupná vrstva

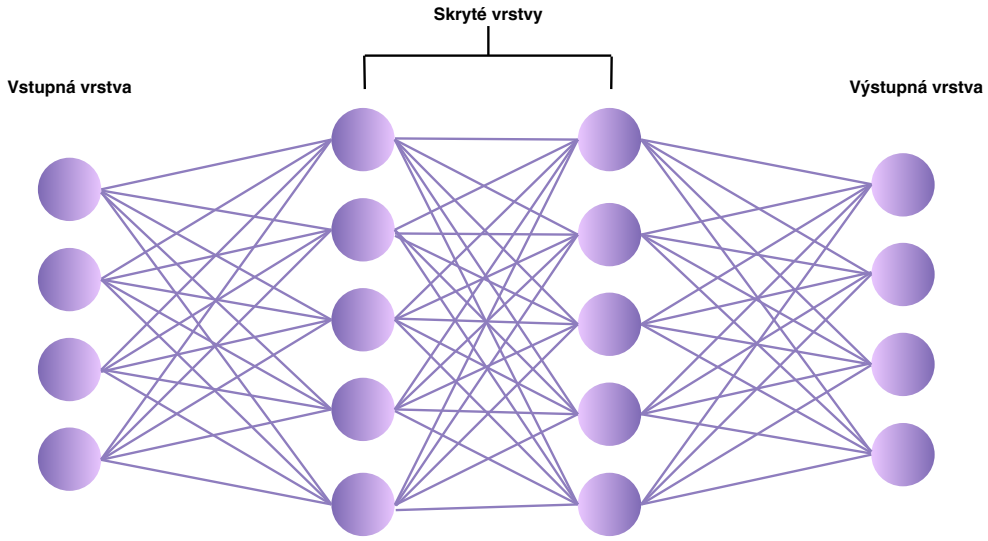
Informácie z vonkajšieho sveta vstupujú do umelej neurónovej siete cez vstupnú vrstvu. Uzly vo vstupnej vrstve spracúvajú dáta, analyzujú ich alebo ich kategorizujú a posúvajú ich do ďalšej vrstvy [45].

Skrytá vrstva

Skryté vrstvy prijímajú dáta zo vstupnej vrstvy alebo od predchádzajúcich skrytých vrstiev. Počet skrytých vrstiev v umelej neurónovej sieti môže byť veľmi vysoký. Každá z týchto vrstiev spracováva výstup z predchádzajúcej vrstvy, transformuje ho a odosiela výsledok ďalej do nasledujúcej vrstvy [45].

Výstupná vrstva

Výstupná vrstva poskytuje konečný výsledok spracovania dát umelou neurónovou sieťou. Môže mať jeden alebo viac uzlov. Napríklad, ak máme binárnu klasifikáciu („áno“/„nie“), výstupná vrstva bude obsahovať jeden výstupný uzol, ktorý vráti výsledok ako 1 alebo 0. Ak však ide o viactriednu klasifikáciu, výstupná vrstva môže obsahovať viac výstupných uzlov [45].



Obr. 1.7: Architektúra neurónovej siete.

2 Analýza rozšírenia záznamu udalostí

V tejto kapitole sa venujeme analýze procesov a techník súvisiacich s rozširovaním bezpečnostných logových záznamov, ktoré zohrávajú kľúčovú úlohu pri budovaní modelov umelej inteligencie v oblasti kybernetickej bezpečnosti. Vzhľadom na rastúci objem a rôznorodosť logových dát je čoraz náročnejšie získať dostatočne kvalitný dataset, ktorý by pokrýval aj zriedkavé alebo špecifické typy udalostí. Práve preto sa pozornosť sústreďuje na augmentáciu dát, ktorá umožňuje rozšíriť existujúce logy o nové varianty s cieľom zlepšiť výkonnosť a robustnosť trénovaných modelov.

Úvod kapitoly je zameraný na význam kvality dát pri trénovaní modelov strojového učenia a na výzvy spojené s ich zberom a anotáciou. Nasleduje podrobný opis procesu syntaktickej analýzy logov, ktorý umožňuje transformovať štruktúrované aj neštruktúrované záznamy do podoby vhodnej na ďalšie spracovanie. V ďalšej časti je predstavené predspracovanie dát, vrátane čistenia, transformácie, normalizácie a rozdelenia dátových množín, ako nevyhnutný krok pre úspešné nasadenie modelov umelej inteligencie.

Druhá časť kapitoly sa zameriava na prehľad aktuálnych prístupov k augmentácii dát, pričom dôraz sa kladie predovšetkým na textovú oblasť. Augmentačné techniky možno aplikovať na rôzne typy vstupných dát, najčastejšie ide o obrazové, zvukové a textové údaje. Každá z týchto kategórií si vyžaduje špecifické metódy spracovania, ktoré zohľadňujú ich štruktúru a charakter. Na obrázku 2.1 sú znázornené tri hlavné oblasti využitia augmentácie kde logy spadajú pod textovú.

V kontexte tejto práce je stredobodom záujmu textová augmentácia, ktorá zohráva dôležitú úlohu pri rozširovaní trénovacích dát v oblasti kybernetickej bezpečnosti. Logové záznamy predstavujú špecifický typ textových údajov, často štruktúrovaných alebo pološtruktúrovaných, ktoré dokumentujú správanie informačných systémov. Pri budovaní modelov strojového učenia určených na detekciu anomálií, klasifikáciu incidentov či predikciu chýb vzniká problém nevyváženosti dát, keďže niektoré typy udalostí sa vyskytujú veľmi zriedkavo. Textová augmentácia pomáha tento problém riešiť generovaním nových, mierne pozmenených verzií existujúcich záznamov, čím zvyšuje variabilitu a objem dostupných dát.

V tejto kapitole rozdeľujeme augmentačné techniky textovej augmentácia podľa úrovne aplikácie od znakov a slov, cez vety až po celé dokumenty. Okrem základných prístupov sú v kapitole predstavené aj pokročilé a hybridné metódy, ktoré kombinujú viaceré techniky s cieľom maximalizovať rozmanitosť generovaných údajov. Cieľom tejto analýzy je identifikovať najvhodnejšie prístupy k rozširovaniu bezpečnostných logov, ktoré prispejú k zlepšeniu kvality dátových vstupov a tým aj k vyššej presnosti a robustnosti trénovaných neurónových sietí.

Obr. 2.1: Typy dátovej augmentácie: textová, obrazová a zvuková.

2.1 Význam kvality dát pri rozšírení logových záznamov

Aby moderné modely strojového učenia dosahovali vysokú presnosť, typicky vyžadujú veľké množstvo kvalitne anotovaných dát. Proces zberu a anotácie dát sa však obvykle vykonáva manuálne a spotrebúva veľa času a zdrojov. Kvalita a reprezentatívnosť spracovaných dát pre konkrétnu úlohu často závisí od dostupnosti čistých dát v danej doméne a od úrovne odbornosti zapojených vývojárov. V reálnych aplikáciách býva často neuskutočniteľné získať dostatočné množstvo tréningových dát. V súčasnosti je najefektívnejším riešením tohto problému augmentácia dát, ktorej hlavným cieľom je zvýšiť objem, kvalitu a rozmanitosť tréningových dát [46].

2.1.1 Proces syntaktickej analýzy logových dát

Aby logovací systém mohol efektívne spracovať logové súbory, musí ich najprv parsovať, čiže extrahovať dôležité informácie, tento proces je znázornený na obrázku 2.2 na ktorom je vidieť príklad procesu parsovania logu, kde z pôvodného záznamu sa extrahujú štruktúrované polia ako časová pečiatka, úroveň, komponent, šablóna udalosti a jej parametre. Parsovanie prekladá štruktúrované alebo neštruktúrované logové záznamy do formátu, ktorý systém vie prečítať, indexovať a uložiť. Vďaka tomu je možné jednoduché filtrovanie, analýzu a prácu s informáciami v pároch kľúč-hodnota [47]. Väčšina logovacích systémov má zabudované parsery na spracovanie bežných dátových formátov, ako sú Windows Event Logs, JSON alebo CSV, ktoré boli zmienené v podkapitole 1.2. Tieto parsery identifikujú štruktúru logov a na základe toho extrahujú kľúčové polia a hodnoty, čo umožňuje efektívne a prehľadné spracovanie. Niekedy sa dáta ukladajú do hierarchických štruktúr, čím používateľ získava možnosť vykonávať podrobné a cielené vyhľadávanie. Pre neštandardné logy je často potrebné vytvoriť vlastné pravidlá parsovania. To sa realizuje pomocou regulárnych výrazov alebo špeciálneho skriptovacieho jazyka, ktorý poskytuje daný logovací systém. Pokročilé nástroje uľahčujú tento proces tým, že ponúkajú grafické rozhranie, v ktorom môžu používatelia označiť dôležité polia, zatiaľ čo systém

automaticky vytvára potrebné pravidlo [47].

Záznam surového logu

```
2025-04-11 14:35:22,789 WARNING firewall.ServiceMonitor: Blocked
unauthorized access to port 8080 from 192.168.1.100
```

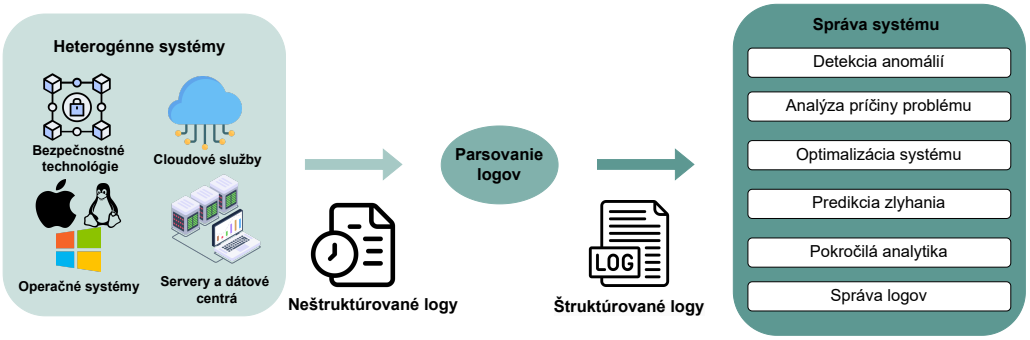
Parsovanie logu

Štruktúrovaný log

POLE	HODNOTA
TIMESTAMP	2025-04-11 14:35:22,789
LEVEL	WARNING
COMPONENT	firewall.ServiceMonitor
EVENT TEMPLATE	Blocked unauthorized access to port <> *from* <>
PARAMETERS	["8080", "192.168.1.100"]

Obr. 2.2: Spracovávanie logových záznamov.

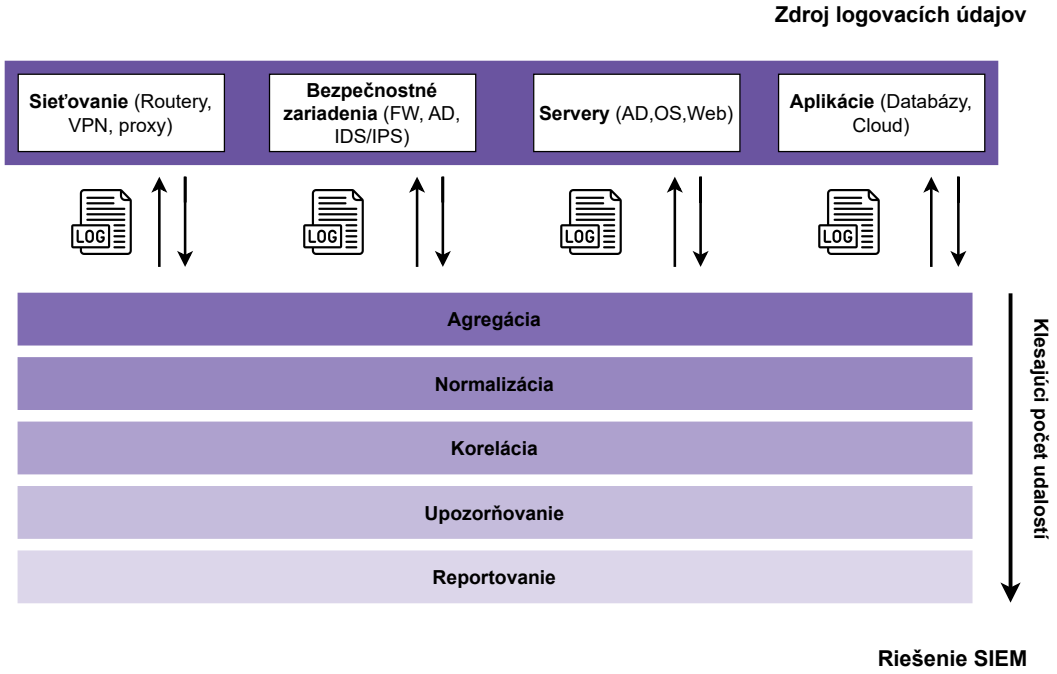
Keď sú logy úspešne parsované a transformované na konzistentný formát, systém ich uloží a umožní ich vyhľadávanie, analýzu a vizualizáciu. Obrázok 2.3 znázorňuje celý proces spracovania logov od ich generovania v rôznych heterogénnych systémoch, cez fázu parsovania neštruktúrovaných údajov, až po ich využitie v rôznych oblastiach správy systému. Tieto parsované logy môžu byť následne augmentované, čo znamená, že sa aplikujú rôzne techniky na rozšírenie trénovacej dátovej sady. Augmentácia pomáha modelom lepšie sa naučiť jemné vzory a odchýlky, čo zlepšuje ich schopnosť detegovať anomálie. V praxi to môže zahŕňať úpravy, ako je náhodné vkladanie, zámenná výmena alebo nahradenie znakov v logových záznamoch, aby sa zvýšila robustnosť a efektívnosť analýzy [47].



Obr. 2.3: Proces parsovania logov.

2.1.2 Predspracovanie logových dát pre strojové učenie

Predspracovanie dát je proces hodnotenia, filtrovania, manipulácie a kódovania údajov tak, aby ich algoritmus strojového učenia dokázal pochopiť a využiť výsledný výstup. Hlavným cieľom predspracovania dát je odstrániť problémy s údajmi, ako sú chýbajúce hodnoty, zlepšiť kvalitu dát a spraviť ich použiteľnými pre účely strojového učenia [48]. Na obrázku 2.4 je znázornený typický proces spracovania logových údajov v rámci riešenia SIEM. Logy pochádzajú z rôznych zdrojov ako sú sieťových prvkov, bezpečnostných zariadení, serverov a aplikačných systémov. Tieto logy následne prechádzajú sériou operácií, medzi ktoré patrí agregácia, normalizácia, korelácia, upozorňovanie a reportovanie, čím aj klesá množstvo udalostí.



Obr. 2.4: Vykonané operácie počas transformácie logových dát v riešení SIEM.

Proces zberu dát

Dáta sú palivom pre každý AI model. Ich získanie a príprava sú jedným z najdôležitejších krokov v procese vývoja. Zber relevantných, kvalitných dát z rôznych zdrojov zabezpečuje, že AI systém bude mať dostatok informácií na učenie a bude schopný robiť presné predikcie. Prvým krokom v procese prípravy dát je získanie potrebných údajov. To zahŕňa identifikáciu relevantných dátových zdrojov, ako sú databázy, API rozhrania, súbory alebo web scraping, a načítanie dát z týchto zdrojov. Je kľúčové zhromaždiť komplexné a presné dáta, ktoré sú v súlade s cieľmi analýzy alebo

projektu. Správne získavanie dát vytvára pevný základ pre ďalšie kroky v procese prípravy dát [49].

Proces čistenia dát

Proces čistenia údajov slúži na detekciu nesprávnych alebo hlučných údajov a ich opravu alebo odstránenie z datasetu. Ide o filtrovanie dát, ktoré zahŕňa odstránenie nepotrebných dát zo súboru, aby sa zameranie presunulo na relevantné informácie. Zvyčajné kroky zahŕňajú:

- **Odstránenie irelevantných alebo duplicitných údajov** – Pri spájaní údajov z rôznych zdrojov alebo získavaní údajov automatizovanými metódami, ako je *web scraping*, často dochádza k duplicite alebo pridávaniu údajov, ktoré nie sú relevantné pre cieľ analýzy [50].
- **Oprava štrukturálnych chýb** – Nekonzistentnosť môže vzniknúť v dôsledku nesprávneho označovania kategórií, zvláštnych názvových konvencií, preklepov alebo nesprávneho veľkého/malého písania [50].
- **Riešenie chýbajúcich hodnôt** – V datasetoch často chýbajú údaje v určitých stĺpcoch. Tento problém môže vzniknúť pri validácii alebo zhromažďovaní údajov. Pri rozumnom počte chýbajúcich hodnôt môžu byť doplnené metódami ako priemer, medián alebo mód [50].

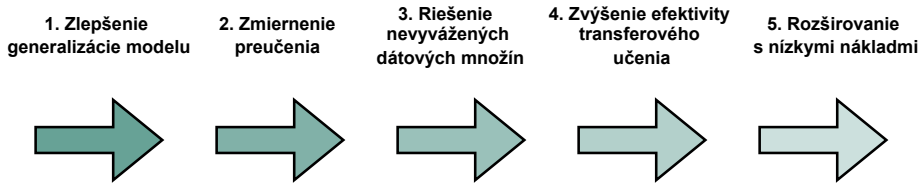
Transformácia dát

Transformácia dát je proces prevodu surových dát z jedného formátu, štruktúry alebo reprezentácie na iný, aby sa stali vhodnejšie pre konkrétnu úlohu alebo analýzu. Tento proces zahŕňa čistenie, agregáciu, filtrovanie alebo preformátovanie dát tak, aby vyhovovali požiadavkám zamýšľaného využitia. Transformácia zabezpečuje kompatibilitu s cieľovými systémami a zlepšuje kvalitu a použiteľnosť dát. Je to základný krok v cykle spracovania dát, ktorý zabezpečuje, že dáta získané prostredníctvom dátových pipeline a uložené v dátových skladoch a iných systémoch sú čisté, štruktúrované a pripravené na analýzu. Tento proces umožňuje organizáciám získavať cenné poznatky, robiť informované rozhodnutia a vyvíjať presné modely v oblastiach strojové učenie [51, 52].

Augmentácia dát

Augmentácia dát je technika, ktorá využíva existujúce dáta na vytváranie nových vzoriek, čím zlepšuje optimalizáciu a generalizáciu modelov strojového učenia. Ide o metódu, ktorá dopĺňa neúplné datasety generovaním modifikovaných kópií existujúcich dát, čím sa zvyšuje veľkosť a rozmanitosť datasetu. V oblasti strojového

učenia sa augmentácia dát prejavuje vytváraním upravených verzií existujúcich dát, čo zvyšuje veľkosť a rozmanitosť datasetu. To pomáha modelom lepšie generalizovať a znižuje riziko preučenia. Augmentácia dát je obzvlášť užitočná pri práci s nevyváženými datasetmi, kde niektoré triedy majú menej vzoriek ako iné. Pomocou augmentácie môžeme vytvoriť nové vzorky pre menej zastúpené triedy, čím sa dosiahne lepšia rovnováha v datasete [53]. Výhody využitia augmentácie dát sú znázornené na obrázku 2.5, kde sú zhrnuté kľúčové prínosy, ako napríklad zlepšenie generalizácie modelu, riešenie nevyvážených dát či zníženie nákladov.



Obr. 2.5: Kľúčové prínosy augmentácie dát.

Rozdelenie dát

Efektívne rozdelenie dát je kľúčovým krokom pri vývoji modelov strojového učenia, pretože zabezpečuje, že model sa naučí správne vzory a zároveň bude schopný generalizovať na nové, neznáme dáta. Dáta sa zvyčajne rozdeľujú do troch hlavných množín:

- **Trénovacia množina** (angl. *Training set*) – Používa sa na učenie modelu. Obsahuje označené príklady, na základe ktorých model upravuje svoje vnútorné parametre s cieľom minimalizovať chybu [54].
- **Validačná množina** (angl. *Validation set*) – Slúži na ladenie hyperparametrov modelu a na monitorovanie jeho výkonu počas trénovania. Pomáha identifikovať problémy ako preučenie (angl. *overfitting*) a zabezpečuje, že model sa dobre generalizuje na nové dáta [54].
- **Testovacia množina** (angl. *Test set*) – Používa sa na finálne zhodnotenie výkonnosti modelu po trénovaní a ladení. Poskytuje nezávislý pohľad na schopnosť modelu predikovať výsledky na neznámych dátach [54].

Správne rozdelenie dát znižuje riziko preučenia a zvyšuje dôveryhodnosť výsledkov modelu [54].

Prevod do vhodného formátu

Po získaní, vyčistení a transformácii dát je nevyhnutné zabezpečiť, aby boli pripravené v technicky vhodnom formáte pre spracovanie modelom umelej inteligencie.

Tento krok je zásadný pre efektívnosť výpočtov, kompatibilitu s knižnicami strojového učenia a minimalizáciu chýb pri načítaní dát.

V rámci tohto procesu sa vykonávajú nasledujúce činnosti:

- **Konverzia do štandardných dátových štruktúr** – Dáta sú často ukladané ako tabuľky (napr. *DataFrame* v knižnici `pandas`), matice (`NumPy array`) alebo tenzory (`PyTorch`, `TensorFlow`), ktoré umožňujú efektívne vektorové operácie [55], [56].
- **Výber dátového formátu podľa účelu** – Pre analýzy a experimenty sa využívajú bežné formáty ako CSV, JSON, Parquet alebo binárne formáty ako HDF5. Výber závisí od veľkosti datasetu, zložitosti dát a požiadaviek na výkon [55], [56].
- **Zabezpečenie konzistentných dátových typov** – Je potrebné uistiť sa, že všetky atribúty majú správne dátové typy (napr. `float32` pre numerické vstupy, `int64` pre kategorizované údaje), čo znižuje riziko chýb počas trénovania modelu [55], [56].
- **Serializácia dát** – V prípade potreby sa dáta ukladajú do formátov, ktoré sú priamo čitateľné pre modely počas inferencie, ako napríklad `TFRecord` pre TensorFlow alebo `Pickle` pre Python [55], [56].

Trénovanie AI

Trénovanie umelej inteligencie je proces, počas ktorého sa algoritmy učia zo vstupných dát aby dokázali čo najpresnejšie predpovedať výsledky. V praxi to znamená, že model dostáva trénovaciu množinu údajov, vrátane augmentovaných dát, ktoré rozširujú a obohacujú pôvodnú množinu a postupne upravuje svoje vnútorné parametre s cieľom znížiť chybu medzi predpoveďou a skutočným výstupom. Tento iteratívny proces je základom strojového učenia. Počas tréningu sa využívajú algoritmy ako napríklad lineárna regresia, rozhodovacie stromy, alebo hlboké neurónové siete. Model sa učí na základe spätnej väzby a doladením váh zlepšuje svoju schopnosť generalizovať na nové, neznáme dáta. Kvalita a reprezentatívnosť trénovacích dát sú kľúčové, nekvalitné alebo neúplné dáta vedú k slabému výkonu modelu [57].

Vyhodnocovanie

Vyhodnocovanie výkonu modelu je kľúčovým krokom v procese vývoja umelej inteligencie. Umožňuje posúdiť, ako dobre model predikuje výsledky a zároveň odhaliť oblasti, v ktorých je potrebné zlepšenie. Výber správnych metrík závisí od typu úlohy, klasifikácia alebo regresia.

Metriky pre klasifikáciu:

- **Miera správnosti** (angl. *Accuracy*) – Podiel správne predikovaných prípadov voči celkovému počtu prípadov. Vhodná pri vyvážených dátach [58].
- **Presnosť** (angl. *Precision*) – Podiel správne predikovaných pozitívnych prípadov voči všetkým predikovaným pozitívnym prípadom. Dôležitá pri minimalizácii falošných pozitív [58].
- **Zachytenie** (angl. *Recall*) – Podiel správne predikovaných pozitívnych prípadov voči všetkým skutočne pozitívnym prípadom. Dôležitá pri minimalizácii falošných negatív [58].
- **F1 skóre** (angl. *F1 score*) – Harmonický priemer presnosti (*Precision*) a zachytenia (*Recall*). Užitočné pri nevyvážených dátach, kde je dôležité nájsť rovnováhu medzi falošnými pozitívmi a negatívmi [58].
- **ROC-AUC** (angl. *Receiver Operating Characteristic - Area Under Curve*) – Oblasť pod krivkou ROC, ktorá hodnotí schopnosť modelu rozlišovať medzi triedami [58].

2.1.3 Účel rozširovania logových záznamov

Modely hlbokého učenia sa spoliehajú na veľké množstvá rozmanitých dát, aby dokázali vytvárať presné predikcie v rôznych situáciách. Augmentácia dát dopĺňa tvorbu variácií existujúcich dát, čo modelu pomáha zlepšiť presnosť jeho predikcií [59].

Zlepšený výkon modelu

Techniky augmentácie dát rozširujú pôvodný dataset vytváraním rôznych modifikácií existujúcich vzoriek. Tým zvyšujú objem tréningových dát a zároveň zvyšujú ich variabilitu, čo modelu umožňuje efektívnejšie sa učiť na rôznorodých vstupoch. Augmentované dáta pomáhajú modelu lepšie generalizovať na neznáme vstupy a zlepšiť jeho celkový výkon v reálnych podmienkach [59].

Znížená závislosť na objemu dát

Zber a príprava veľkého množstva tréningových dát môže byť nákladná a časovo náročná. Pomocou augmentácie je však možné zvýšiť efektivitu aj menších datasetov, čím sa znižuje potreba zhromažďovať rozsiahle množstvá dát. Menšie dátové súbory je možné vhodne rozšíriť pomocou syntetický vytvorených vzoriek [59].

Obmedzenie preučenia modelu (angl. *overfitting*)

Augmentácia dát pomáha predchádzať *overfittingu* počas tréningu modelov strojového učenia. *Overfitting* je nežiaduci jav, pri ktorom model síce správne predikuje

výsledky na tréningových dátach, ale zlyháva pri nových vstupoch. Ak sa model učí len z úzkeho spektra dát, môže sa „naučiť naspamäť“ a reagovať len na špecifické typy dát. Vďaka augmentácii vzniká rozsiahlejší a pestrejší dataset, ktorý pomáha modelom naučiť sa všeobecnejšie vzory a neviazať sa len na konkrétne vlastnosti dát [59].

Zvýšená ochrany súkromia dát

Ak je potrebné trénovať model hlbokého učenia na citlivých dátach, môžu sa použiť techniky augmentácie na vytvorenie syntetických dát. Takto vzniknuté augmentované dáta si zachovávajú význam pôvodných dát, zároveň však chránia citlivý obsah a obmedzujú prístup k originálnym dátam [59].

2.1.4 Aktuálny stav rozširovania logových záznamov

Počas analýzy textovej augmentácie bol identifikovaný prístup, ktorý sa na rozšírenie logových záznamov pozerá odlišným spôsobom. V článku [60] autori navrhujú množinu transformácií založených na zavádzaní šumu do štruktúry samotných logov, čím vytvárajú syntetické vzorky zachovávajúce štrukturálne vlastnosti procesu. Zatiaľ čo autori uvedeného článku sa sústreďujú na štrukturálne transformácie logov, v tejto práci sa zameriavame na augmentáciu samotných entít v rámci logových záznamov prostredníctvom techník textovej augmentácie. Aj keď sa táto práca nešpecializuje na textovú augmentáciu logových záznamov, bolo dôležité zmieniť článok, pretože počas analýzy dostupnej literatúry sa ukázal ako jedna z mála štúdií, ktorá sa explicitne zaoberá augmentáciou logov v kontexte prediktívneho monitorovania.

2.2 Aktuálny stav textovej augmentácii

V mnohých prípadoch strojového učenia nie je k dispozícii dostatok údajov na trénovanie vysoko kvalitného klasifikátora. Riešením tohto problému môže byť augmentácia dát, ktorá umelo zvyšuje množstvo dostupných tréningových údajov pomocou rôznych transformácií. Táto technika je nielen užitočná na prekonanie obmedzeného objemu dát, ale dokáže riešiť aj množstvo ďalších výziev. Patria sem napríklad zlepšenie schopnosti modelu generalizovať, regularizácia cieľovej funkcie, či zníženie množstva údajov potrebných na trénovanie s cieľom lepšie chrániť súkromie [61]. Nasledujúce podkapitoly sa zameriavajú na rôzne prístupy textovej augmentácie, ktoré môžu významne podporiť výkonnosť modelov v úlohách spracovania prirodzeného jazyka. Metódy augmentácie textu zlepšujú modely spracovania prirodzeného jazyka (NLP) úpravou a rozširovaním existujúcich textových dát. Tieto techniky fungujú na rôznych úrovniach od úprav jednotlivých znakov a slov až po transformáciu celých

viet a dokumentov. Zavádzaním rozmanitosti do tréningových dátových súborov pomáha augmentácia zvyšovať robustnosť modelov a ich schopnosť generalizácie. Augmentácia sa dá účinne využiť v rôznych NLP úlohách, čo dokazuje množstvo štúdií skúmajúcich rôznorodé prístupy a stratégie augmentácie, ako napríklad tie, ktoré predstavili Bayer a kol. [61], Shorten a kol. [62], Li a kol. [63]. V týchto prácach sú preskúmané rôzne metódy augmentácie textu s cieľom identifikovať celý rozsah dostupných prístupov k augmentácii v oblasti NLP. Na základe tohto prehľadu bude vybrané vhodné metódy, ktoré budú aplikované v nasledujúcej kapitole.

2.2.1 Augmentácia na úrovni znakov

Jednou z najjednoduchších metód augmentácie na úrovni znakov je **Indukcia šumu** (angl. *Noise Induction*), ktorá spočíva v zavádzaní náhodných chýb do textu. Štúdia Belinkov a Bisk (2018) [64] testovala tento prístup v kontexte strojového prekladu, pričom autori aplikovali rôzne typy umelého šumu, ako napríklad **Prehodenie znakov** (angl. *Switching of Single Letters*), **Náhodné premiešanie vnútornej časti slova** (angl. *Randomization of the Middle Part of the Word*) či **Kompletná náhodná zmena poradia znakov** (angl. *Complete Randomization of a Word*). Okrem toho experimentovali so **Zámenou znakov za susediace klávesy na klávesnici** (angl. *Keyboard Neighbor Substitution*), čo pomohlo zvýšiť odolnosť modelov voči typografickým chybám. Feng et al. (2020) [65] rozšírili tento prístup **Náhodným mazaním** (angl. *Random Deletion*), **Transpozíciou** (angl. *Transposition*) a **Vkladaním znakov** (angl. *Insertion of Characters*), pričom ignorovali prvý a posledný znak slova. Tieto metódy zlepšili diverzitu, plynulosť a zachovanie sentimentu textu a dosiahli lepšie výsledky v porovnaní so základnými metódami. V štúdii Dai et al. (2023) [66] boli testované viaceré techniky manipulácie so znakmi. Ich **Metóda augmentácie vložením znakov** (angl. *Insert Character Augmentation*), ktorá vkladá náhodné znaky do textu, sa ukázala ako efektívny spôsob na zvýšenie odolnosti modelu voči šumu, čím sa dosiahlo presnosti 82,6% (BERT) na datasete PubMed20K v porovnaní so základným modelom (79,2%). Podobne prístup **Náhodného prehodenia znakov** (angl. *Swap Character Augmentation*), ktorý simuloval preklepy náhodným prehodením dvoch znakov, priniesol presnosť 76,2% (BERT) na datasete Amazon v porovnaní so základným modelom (73,4%). V štúdii zmienili tiež metódu **Náhodné vymazanie** (angl. *Delete Character Augmentation*), ktorá náhodne odstraňovala znaky aby pomohla modelu lepšie ignorovať drobné chyby. Rozšírením oblasti augmentácie na úrovni znakov sa Dai et al. zaoberali aj **OCR augmentáciou** (angl. *OCR Augmentation*), ktorá simuluje chyby vznikajúce pri optickom rozpoznávaní znakov (OCR). Táto technika sa ukázala ako mimoriadne užitočná pri spracovaní medicínskych textov, kde presnosť

dosiahla 76,8% (BERT) v porovnaní so základným modelom (63,6%). Ukážku tohto typu transformácie spolu s ďalšími príkladmi znázorňuje tabuľka 2.1, ktorá vizuálne demonštruje, ako jednotlivé augmentačné techniky menia pôvodný text do alternatívnych foriem. V nedávnych štúdiách boli preskúmané aj **Fonetické substitúcie znakov** (angl. *Phonetic-Based Character Substitutions*), kde zmeny písmen odrážajú bežné fonetické chyby (napr. nahradenie „ph“ písmenom „f“). Tento prístup zvýšil variabilitu dát a pomohol zlepšiť generalizáciu modelu [63]. Nakoniec bola hodnotená aj **Pravopisná augmentácia** (angl. *Spelling Augmentation*), ktorá zámerne zavádzala pravopisné chyby. Hoci táto metóda posilnila robustnosť modelu, jej celkový prínos k presnosti bol relatívne nízky, dosahujúc iba 80,8% (BERT) v porovnaní so základným modelom (79,2%). Ebrahimi et al. (2018) [67] využili predtrénovaný model na vytváranie adversariálnych príkladov (angl. *Adversarial Examples*), pričom priamo menili písmená v dátach tak, aby zvýšili chybovosť predikcií modelu. Po následnom trénovaní na augmentovaných dátach sa chybovosť výrazne znížila a účinnosť adversariálnych útokov sa podstatne oslabila. Coulombe (2018) [68] sa venoval aj vplyvu **Zmeny veľkosti písmen** (angl. *Alteration of Upper and Lower Case*) a **Úpravy interpunkcie** (angl. *Modification of Punctuation*). Najvyššie absolútne zlepšenie presnosti dosiahlo hodnotu 2,5 % v porovnaní s najlepšie fungujúcou základnou metódou. Belinkov a Bisk (2018) [64] taktiež testovali **Prirodzený šum** (angl. *Natural Noise*), kde boli slová nahrádzané bežnými preklepmi na základe databáz pravopisných chýb. Hoci tento prístup zhoršil výkon prekladových modelov, pomohol zvýšiť odolnosť klasifikačných modelov voči reálnym preklepom. Coulombe (2018) [68] dosiahol dodatočné zlepšenie presnosti o 1,5 % pri aplikácii tejto metódy na XGBoost (najlepší základný model). V kontexte **Pravidlovo založených transformácií** (angl. *Rule-Based Transformations*) Coulombe (2018) [68] implementoval **Metódy založené na regulárnych výrazoch** (angl. *Regular Expression-Based Methods*), ktoré umožňovali systematické vkladanie pravopisných chýb, modifikáciu názvov entít a nahrádzanie skrátených tvarov slov (napr. „I am“ na „I'm“). Testovanie ukázalo, že pri aplikácii na model XGBoost (najlepší základný model) táto metóda viedla k dodatočnému zlepšeniu presnosti o 0,5 %.

Augmentácia	**Príklad**
Originálna veta	Kto druhému jamu kope, sám do nej spadne.
OCR augmentácia	Kto druhému jamu kop3, sám do nej spadne.
Prehodenie znakov	Kto druemhů jamu kope, sám do nej spadne.
Náhrada susedným znakom na klávesnici	Kto druhému jamu kppu, sám do nej spadne.
Kompletná zmena poradia znakov	Kto druhému jamu kope, sám do nej psdaen.
Premiešanie vnútornej časti slova	Kto druhému jmau kope, sám do nej spadne.
Náhodné vymazanie znakov	Kto druhému jamu kope, sám do ne spadne.
Vloženie náhodného znaku	Kto druhému jamu kople, sám do nej spadne.
Fonetická substitúcia	Kdo druhému jamu kope, sám do nej spadne.
Pravopisná chyba	Kto druhému jma kope, sám do nej spadne.
Zmena veľkosti písmen	Kto DRUHÉMU jamu kope, sám do nej spadne.
Úprava interpunkcie	Kto druhému jamu kope; sám do nej spadne.

Tab. 2.1: Ukážka použitia augmentačných metód na úrovni znakov.

2.2.2 Augmentácia na úrovni slov

Augmentačné metódy na úrovni slov zahŕňajú techniky zamerané na syntaktické a sémantické modifikácie textu. Xie et al. (2017) [69] aplikovali metódu **Unigramový šum** (angl. *Unigram Noising*), kde sa vybrané slová nahradili inými slovami s určitými pravdepodobnosťami, a metódu **Šum prázdnym znakom** (angl. *Blank Noising*), kde sa slová nahradili podčiarkovníkom. Ich experimenty preukázali zlepšenie klasifikačných modelov. Li et al. [70] rozšírili tento prístup o **Syntaktický šum** (napr. skracovanie viet, zmena prídavných mien) a **Sémantický šum** (nahrádzanie slov synonymami). Okrem toho použili **Word Dropout**, kde sa náhodne maskujú slová počas tréningu. Kombinácia týchto metód zlepšila presnosť až o 1.7 %. Wei a Zou (2019)[71] predstavili **Easy Data Augmentation (EDA)**, ktorá zahŕňa techniky **Náhodného vloženia** (angl. *Random Insertion*), **Náhodnej zámene** (angl. *Random Swap*) a **Náhodného vymazania** (angl. *Random Deletion*). Výsledky ukázali, že EDA dosahuje významné zlepšenia v klasifikácii pri práci s malými datasetmi, no v niektorých prípadoch môže znížiť presnosť, ak náhodné zmeny narušia význam textu. Ukážku vybraných techník augmentácie na úrovni slov možno vidieť v tabuľke 2.2, kde sú aplikované rôzne transformačné operácie na pôvodný text. **Redukcia funkčných slov** (angl. *Reduction of Function Words*) je metóda podobná náhodnému vymazaniu slov, no zameriava sa na odstraňovanie funkčných slov, ako sú predložky či spojky, pričom obsahové slová zostávajú zachované. Štúdia [72] ukázala, že táto technika zvýšila presnosť klasifikácie o 3,06 %. Rizos et al. (2019) [73] navrhli špecifickú metódu indukcie šumu pre neurónové siete, kde posúvali sekvencie v rámci paddingu, čím zvýšili výkon v detekcii nenávistných prejavov až o 5.8 % (Macro-F1). Sun a He [74] testovali metódu kde pridávali bezvýzna-

move slová na začiatok alebo koniec textu, hoci izolovaný vplyv tejto metódy nebol samostatne vyhodnotený, keďže autori kombinujú túto metódu s inými.

Augmentácia	**Príklad**
Originálna veta	My o vlku a vlk za dverami.
Unigramový šum	Hovoríme o vlku a vlk za dverami.
Šum prázdnym znakom	_ o vlku a vlk za dverami.
Náhodné vymazanie slova	My o a vlk za dverami.
Náhodné vloženie	My náhle o vlku a vlk za dverami.
Náhodná zámena slov	My o vlk a vlku za dverami.
Synonymická náhrada	My o šelme a vlk za dverami.
Redukcia funkčných slov	My vlku vlk za dverami.
Antonymická náhrada	My o baránkovi a vlk za dverami.
Kontextová náhrada slov (BERT)	My o vlku a vlk za rohom.
Embeddingová substitúcia	My o vlku a vlk za bránou.

Tab. 2.2: Ukážka použitia augmentačných metód na úrovni slov.

Xie et al. (2017) [69] testovali metódu nahrádzania neinformatívnych slov na základe TF-IDF, čím model lepšie ignoroval irelevantné slová. Podobne Choi et al. [75] maskovali kľúčové slová na generovanie kontrafaktuálnych príkladov, pričom táto metóda bola efektívna v kontrastívnom učení. Synonymická náhrada sa stala populárnou metódou textovej augmentácie, pričom výskumníci využívali rôzne slovníky, ako napríklad WordNet, alebo embeddingy slov. Zhang et al. (2015) [76] boli prví, ktorí použili tezaurus na rozšírenie údajov. Používajú tezaurus odvodený z WordNetu, ktorý triedi synonymá slov podľa ich podobnosti. V štúdii Li et al. (2022) [63] sa tieto metódy rozšírili o použitie WordNetu a VerbNetu na získanie synonymických náhrad, pričom synonymá boli vyberané na základe podobnosti v embedding priestore (napr. Glove alebo Word2Vec). V článku [72] autori analyzovali viacero techník augmentácie na úrovni slov a ich vplyv na presnosť klasifikácie textu. Jednou z nich je aj **Synonymická náhrada** (angl. *Synonym Replacement*) pri ktorej sú slová v texte nahrádzané synonymami z WordNetu. Experimenty ukázali, že aplikácia tejto metódy viedla k zvýšeniu presnosti klasifikácie pomocou SVC modelu o 2.85 %. Zaujímavou je aj druhá čiastková metóda EDA (angl. *Easy Data Augmentation*) od Wei a Zou (2019) [71], kde synonymá nenahrádzajú konkrétne slová, ale sú náhodne vložené do inštancie. Opačným prístupom k metóde synonymickej náhrady je **antonymická náhrada** (angl. *Antonym Replacement*), ktorá namiesto zachovania pôvodného významu slov v texte cielene vytvára významovo opačné verzie. Táto metóda bola v štúdii Zaiton a Alansary (2015) [77] použitá na generovanie významovo opačných verzií textu. Hoci zvyšuje diverzitu trénovacích dát, často spôsobuje zmenu

významu viet, čím sa znižuje kvalita augmentovaných údajov. Podobne ako pri náhrade synonymami, metódy založené na embeddingoch hľadajú slová, ktoré najlepšie zapadajú do textového kontextu a zároveň nemenia základný význam textu. Rizos et al. (2019) [73] tvrdia, že táto metóda podporuje model, aby kládol menší dôraz na priradenie jednotlivých slov k štítku a väčší dôraz na zachytenie podobných sekvenčných vzorov, teda kontextu nenávistných prejavov. Výhody tejto techniky v porovnaní so synonymickou náhradou sú v tom, že metódy založené na distribučnej hypotéze sú komplexnejšie a berú do úvahy kontext textu. To znamená, že nahradenia nie sú obmedzené databázou, ako je WordNet a že možno generovať gramaticky správnejšie vety. Wang a Yang (2015) [78] použili tento typ augmentácie na lepšiu klasifikáciu urážlivých tweetov. Pomocou algoritmu najbližšieho suseda (k-NN) identifikovali najvhodnejšie embeddingy ako náhrady slov v tréningových údajoch a dosiahli zlepšenie F1 skóre až o 2,4 bodu pri logistickej regresii. Hlavným problémom však je, že embeddingová náhrada nemusí vždy zachovať kontextuálny význam inštancií, čo môže viesť k skresleniu štítkov. Riešením je metóda „counter-fitting“ od Mrkšića et al. (2016) [79], ktorú použili Li et al. [63] na zosúladenie embeddingov tak, aby sa posilňovali podobnosti medzi synonymami a penalizovali podobnosti medzi antonymami. Alzantota et al. (2018) [80] ukázali, že generovanie podobných slov pomocou GloVe embeddingov a metódy „counter-fitting“ môže byť efektívne pri výbere slov, ktoré najlepšie zapadajú do daného kontextu. V článku [66] testovali metódu ***SubstituteWordByGoogleNewsEmbeddings***, kde sa slová nahrádzajú na základe podobnosti v embedding priestore a ***InsertWordByGoogleNewsEmbeddings*** náhodne vyberá slovo zo slovníka korpusu GoogleNews a vkladá ho do náhodnej pozície v texte. Lepšiu stabilitu zabezpečila technika ***CounterFittedEmbeddingAug***, ktorá vylepšuje embeddingy tak, aby synonymá boli v priestore bližšie k sebe, zatiaľ čo antonymá sa od seba vzďaľovali. Tento prístup zvýšil presnosť modelu na Amazon datasete na 75,4 % (BERT) a preukázal konzistentnú výkonnosť pri rôznych typoch textov v porovnaní so základným modelom (73,4%). Jazykové modely reprezentujú jazyk predpovedaním nasledujúcich alebo chýbajúcich slov na základe predchádzajúceho kontextu, čo je základom klasického a maskovaného jazykového modelovania. Tento prístup umožňuje modelom filtrovať nevhodné slová a generovať texty, ktoré sú gramaticky správne a kontextovo vhodné. S rozvojom metód, ako je BERT, sa nahrádzanie slov v textoch stalo ešte realistickejšie. Wu et al. (2018) [81] zaviedli model c-BERT, ktorý je kondicionovaný na labely, čo zabezpečuje správnosť označení aj pri náročných úlohách. Tento prístup viedol k výraznému zlepšeniu presnosti v scenároch s nízkymi dátovými zdrojmi. Avšak c-BERT má svoju nevýhodu, je fixný pri aplikácii, a v prípade malého množstva údajov môže augmentácia stratiť schopnosť zachovať správne štítky. Riešením tejto nevýhody sa ukázalo integrovanie c-BERT do schémy posilňovaného učenia (angl. *reinforcement*

learning), čo predložili Hu et al. (2019) [82]. Tento prístup spája klasické učenie pod dohľadom s jemným dolaďovaním jazykového modelu c-BERT, čím sa dosiahli oveľa lepšie výsledky v scenároch s nízkym objemom údajov, oproti pôvodnému c-BERT. Ďalší vývoj tejto metódy bol predstavený v práci [66], kde autori zaviedli metódu ***ContextualWordAugUsing***, ktorá využíva BERT, DistilBERT a RoBERTa na kontextuálne vkladanie alebo nahrádzanie slov. Týmto spôsobom sa modely stali ešte flexibilnejšími a schopnými generovať realistické texty s lepším zachovaním kontextu aj v situáciách s nízkym množstvom dát. Jiao et al. (2019) [83] používajú ako slová, tak aj maskované jazykové modely na získanie augmentovaných dát. Aplikujú tokenizer BERT na tokenizovanie slov do viacerých kusov slov. Každý kus slova je nahradený s pravdepodobnosťou 0,4. Ak kus slova nie je celé slovo (napríklad „est“), nahradia ho jeho K-najbližšími susedmi v priestore Glove embeddingov. Ak je kus slova celé slovo, autori ho nahradia [MASK] a použijú BERT na predpovedanie K slov na doplnenie medzery.

2.2.3 Augmentácia na úrovni viet

Na úrovni viet sú populárne metódy, ktoré menia gramatickú štruktúru alebo zachovávajú význam pri generovaní nových viet. Min et al. (2020) [84] ukázali, že **Inverzia** (angl. *Inversion*) a **Pasivizácia** (angl. *Passivation*) môžu zlepšiť generalizáciu modelov v úlohách prirodzeného jazyka. Tieto metódy pomohli modelom ako BERT lepšie extrahovať syntaktické informácie, ktoré inak zostali nevyužité pri nedostatku relevantných príkladov v datasetoch, ako je MNLI. **MixUp augmentácia** (angl. MixUp Augmentation) interpoluje medzi rôznymi vzorkami textu. Guo et al. (2019) [85] testovali MixUp na úrovni slov a viet, pričom výsledky ukázali, že významne redukuje overfitting a zlepšuje výkon modelov na klasifikačných úlohách. Feng et al. (2019) [86] navrhli **Sémantickú výmena textu** (angl. *Semantic Text Exchange*), ktorá selektívne mení frázy pri zachovaní sentimentu textu. Testy ukázali, že táto metóda mierne znižuje plynulosť a obsahovú konzistenciu, najmä pri práci s kratšími textami. Şahin a Steedman (2018) [87] zaviedli metódy **Orezania** (angl. *Cropping*) a **Rotácie** (angl. *Rotation*) pre nízkozdrojové jazyky. *Cropping* skracuje vety zameraním sa na subjekty a objekty, zatiaľ čo *Rotation* presúva flexibilné fragmenty viet. Hoci v angličtine tieto techniky môžu vytvárať šum, v nízkozdrojových jazykoch priniesli výrazné zlepšenie pri úlohách, ako je POS-tagging a závislostné parsovanie. Na to nadvezuje štúdia od Haralabopoulos et al. [88] kde pomocou permutácie slov vo vetách, kde poradie slov v rámci vety sa náhodne mení a tak zvýšuje robustnosť modelov tým, že ich naučí rozpoznávať význam vety nezávisle od presného slovosledu, zaznamenali nárast presnosti klasifikácie až o 4.1 %. Shi et al. (2021) [89] predstavili metódu **Náhrada podštruktúry** (angl. *Substructure Substitution*), ktorá nahrá-

dza subštruktúry textu (napr. frázy alebo sekvencie POS-tagov) medzi vzorkami s rovnakým labelom. Táto metóda takmer zdvojnásobila presnosť v nízkozdrojových úlohách, ako sú SST-2 a AG News, a prekonala dokonca aj pokročilé jazykové modely ako c-BERT. Kim et al. (2021) [90] použili **Lexikalizované gramatické stromy** (angl. *Lexicalized Grammatical Trees*), kde extrahovali stromové štruktúry viet a nahradzovali slová s rovnakými POS-tagmi zo vzoriek tej istej triedy. V prostredí few-shot a semi-supervised učenia táto metóda priniesla významné zlepšenia presnosti. Ukážku rôznych techník augmentácie na úrovni viet znázorňuje tabuľka 2.3, kde sú jednotlivé transformačné metódy aplikované na pôvodný text. Každá metóda demonštruje inú syntaktickú alebo sémantickú úpravu, ako napríklad zmenu slovosledu (inverzia, rotácia), zmenu gramatického tvaru (pasivizácia), interpoláciu medzi vetami (MixUp) či výmenu podštruktúr a fráz.

Augmentácia	**Príklad**
Originálna veta	Bez vetra sa ani lístok nepohne.
Inverzia	Ani lístok sa nepohne, bez vetra.
Pasivizácia	Lístok nie je pohnutý bez vetra.
MixUp augmentácia	Bez vetra sa lístok pohol v príjemnom vánku.
Orezanie	Bez vetra lístok nepohne.
Rotácia	Ani lístok sa nepohne bez vetra.
Permutácia slov	Vetrom bez ani sa lístok nepohne.
Lexikalizované gramatické stromy	Bez vetra sa lístok ani nepohne.

Tab. 2.3: Ukážka použitia augmentačných metód na úrovni viet.

2.2.4 Augmentácia na úrovni dokumentov

Metódy na úrovni dokumentov sa zameriavajú na manipuláciu s celými textovými blokmi, pričom jednou z najčastejšie používaných techník je **Round-trip Translation** (angl. *Obojsmerný preklad*). Táto metóda zahŕňa preklad dokumentu do iného jazyka a následný spätný preklad do pôvodného jazyka. Výskumy ukázali, že táto technika je užitočná pri generovaní parafráz a môže zlepšiť klasifikačné modely až o 5.8 % [91]. V článku [72] výsledky experimentov ukázali, že táto metóda bola najúčinnejšia pri použití SVC klasifikátora, pričom presnosť modelu sa zvýšila o 3.06 %. Dai et al. (2023) [66] pomocou ***BackTranslationAug*** augmentovali dataset Symptoms, kde táto metóda zlepšila presnosť na 77,8 % (BERT) v porovnaní so základným modelom (63,6%), no pri zložitejších textoch vznikali významové odchýlky. Táto metóda sa ukázala ako mimoriadne prínosná pri zachovaní lexikálno-sémantických vzťahov medzi slovami. V článku Gupta a Mahmood (2024) [92] bol testovaný s jazykmi ruština a nemčina. Ukázalo sa, že pre sentimentálnu klasifikáciu dosahovala verzia angličtina—ruština lepšie výsledky ako angličtina—nemčina. Xie

et al. (2020) [93] implementovali variant s náhodným vzorkovaním v rámci beam search algoritmu, čím zvýšili diverzitu generovaných vzoriek. Rôzne implementácie tejto techniky sa líšia v použitých jazykoch a filtrovaní výsledkov, ktoré sú kľúčové na minimalizovanie potenciálnych chýb spôsobených nepresnosťami v prekladoch [91]. Niektoré práce pridávajú dodatočné funkcie k základnému spätnému prekladu. Nugent et al. (2021) [94] skúmajú rôzne nastavenia teploty softmaxu s cieľom dosiahnuť väčšiu rozmanitosť, pričom zachovávajú význam textu. Qu et al. (2021) [95] spájajú spätný preklad s adversariálnym učením, čím vytvárajú rôznorodé a informatívne augmentované príklady prostredníctvom kombinácie viacerých transformačných metód.

2.2.5 Pokročilé metódy augmentácie

Okrem tradičných prístupov existujú aj pokročilé techniky, ako napríklad **Augmentácia v priestore vlastností** (angl. *Feature Space Augmentation*), ktorá transformuje reprezentácie v skrytých vrstvách modelov, čím umožňuje interpoláciu medzi existujúcimi vzorkami. Táto metóda sa ukázala ako efektívna pri zvyšovaní robustnosti klasifikačných modelov [96]. Jednou z techník je **indukcia šumu**, kde Kumar et al. (2019) [97] aplikovali náhodný multiplikatívny a aditívny šum na reprezentácie vlastností. Tento prístup mierne zlepšil klasifikačné výsledky v štandardnom nastavení, avšak v few-shot scenároch priniesol výrazné zlepšenia. Ďalšou metódou je **Linear Delta**, ktorá pridáva rozdiel medzi dvoma inštanciami k tretej z rovnakej triedy, čím rozširuje rozmanitosť dát a zvyšuje schopnosť modelu generalizovať. V rámci adverzariálneho trénovania sa využívajú techniky ako **PGD** [98] alebo **FreeLB** [99], ktoré zavádzajú malé perturbácie v priestore vlastností na vytváranie adverzariálnych príkladov. Zatiaľ čo klasické PGD je výpočtovo náročné, FreeLB akumuluje gradienty, čím výrazne zlepšuje výsledky bez zbytočného výpočtového *overheadu*. **Virtuálne adverzariálne trénovanie** (Miyato et al. [96]) maximalizuje KL divergenciu medzi predikciami s a bez perturbácií, čím zvyšuje robustnosť modelov aj bez použitia labelov. Jiang et al. (2021) [100] túto metódu rozšírili prístupom SMART, ktorý zabraňuje príliš agresívnym aktualizáciám modelu, čím stabilizuje učenie. Na úrovni jazykových modelov sa osvedčila **adverzariálna predtréningová optimalizácia**, ktorú implementovali Wang et al. (2019) [101] a Liu et al. (2020) [102]. Technika **ALUM** aplikovaná na modely ako RoBERTa viedla k lepšej generalizácii a zvýšenej odolnosti voči šumom. Okrem komplexných prístupov však existujú aj jednoduché a efektívne metódy. Shen et al. (2020) [103] navrhli techniky ako **Vynulovanie embeddingu slova** (angl. *token cutoff*), **Vynulovanie dimenzie embeddingu** (angl. *feature cutoff*) a **Vynulovanie súvislého segmentu slov** (angl. *span cutoff*). Napriek ich jednoduchosti tieto prístupy

v niektorých úlohách prekonali aj náročnejšie adverzariálne techniky. Ďalším prístupom je **Generatívna augmentácia** (angl. *Generative Augmentation*), pri ktorej sa využívajú generatívne modely, ako sú GPT a BERT, na tvorbu syntetických vzoriek na základe existujúcich údajov. Výskumy naznačujú, že táto technika môže výrazne zlepšiť presnosť v úlohách spracovania prirodzeného jazyka (NLP) [104]. Rôzne generatívne modely ponúkajú odlišné prístupy k tvorbe nových dát, pričom každý z nich prináša špecifické výhody. **Variational Autoencoders** (VAEs), ako ukázali Qiu et al. (2020) [105] a Malandrakis et al. (2019) [106], umožňujú generovať nové texty buď z prior alebo posterior distribúcie. Prior distribúcia vedie k väčšej diverzite syntetických vzoriek, zatiaľ čo posterior distribúcia vytvára texty bližšie k tréningovým dátam. **Conditional VAEs**, ktoré kombinujú generovanie s rekonštrukčnou úlohou, často prekonávajú klasické VAEs práve v schopnosti vyvážiť diverzitu a relevanciu generovaných textov. Ďalšou zaujímavou technikou je **NeuralEditor** a **Edit-transformer**, ktorú predstavili Guu et al. (2018) [107] a Raille et al. (2020) [108]. Tieto modely využívajú editovacie vektory na generovanie nových textov na základe lexikálnej podobnosti so vstupnými vzorkami, čo je obzvlášť užitočné pri jazykovom modelovaní a pri transfer learningu medzi rôznymi doménami. **Recurrent Neural Networks** (RNNs) a **LSTM-CNN** prístupy testovali Rizos et al. (2019) [73] a Ollagnier & Williams (2020) [109]. Aj keď RNN-based generovanie prinieslo slabšie výsledky v porovnaní s modernejšími architektúrami, rozdelenie textu na menšie segmenty a generovanie nových viet pomohlo zvýšiť diverzitu dát, čo následne prispelo k lepšej generalizácii modelov. Generatívne adversariálne siete (GANs) sa tiež ukázali ako účinný nástroj. Sun a He (2020) [74] navrhli **seqGAN**, kde generátor a diskriminátor spolupracujú cez reinforcement learning, aby iteratívne zlepšovali kvalitu generovaných textov. Li et al. (2018) [110] tento prístup rozšírili technikou **CS-GAN**, ktorá pridáva klasifikátor zabezpečujúci zhodu medzi generovaným textom a cieľovou triedou, čím sa zlepšuje presnosť modelu pri klasifikačných úlohách. Výrazný pokrok priniesli aj veľké jazykové modely, ako GPT-2 a GPT-3. Wang a Lillis (2019) [111], Anaby-Tavor et al. (2019) [104] a Yoo et al. (2021) [112] využili tieto modely na generovanie kompletných textových inštancií. **Fine-tuning GPT-2** v kombinácii so starostlivou selekciou relevantných príkladov výrazne zlepšil presnosť v režimoch s nízkym počtom tréningových dát. Ešte silnejšie výsledky dosiahli výskumníci s GPT-3, ktorý vďaka prompt engineeringu a pseudo-labelingu dokázal prekonať viaceré tradičné augmentačné techniky, a to bez potreby rozsiahleho re-trénovania. Napokon, **slice-based generovanie** predstavili Lee et al. (2021) [113], ktorí rozdelili dáta na podskupiny a generovali nové inštancie špecificky pre underrepresented slices. Táto metóda výrazne zlepšila výsledky v úlohách textovej klasifikácie, intent recognition a relation extraction, pretože model získal lepšiu reprezentáciu aj menej frekventovaných vzoriek. Doi et al. (2023) [66]

testovali metodu augmentovania pomocou ChatGPT. Metóda **AugGPT** využíva ChatGPT na refrázovanie každej vety v tréningovej množine do šiestich semanticky podobných, no formuláciou odlišných viet, čím zvyšuje variabilitu dát bez straty významu. Po augmentácii sa model BERT doladí na kombinácii pôvodných a syntetických vzoriek, pričom kontrastná strata pomáha zlepšiť separáciu tried. Táto technika výrazne zlepšila presnosť v few-shot textovej klasifikácii, na datasete Amazon presnosť vzrástla z 73,4 % na 81,6 %, pri symptómoch z 63,6 % na 88,9 % a na PubMed20K z 79,2 % na 83,5 %. Analýza latentných reprezentácií navyše ukázala, že generované vzorky sú nielen podobné reálnym dátam, ale aj lepšie organizované pre efektívne učenie modelu. Vďaka schopnosti ChatGPT vytvárať kvalitné a rôznorodé texty bez potreby manuálnej anotácie predstavuje AugGPT jednoduché a výkonné riešenie na rozšírenie dát v NLP úlohách.

2.2.6 Hybridné prístupy

Nakoniec, **Hybridné prístupy** (angl. *Hybrid Approaches*) kombinujú viaceré augmentačné metódy, napríklad spájanie obojsmerného prekladu so synonymickou náhradou, čím sa dosahuje vyššia diverzita generovaných dát a stabilnejší výkon modelov [114]. Okrem toho môžu byť hybridné prístupy založené na kombinovaní viacerých parafrázovacích techník, čím sa ešte viac rozširuje variabilita textu. Napríklad Liu et al. (2020) [115] kombinovali synonymické slovníky so sémantickými embeddingmi, zatiaľ čo Jiao et al. [83] použili kombináciu embeddingov a maskovaných jazykových modelov. V oblasti šumových metód sú často kombinované jednoduché a neparametrické techniky, ako ukázali Peng et al. [116], kde sa aplikovali viaceré druhy šumu na ten istý text na zlepšenie robustnosti modelov. Rovnako Regina et al. (2021) [117] a Xie et al. (2017) [69] ukázali, že kombinácia rôznych zdrojov šumu alebo parafrázovania dokáže zlepšiť generalizáciu modelov na reálne dáta. **Neriadené metódy** (angl. *Unsupervised Methods*) často kombinujú viaceré jednoduché techniky, ktoré fungujú nezávisle od konkrétnej úlohy. Wei a Zou [71] vytvorili rámec EDA (angl. *Easy Data Augmentation*), ktorý zahŕňa náhradu synonymami, náhodné vkladanie, prehadzovanie a vymazávanie slov. Táto kombinácia bola úspešne použitá v mnohých úlohách, ako ukázali Longpre et al. (2020) [118] a Rastogi et al. (2020) [119]. Podobne Xie et al. (2020) [93] v rámci metódy UDA (angl. *Unsupervised Data Augmentation*) kombinovali spätný preklad s neriadenými šumovými technikami, čím dosiahli lepšiu diverzitu trénovacích dát. Ďalším efektívnym prístupom je **Viacúrovňová augmentácia** (angl. *Multi-Granularity Augmentation*), kde sa tá istá metóda aplikuje na rôznych úrovniach textu. Wang a Yang (2015) [78] trénovali súčasne slovné a rámcové embeddingy pomocou Word2Vec. Guo et al. (2019) [85] aplikovali techniku Mixup nielen na úrovni jednotlivých slov, ale aj na celých ve-

tách. Yu et al. (2019) [120] kombinovali sériu šumových metód na úrovni slov aj viet, čím dosiahli zvýšenie robustnosti modelov na rôzne druhy jazykových variácií. Taktiež, efektivita hybridných metód závisí aj od správneho nastavenia optimalizačných techník. Niektoré metódy, ako *oversampling* pôvodných dát alebo predtrénovanie na augmentovaných vzorkách, môžu výrazne ovplyvniť kvalitu modelu. Okrem toho hyperparametre, ako počet aplikovaných transformácií či pravdepodobnosť aplikácie jednotlivých metód, hrajú kľúčovú úlohu pri dosahovaní vyváženého zlepšenia výkonu modelu.

2.3 Záver analýzy textovej augmentácie

Analýza ukázala, že techniky textovej augmentácie predstavujú efektívny spôsob, ako zvýšiť rozmanitosť a objem logových dát určených pre trénovanie modelov umelej inteligencie. Z analýzy vyplýva, že viaceré techniky popísané v jednotlivých úrovniach augmentácie sú aplikovateľné aj v rámci spracovania bezpečnostných logov. Na úrovni znakov sú užitočné najmä jednoduché šumové techniky, ako vkladanie, mazanie či výmena znakov, ktoré môžu imitovať bežné chyby pri zadávaní údajov. Takéto prístupy môžu zvýšiť robustnosť modelov voči drobným odchýlkam vo formáte dát. Augmentácia na úrovni slov umožňuje meniť štruktúru a význam textu bez jeho skreslenia. Popri náhrade slov synonymami alebo vkladaní náhodných prvkov boli identifikované aj techniky ako redukcia funkčných slov, náhodné vymazanie, náhodná zámena a náhodné vloženie. Tieto prístupy sa ukazujú ako obzvlášť vhodné pri práci s neštruktúrovanými časťami logov, ako sú popisy udalostí. V prípade augmentácie na úrovni viet sa využívajú transformácie poradia slov alebo vetných konštrukcií, čo je zaujímavé najmä pri logoch s dlhšími textovými hláškami. Pri augmentácii na úrovni dokumentov sa ukázala ako efektívna technika round-trip translation preklad dokumentu do cudzieho jazyka a späť, ktorá napomáha generovaniu prirodzene znejúcich parafráz. V kontexte bezpečnostných logov sa však tento prístup ukazuje ako menej vhodný, keďže logy majú často štruktúrovaný alebo pološtruktúrovaný formát a preklad by mohol narušiť ich integritu. Z pohľadu praktickej aplikácie sú mimoriadne prínosné hybridné prístupy, ktoré kombinujú techniky z viacerých úrovní (znaky, slová, vety, dokumenty). Tieto metódy dokážu výrazne zvyšiť variabilitu dát, zlepšiť generalizáciu modelov a zároveň zachovať logickú konzistenciu v syntetických záznamoch. Prehľad jednotlivých tabuliek podľa úrovne aplikácie daných augmentačných metód, ich výsledkov a štúdií v ktorých boli zmienené sa nachádzajú v kapitole 5.1.

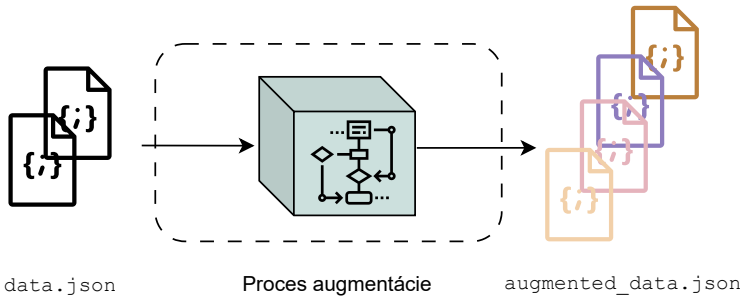
3 Návrh nástroja na rozširovanie logových záznamov

Pred samotným vytváraním augmentačného nástroje je potrebné spraviť analýzu možných nástrojov na generovanie jednotlivých entit logu a vybrať vhodné augmentačné techniky ktoré boli popísané v podkapitole 2.2. Použitý návrh by mal spĺňovať určité kritéria. Nástroj by mal podporovať JSON vstup so štruktúrovaným formátom logov. Na základe vstupu by sa mali vybrať jednotliví poskytovatelia vyskytujúci sa pre enitity vo vstupnom logu. Následne by sa mali aplikovať vhodné augmentačné metódy pre vybrané entity a nahradiť pôvodné hodnoty. Výstupom by mal byť novo vytvorený dataset logov. V nasledujúcej kapitole bude opísaný návrh nástroja, jeho štruktúry kódu, výber modernej knižnice na generovanie syntetických dát, následne zvolenie prístupu generovania dát z textových súborov a návrh čiastkových funkcií využívajúcich textovú augmentáciu.

3.1 Návrh štruktúry kódu

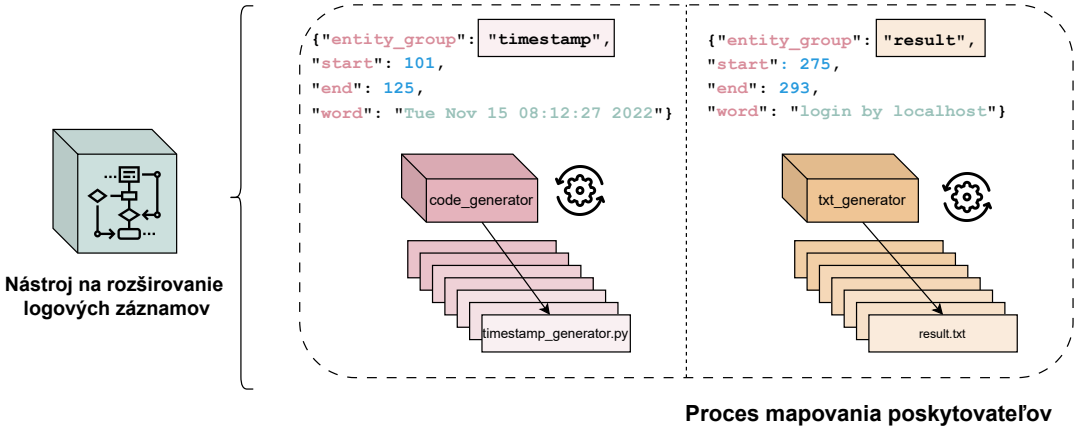
Pre správnu aplikáciu novo vygenerovaných dát je potrebné mať správne navrhnutú štruktúru kódu.

Podpora pre spracovanie vstupných súborov – Kód umožňuje načítanie logov z externého súboru, ktorý sa následne spracuje a augmentuje podľa požiadaviek. Po augmentácii sa vygenerované dáta ukladajú do nového súboru. Týmto spôsobom je možné automatizovane spracovávať veľké množstvo logov. Schematické znázornenie procesu augmentácie logových dát je možné vidieť na obrázku 3.1, vstupné JSON súbory sú spracované nástrojom, ktorý generuje rozšírený dataset vo forme nových JSON súborov.



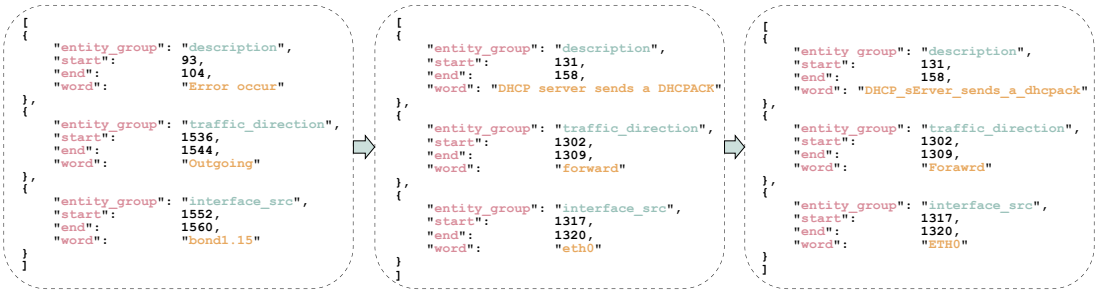
Obr. 3.1: Proces augmentácie logových dát zo vstupných JSON súborov.

Mapovanie entít na konkrétne funkcie – Každá entita je mapovaná na špecifickú funkciu, ktorá generuje nové syntetické dáta. To zaručuje, že každý generovaný log bude obsahovať relevantné a správne údaje, čím sa udržiava konzistencia v štruktúre dát. Tento mechanizmus sa využíva jak pri `Faker` poskytovateľoch tak aj pri textových poskytovateľoch, ktorý umožňuje načítať zoznamy zo súborov a na ich základe generovať dáta, ktoré sú následne použité v augmentácii. Na obrázku 3.2 je znázornený proces mapovacích funkcií na konkrétnych poskytovateľov (napr. `timestamp_generator.py` či `result.txt`).



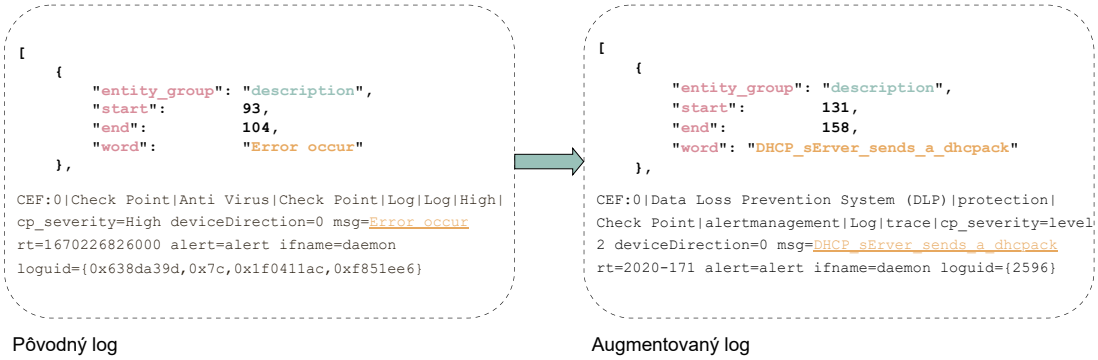
Obr. 3.2: Proces mapovania entít na generátory pri augmentácii logových dát.

Aplikovanie augmentácie – Kľúčová časť v systéme, ktorá vykonáva augmentáciu generovaných dát. Táto funkcia využíva rôzne techniky augmentácie, ktoré sú definované v konfigurácii nástroja. Používateľ môže prispôsobiť, ktoré techniky sa majú použiť, aby sa dosiahla požadovaná zmena v dátach. Použijú sa iba tie, ktoré sú povolené vo *whiteliste* pre konkrétny typ poskytovateľa, čo umožňuje flexibilné prispôsobenie procesu augmentácie pre rôzne typy dát. Tento mechanizmus zaručuje, že sa aplikujú len tie metódy, ktoré sú kompatibilné s typom dát generovaných daným poskytovateľom. Na obrázku 3.3 je znázornený postupný proces transformácie entít z pôvodných dát, kde sú najskôr vygenerované nové dáta pomocou generátorov, následne modifikované prostredníctvom zvolených augmentačných techník.



Obr. 3.3: Transformácia entít počas aplikovania augmentácie.

Zachovanie štruktúry a významu pôvodného textu – Dôležitým aspektom augmentácie je zachovanie pozícií začiatku a konca entít v texte, čo zabezpečuje, že štruktúra a význam pôvodného textu nie sú porušené. Na obrázku 3.4 je ukázaný príklad pôvodného a augmentovaného logu, pričom je zrejmé, že sa zmenil obsah entít aj samotný text, no štruktúra a pozície entít zostali správne aktualizované.

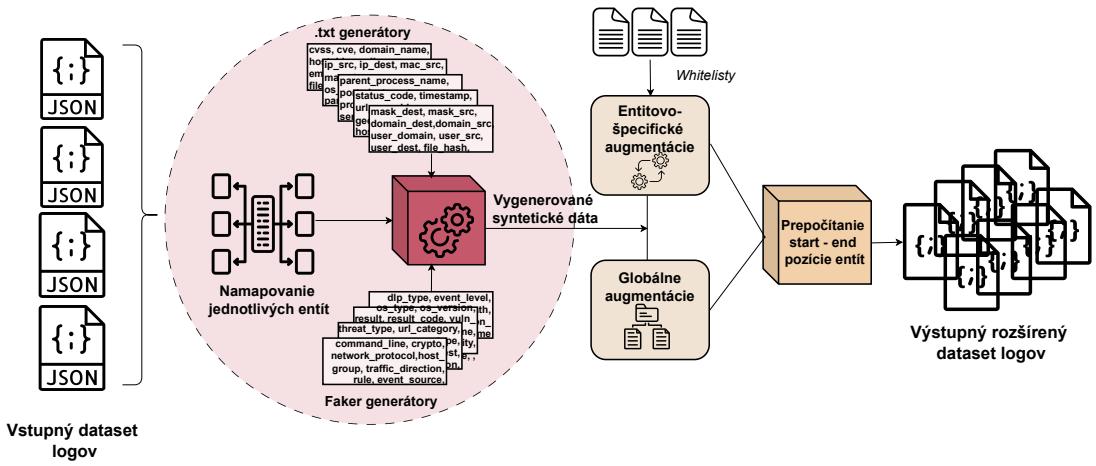


Obr. 3.4: Zachovanie štruktúry a aktualizácia entít počas augmentácie.

Logovanie a testovanie – Kód bude využívať logovanie na zaznamenávanie dôležitých informácií počas behu programu. Tento mechanizmus pomáha pri diagnostikovaní a sledovaní priebehu augmentácie, a to najmä pri testovaní s lokálnymi dátami. Kód tiež poskytuje možnosť testovania nástroja pomocou testovacieho režimu, ktorý pracuje s preddefinovanými dátami.

Navrhnutý prístup je vizuálne znázornený na obrázku 3.5, ktorý prezentuje architektúru celého nástroja na rozširovanie logových dát. Diagram zachytáva tok spracovania dát od vstupného datasetu vo formáte JSON, cez mapovanie jednotlivých entít na konkrétnych poskytovateľov (generátory), až po fázu augmentácie. Architektúra rozlišuje medzi entitami generovanými prostredníctvom knižnice `Faker` a tými, ktoré sú odvodené zo zoznamov uložených v textových súboroch (`.txt`). Po syntéze nových údajov prebieha ich augmentácia v dvoch úrovniach, na úrovni jednotlivých entít a globálnej štruktúry. Výstupom celého procesu je rozšírený dataset

logov, pričom sa zachováva správne prepočítaná pozícia každej entity, čím sa zabezpečuje konzistentná integrita dát.



Obr. 3.5: Návrh nástroja na rozširovanie logových dát.

3.2 Určenie prístupu a použitých nástrojov

Pri generovaní nových entít logov je možné vybrať z rôznych knižníc v Pythone, ako sú `Mimesis`, `FauxFactory` a `Faker`[1]. Tieto knižnice ponúkajú rôzne možnosti generovania dát, ktoré môžu byť využité pri augmentácii logov. Výber vhodnej knižnice závisí od špecifických potrieb projektu, ako je množstvo a typ generovaných dát, flexibilita a implementácia. V nasledujúcej podkapitole sa podrobne porovnajú vlastnosti jednotlivých knižníc, ktoré sú k dispozícii pre tento účel.

3.2.1 Porovnanie knižníc: `Faker`, `Mimesis`, `FauxFactory`

- **`Faker`** – `Faker` je knižnica pre generovanie realistických falošných dát, ktorá ponúka širokú škálu možností, ako sú mená, IP adresy, telefónne čísla, e-mailové adresy a mnoho ďalších údajov. Jej hlavnou výhodou je rozsiahla dokumentácia, ktorá výrazne zjednodušuje implementáciu a podporu.
  Významnou prednosťou knižnice je aj schopnosť generovať rozsiahle množstvo rôznych druhov údajov, čo umožňuje jej univerzálne využitie. Ďalšou výhodou je ľahká implementácia, čo predstavuje dôležitý faktor pri jej použití v projekte. Flexibilita `Fakera` umožňuje vytvárať presné dátové štruktúry prispôsobené konkrétnym potrebám [121].

[1]Dokumentáciu k jednotlivým knižniciam je možné nájsť na adresách `Mimesis`: `https://mimesis.name/master/` `FauxFactory`: `https://fauxfactory.readthedocs.io/en/latest/index.html` a `Faker`: `https://faker.readthedocs.io/en/master/providers.html`

- **`Mimesis`** – `Mimesis` je Python knižnica, ktorá je navrhnutá na generovanie falošných údajov pre rôzne aplikácie, ako je vyplňovanie databáz alebo tvorba realistických falošných údajov používateľov. Ponúka množstvo poskytovateľov tried pre rôzne typy dát, vrátane osobných údajov, emailových adries a ďalších [122]. Pri analýze knižnice `Mimesis` sa zistilo, že ponúka širokú škálu funkcií, podobne ako `Faker` a pokrýva rôzne oblasti generovania dát. Avšak, v rámci projektu sa ukázalo, že možnosti `Mimesis`, hoci rozsiahle, nie sú tak prispôsobiteľné a flexibilné ako v prípade `Fakera`. `Mimesis` síce disponuje množstvom rôznych dát, ale jej využiteľnosť v kontexte štruktúrovaných logových entít je obmedzená v porovnaní s `Fakerom`, ktorý lepšie spĺňa požiadavky na prispôsobenie dát pre špecifické logové štruktúry.
- **`FauxFactory`** – `FauxFactory` je knižnica určená predovšetkým na generovanie náhodných používateľských údajov, ako sú mená, e-mailové adresy a ďalšie jednoduché dáta. Hoci poskytuje základné funkcie pre generovanie údajov, jej rozsah a flexibilita sú obmedzené v porovnaní s `Mimesis` a `Faker`. `FauxFactory` je vhodná pre jednoduché testovacie účely, kde nie je potrebné generovať komplexné štruktúrované dáta. Knižnica neponúka tak širokú škálu funkcií ako `Faker` a `Mimesis`, čo ju robí menej vhodnou pre projekty, ktoré potrebujú širokú paletu rôznych dátových entít [123].

Na základe analýzy všetkých troch knižníc (`Faker, Mimesis, FauxFactory`) sa ukazuje, že `Faker` je najvhodnejšou voľbou pre generovanie logových entít v tomto projekte. Jeho silné stránky zahŕňajú širokú podporu pre generovanie rôznych typov štruktúrovaných dát, ako sú IP adresy, názvy hostiteľov, procesy, časové pečiatky a ďalšie. Okrem toho ponúka jednoduchú implementáciu a rozsiahlu dokumentáciu, čo výrazne zjednodušuje jeho používanie pri rozširovaní logov. Projekt vyžaduje generovanie 75 poskytovateľov entít logov, kde viaceré entity vyžadujú rovnaký druh dát. Po analýze knižnice `Faker` a ostatných knižníc sa dospelo k záveru, že `Faker` je najvhodnejšia knižnica na použitie pomocou ktorej sa dokáže generovať 46 druhov entít logov.

3.2.2 Generovanie dát pomocou Python knižnice

Na účely generovania testovacích dát v rámci tohto projektu je zvolená knižnica `Faker`, ktorá poskytuje rozsiahle množstvo poskytovateľov určených na tvorbu syntetických údajov. Knižnica `Faker` zahŕňa až 46 rôznych poskytovateľov v rámci nášho projektu umožňujúcich generovanie rôznorodých dátových typov, medzi ktoré patria mená, adresy, telefónne čísla, IP adresy, dátumy a ďalšie typy údajov potrebné pre komplexné logové záznamy. Pri každej entite si musíme zároveň položiť otázku z tabuľky 3.1 s metadátovými kľúčmi, ktorá definuje význam a kontext danej entity

v reálnych logovacích systémoch. Tieto otázky nám slúžia ako referencia pre výber zodpovedajúcich typov údajov, aby vygenerované hodnoty korešpondovali so skutočnými entitami pozorovanými v reálnom svete. Táto metodika zaručuje vysokú mieru konzistencie, relevantnosti a realizmu generovaných údajov, čo je kľúčové pre efektívne testovanie a validáciu logovacích procesov.

3.2.3 Generovanie dát z textových súborov

Vzhľadom na skutočnosť, že nie všetky typy údajov potrebných pre logové záznamy je možné generovať priamo pomocou knižnice `Faker`, je nutné pristúpiť k využitiu alternatívnych metód extrakcie dát. V prípadoch, kde knižnica neposkytuje špecializovaného poskytovateľa pre konkrétne kategórie údajov, budú tieto údaje získané z predpripravených textových súborov vo formáte `.txt`. Medzi primárne zdroje dát patria anonynimizované logové datasety získané v spolupráci s vedúcim, odborné a štandardizované databázy, predovšetkým databáza `MITRE ATT&CK`[2]. Z tejto databázy sa budú extrahovať špecifické entity ako napríklad `technique`, `threat actor` a `threat type`. Rovnako ako pri údajoch generovaných pomocou knižnice `Faker`, aj v prípade entít pochádzajúcich z textových súborov budeme pri každej entite uvažovať nad otázkami z tabuľky 3.2. Tieto otázky nám pomáhajú určiť význam, kontext a vhodnosť danej entity v rámci logovacích procesov. Vďaka tomu bude zabezpečené, že extrahované údaje sú nie len technicky korektné, ale aj kontextovo relevantné vo vzťahu k modelovaným hrozbám a udalostiam. Na doplnenie údajov je využitá aj umelá inteligencia (AI), konkrétne jazykový model GPT-4 [124], ktorý umožňuje syntetizovať alebo navrhovať ďalšie relevantné hodnoty entít založené na definovaných špecifikáciách projektu. Použitie AI prináša významné rozšírenie možností generovania autentických údajov, ktoré realisticky reflektujú vzory správania v rámci logových záznamov.

[2] `MITRE ATT&CK` predstavuje otvorenú a široko akceptovanú databázu taktických a technických detailov známych kybernetických útokov, poskytujúcu systematickú podporu pri detekcii, analýze a prevencii kybernetických hrozieb. Bližšie informácie sú dostupné na: `https://attack.mitre.org/`

Metakľúč	Otázka	Popis
result_code	Aký je konkrétny kód, ktorý zodpovedá výsledku udalosti?	Poskytuje číselnú alebo alfanumerickú hodnotu, ktorá podrobnejšie opisuje výsledok udalosti.
event_id	Aká konkrétna akcia alebo udalosť je zaznamenaná v tomto logu?	Jedinečný identifikátor priradený typu udalosti v logu systému.
timestamp	Aký je časový údaj tejto udalosti?	Čas, kedy sa udalosť odohrala, slúži na časové zoradenie udalostí.
user_dest	Aký je cieľový používateľský účet ovplyvnený touto udalosťou?	Používateľ, ktorý bol cieľom udalosti (prístup, útok, manipulácia).
user_domain	V akej doméne sa nachádza používateľský účet?	Doména, ku ktorej používateľský účet patrí (napr. Active Directory).
user_id	Aký je jedinečný identifikátor používateľského účtu?	Unikátne ID priradené používateľovi v systéme.
user_name	Aké je meno používateľa zapojeného do udalosti?	Zobrazované meno alebo prihlasovacie meno používateľa.
email_dest	Aká je cieľová e-mailová adresa v tejto udalosti?	E-mailová adresa príjemcu, cieľ udalosti (napr. phishing, únik dát).
email_src	Aká je zdrojová e-mailová adresa v tejto udalosti?	E-mail odosielateľa, často použitý na analýzu pôvodu (spam, útok).
host_dest	Aký je cieľový hostiteľ v tejto udalosti?	Zariadenie, ktoré bolo cieľom komunikácie alebo útoku.
host_id	Aké je jedinečné ID cieľového hostiteľa?	Unikátny identifikátor zariadenia v sieti alebo systéme.
host_name	Aký je názov cieľového hostiteľa?	Názov hostiteľa alebo zariadenia v sieti.
host_src	Aký je zdrojový hostiteľ udalosti?	Hostiteľ, z ktorého bola udalosť spustená.
os_build	Aké je číslo buildu operačného systému?	Build verzia operačného systému vrátane opráv a aktualizácií.
file_hash	Aký je kryptografický hash súboru?	Hash súboru (napr. MD5, SHA-1) na overenie integrity alebo detekciu známeho súboru.
file_name	Aký je názov súboru spojeného s touto udalosťou?	Názov súboru, ktorý bol použitý alebo detegovaný.
file_path	Aká je úplná cesta k tomuto súboru?	Celá cesta v súborovom systéme k danému súboru.
parent_process_id	Aké je ID nadradeného procesu?	ID procesu, ktorý spustil (vytvoril) iný proces.
parent_process_name	Aký je názov nadradeného procesu?	Názov aplikácie alebo procesu, ktorý vytvoril podriadený proces.
process_id	Aké je ID spusteného procesu?	Unikátne ID daného procesu v systéme.
process_name	Aký je názov procesu vykonaného v tejto udalosti?	Názov procesu, ktorý bol spustený.
service_id	Aký je identifikátor služby zapojenej do udalosti?	Jedinečný identifikátor systémovej služby.
session_id	Aký je identifikátor používateľskej relácie?	ID relácie používateľa alebo systému.
domain_dest	Aká je cieľová doména?	Doména cieľového zariadenia alebo servera.
domain_src	Aká je zdrojová doména?	Doména odosielateľa alebo zdrojového zariadenia.
geoip_dest	Aká je geografická poloha cieľovej IP adresy?	Fyzická lokalita podľa IP adresy cieľa.
geoip_src	Aká je geografická poloha zdrojovej IP adresy?	Fyzická lokalita podľa zdrojovej IP adresy.
interface_src	Z ktorého sieťového rozhrania bola komunikácia odoslaná?	Fyzické alebo logické rozhranie na odosielajúcom zariadení.
device_ip	Aká je IP adresa zariadenia, ktoré vytvorilo tento log?	IP adresa zariadenia, ktoré zaznamenalo alebo odoslalo log.
interface_dest	Ktoré sieťové rozhranie prijalo túto komunikáciu?	Rozhranie cieľového zariadenia, ktoré prijalo dátový tok.
ip_dest	Aká je cieľová IP adresa hostiteľa?	IP adresa cieľového zariadenia.
ip_src	Aká je zdrojová IP adresa hostiteľa?	IP adresa pôvodcu komunikácie.
mac_dest	Aká je MAC adresa zariadenia, ktoré prijalo komunikáciu?	Fyzická adresa cieľového sieťového zariadenia.
mac_src	Aká je MAC adresa zariadenia, ktoré komunikáciu odoslalo?	MAC adresa odosielajúceho zariadenia.
mask_dest	Aká je maska podsiete pre cieľovú IP adresu?	Maska siete určujúca rozsah IP adresy cieľa.
mask_src	Aká je maska podsiete pre zdrojovú IP adresu?	Maska siete zdrojovej IP adresy.
port_dest	Aké je cieľové číslo portu?	Port na cieľovom zariadení (napr. 80, 443).
port_src	Aké je číslo zdrojového portu?	Port na odosielajúcom zariadení.
url	Ktoré URL bolo navštívené alebo spomenuté?	Navštívená alebo spracovaná URL adresa.
cve	Ktoré CVE identifikátory sú v udalosti spomenuté?	Identifikátor známej zraniteľnosti podľa MITRE (napr. CVE-2021-34527).
cvss	Aké CVSS skóre je priradené tejto zraniteľnosti?	Číselné skóre vyjadrujúce závažnosť zraniteľnosti (0.0 – 10.0).
status_code	Aký je HTTP stavový kód alebo výsledok operácie?	Číselný kód, ktorý udáva stav požiadavky, spracovania alebo odpovede systému (napr. HTTP 200, 404).
domain_name	Aký je názov domény?	FQDN alebo iný formálny názov domény spojenej s entitou alebo udalosťou.
user_src	Aké je meno používateľa, ktorý akciu inicioval?	Používateľský účet, ktorý spustil operáciu alebo bol jej zdrojom.
name	Aké je meno používateľa alebo entity spomenutej v udalosti?	Celé meno alebo identifikátor osoby, organizácie, zariadenia alebo inej entity.
hash	Aký je hash reťazec identifikujúci obsah?	Kryptografický odtlačok (napr. SHA-256) používaný na overenie integrity objektu alebo súboru.

Tab. 3.1: Prehľad metadátových kľúčov typu pre poskytovateľov typu `Faker`.

Metakľúč	Otázka	Popis
action	Aká konkrétna operácia alebo aktivita bola vykonaná?	Popisuje konkrétnu operáciu alebo aktivitu, ktorá bola vykonaná počas udalosti. Vyjadruje, čo bolo pokusom alebo čo sa v systéme spustilo.
command_line	Aký príkaz bol spustený na začatie procesu?	Udáva presný príkaz alebo shellový príkaz, ktorý bol použitý na spustenie procesu alebo aplikácie.
crypto	Aký kryptografický algoritmus alebo primitíva sa používa alebo je spomenutá?	Označuje kryptografický algoritmus alebo primitívu použitú počas udalosti. Môže ísť o šifrovanie, hashovanie alebo asymetrickú kryptografiu.
description	Aký ďalší kontext alebo informácia vysvetľuje túto udalosť?	Poskytuje detailný opis alebo zhrnutie udalosti s doplňujúcim kontextom.
dlp_type	Aký typ DLP klasifikácie bol priradený dokumentu alebo dátam?	Označuje klasifikáciu citlivosti dát podľa DLP systému.
event_level	Aká je závažnosť tejto udalosti?	Určuje úroveň dôležitosti udalosti (napr. Info, Chyba, Varovanie).
event_source	Odkiaľ táto udalosť pochádza?	Určuje systém alebo komponent, z ktorého udalosť pochádza (napr. Firewall, Windows).
event_type	Aký typ akcie alebo incidentu sa udial?	Kategorizuje typ udalosti (napr. Audit, Systémová udalosť).
host_group	Do akej skupiny patrí tento cieľový hostiteľ?	Skupina alebo kategória, do ktorej patrí hostiteľ.
host_platform	Aký typ platformy používa tento hostiteľ?	Typ alebo OS platformy, kde udalosť prebehla.
network_protocol	Aký sieťový protokol bol použitý?	Vyššia sieťová vrstva alebo protokol použitý počas komunikácie.
os_architecture	Aká je architektúra operačného systému hostiteľa?	Architektúra OS (napr. x86, x64, ARM).
os_name	Aký je názov operačného systému?	Názov OS inštalovaného na hostiteľovi (napr. Windows, Linux).
os_type	Aký typ operačného systému je použitý?	Typ OS podľa rodiny alebo distribúcie.
os_version	Aká je verzia operačného systému?	Verzia OS vrátane minoritných čísel a buildu.
result	Aký bol výsledok operácie alebo akcie?	Výsledok vykonanej akcie – napr. úspech alebo zlyhanie.
rule	Aké pravidlo alebo identifikátor spustilo túto udalosť?	Identifikátor politiky alebo pravidla, ktoré udalosť spustili.
service_name	Aký je názov služby zapojenej do udalosti?	Názov systémovej služby, ktorá bola do udalosti zapojená.
severity	Ako naliehavo treba na túto udalosť reagovať?	Úroveň dopadu alebo rizika spojenej s udalosťou.
tactic	S ktorou MITRE ATT&CK taktikou je táto udalosť spojená?	MITRE ATT&CK taktika (napr. Eskalácia privilégií).
tags	Aké štítky sú s touto udalosťou spojené?	Štítky priradené k udalosti na kategorizáciu.
technique	Aká technika alebo ID techniky MITRE je tu reprezentovaná?	MITRE ATT&CK technika (napr. skenovanie, získanie identity).
threat_actor	Ktorý útočník alebo skupina je s touto udalosťou spojená?	Identifikovaný útočník alebo APT skupina.
threat_type	Aký typ kybernetickej hrozby je popísaný alebo detegovaný?	Typ hrozby ako phishing, malware, ransomware.
url_category	Ako je kategorizované toto URL?	Kategória URL podľa rizika alebo účelu (napr. sociálne siete).
user_group	Do akej skupiny patrí tento používateľský účet?	Skupina používateľa (napr. AD skupiny).
user_type	Aký typ používateľa je zapojený do udalosti?	Typ účtu (napr. administrátor, hosť).
vuln_name	Aký je názov zraniteľnosti v udalosti?	Názov známej zraniteľnosti spojenej s udalosťou.
zone_dest	Do ktorej sieťovej zóny je komunikácia alebo udalosť smerovaná?	Cieľová sieťová zóna, kam bola aktivita smerovaná.
zone_src	Z ktorej sieťovej zóny pochádza komunikácia alebo udalosť?	Zdrojová zóna, odkiaľ udalosť vznikla.
logon_method	Aký spôsob prihlásenia používateľ použil?	Spôsob autentifikácie použitý na prístup do systému (napr. heslo, viacfaktorová autentifikácia, biometria).
traffic_direction	Akým smerom sa pohybovala sieťová komunikácia?	Určuje orientáciu sieťového prenosu, napríklad či išla komunikácia von zo systému (outbound) alebo dovnútra (inbound).

Tab. 3.2: Prehľad metadátových kľúčov typu pre textových poskytovateľov.

3.3 Návrh čiastkových funkcií

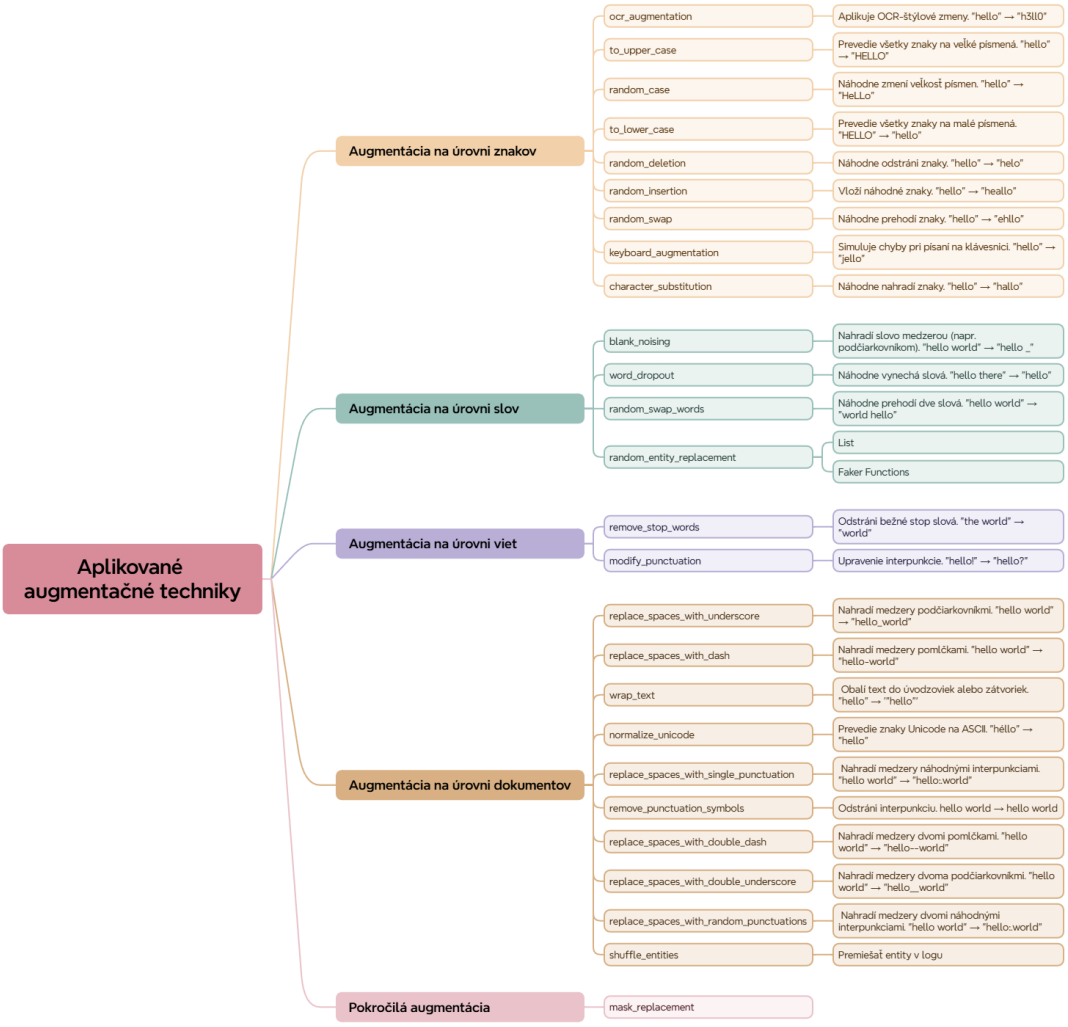
V rámci návrhu implementácie augmentačných metód sa budú využívať techniky identifikované v analytickej fáze projektu. Tieto techniky budú kombinované s cieľom maximalizovať efektivitu augmentácie datasetu logových záznamov pri zachovaní ich sémantickej aj syntaktickej štruktúry. Dôležitým krokom je identifikovať, ktoré z analyzovaných metód augmentácie sú vhodné pre jednotlivé entity logových záznamov. Vybrané metódy budú klasifikované na statické techniky a metódy využívajúce pokročilé technológie, ktoré umožňujú generovanie syntetických údajov, ktoré reflektujú reálne vzory logových dát. Dôraz bude kladený na systematickú kombináciu statických augmentačných techník s pokročilými prístupmi. Cieľom je dosiahnutie optimálnej variability výsledných logových záznamov, pričom sa zároveň minimalizuje riziko zavedenia neželaných nepresností alebo anomálií.

Metódy podľa úrovne aplikácie:

- **Globálne metódy** – aplikujú sa na celý log ako celok, čím ovplyvňujú jeho štruktúru, syntax alebo formátovanie.
- **Entitovo-špecifické metódy** – zameriavajú sa na konkrétne entity v rámci logov (napr. IP adresy, používateľské mená, názvy súborov).

Na obrázku 3.6 je znázornená myšlienková mapa prehľadne sumarizujúca všetky navrhnuté augmentačné techniky, ktoré sú rozdelené podľa úrovní, na ktorých sa vykonáva samotná augmentácia dát. Myšlienková mapa rozlišuje medzi nasledovnými kategóriami:

- **Augmentácia na úrovni znakov** – substitúcia znakov, mazanie znakov, vkladanie znakov, simulácia preklepov (napr. náhodné prehodenie znakov alebo použitie znakov z blízkych kláves na klávesnici), OCR simulácie, zmena veľkosti znakov.
- **Augmentácia na úrovni slov** – vymazanie slov, manipulácia poradia slov, vygenerovanie nových entít, zámena slov za podtržítka
- **Augmentácia na úrovni viet** – odstránenie nepotrebných slov, zmena interpunkcie
- **Augmentácia na na úrovni dokumentov** — reorganizácia entít logov, zámena medzier za iné znaky v celom logu, obalenie jednotlivých entít zvolenými znakmi.
- **Pokročilá augmentácia** — vygenerovanie entít `[mask]` pomocou AI modelu.



Obr. 3.6: Myšlienková mapa navrhnutých augmentačných metód.

4 Implementácia nástroja

Nástroj pre rozširovanie logových záznamov je napísaný v jazyku Python, ktorý bol zvolený ako programovací jazyk kvôli jeho jednoduchosti, čitateľnosti a širokej podpore predimplementovaných knižníc a funkcionalít, ktoré výrazne zjednodušili implementáciu projektu. Nástroj funguje na princípe spracovania vstupného logu vo formáte JSON. Tento log je štruktúrovaný na jednotlivé entity, pričom každá entita obsahuje názov poskytovateľa (angl. *entity group*), hodnotu entity (angl. *word*) a pozície (angl. *start a end position*). Na základe názvu entity sú vyvolaní príslušní poskytovatelia (angl. *providers*), ktorí generujú nové hodnoty buď z definovaných zoznamov (listov), alebo prostredníctvom knižnice Faker. Tieto novo vygenerované hodnoty sú následne rozšírené pomocou čiastkových funkcií, ktoré využívajú techniky textovej augmentácie. Výstupom nástroja je novo vytvorený dataset logov, kde sú jednotlivé hodnoty entit upravené a doplnené. Nástroj taktiež zachováva štruktúru a formát pôvodných hodnôt, čím je zabezpečená kompatibilita a správnosť generovaných údajov.

4.1 Implementácia poskytovateľov

Do projektu bola integrovaná knižnica `Faker`, ako bolo zmienené v predchádzajúcej kapitole. Ako prvé boli zmapovaní existujúci poskytovatelia, ktorých knižnica Faker ponúka. Tabuľka 4.1 sumarizuje konkrétnych poskytovateľov použitých v projekte spolu s príslušnými dátovými typmi, ktoré sú generované pre jednotlivé entity logových záznamov. Výber poskytovateľov bol vykonaný na základe detailnej analýzy požiadaviek na štruktúru logových dát, ako aj typických scenárov spracovania udalostí v bezpečnostných systémoch. Nie všetky, však vyhovovali požiadavkám projektu, a preto boli vytvorené vlastné varianty prispôsobené špecifickým potrebám jednotlivých entít. Tieto vlastné verzie si vyžadovali detailnú analýzu a návrh generátorov, ktoré zohľadňovali požadované parametre. Ako príklad je uvedená tabuľka 4.2 , kde je možné vidieť rôzne formáty časových pečiatok, ktoré boli zavedené pomocou vlastného generátora. Po zmapovaní všetkých možností v knižnici Faker boli implementované nové generátory vo formáte `.txt`, čím sa zabezpečila ich jednoduchá integrácia do ďalších častí projektu. Ako už bolo uvedené v predchádzajúcej kapitole, tieto dáta boli získané z viacerých zdrojov ako sú anonymizované datasety, databáza `MITRE ATT&CK` a využitá bola aj umelá inteligencia (AI), konkrétne jazykový model GPT-4 . V tabuľke 4.3 sú uvedení konkrétni poskytovatelia spolu s prehľadom typov dát, ktoré boli získané z týchto textových súborov.

Poskytovatelia	Popis	Príklady hodnôt
cvss	Skóre zraniteľnosti (0.0–10.0) podľa CVSS	9.6
cve	Identifikátor zraniteľnosti podľa CVE štandardu	CVE-2000-12005
domain_name	Názov domény	davis.com
host_id	Unikátne ID hostiteľa	52cd430afacd
email_src	E-mailová adresa odosielateľa	john39@example.org
email_dest	E-mailová adresa príjemcu	yvonnelittle@example.net
event_id	ID udalosti je jedinečný identifikátor priradený typu udalosti v systéme.	4624
file_name	Názov súboru s príponou	morning.webm
host_name	Názov hostiteľa v sieti	desktop-00.marshall-flores.biz
ip_src	Zdrojová IP adresa	213.67.125.57
ip_dest	Cieľová IP adresa	3b08:c6e3:3c72:9578:2d6c:7980:8f7d:9b78
device_ip	IP adresa zariadenia, ktoré je súčasťou udalosti (napr. cieľové alebo zdrojové)	3b08:c6e3:3c72:9578:2d6c:7980:8f7d:9b78
mac_src	Zdrojová MAC adresa	bc:45:b8:5d:e4:d4
mac_dest	Cieľová MAC adresa	f4:ba:b5:b9:e4:52
name	Meno používateľa alebo entity	Michelle Murray
os_build	Build číslo operačného systému	4192656
process_id	Identifikátor procesu (PID)	3814206
parent_process_id	Identifikátor nadradeného procesu	4110535
parent_process_name	Názov nadradeného procesu	wish.exe
port_dest	Cieľový port	65282
port_src	Zdrojový port	46389
process_name	Názov vykonaného procesu	have.js
service_id	ID služby (napr. UUID)	8ebdbfe3-eb9a-4688-b9d3-9cca91551e82
session_id	Identifikátor používateľskej relácie	ef7a58f99d96fb2a0631187348761d11bb57...
status_code	HTTP stavový kód alebo výsledok operácie	232
timestamp	Časová značka udalosti	14.4.2021 17:47
url	Webová adresa alebo odkaz (URL)	https://www.jackson.com/categories/main/appindex.html
user_id	Identifikátor používateľa	3bxD7
user_name	Používateľské meno	vangjesse
geoip_dest	Krajina cieľovej IP adresy	Kenya
geoip_src	Krajina zdrojovej IP adresy	Barbados
host_src	Názov zdrojového hostiteľa	desktop-19.weber.com
host_dest	Názov cieľového hostiteľa	desktop-64.christensen.com
mask_dest	Maska podsiete pre cieľovú IP	255.255.240.0
mask_src	Maska podsiete pre zdrojovú IP	255.255.255.0
domain_dest	Doména cieľa v sieti	hunt.com
domain_src	Doména zdroja v sieti	garcia.com
user_domain	Doména používateľa	robinson.info
user_src	Zdrojové používateľské meno	pcollins
user_dest	Cieľové používateľské meno	wlewis
file_hash	Kryptografický hash súboru	6ebdebf0940ca07664680144c6b6aaa241b41430
hash	Kryptografický hashový reťazec	b870f3e3011ccce369dc7b1c1cb6711d2aa1...
interface_dest	Cieľové sieťové rozhranie	usbz
interface_src	Zdrojové sieťové rozhranie	vmnetf
file_path	Úplná cesta k súboru v systéme	C:\Windows\System32\ja-jp\JpnComponentLayouts.dgml
result_code	Kód výsledku operácie	512

Tab. 4.1: Ukážka poskytovateľov a generovaných hodnôt `Faker` knižnice.

Formát časovej pečiatky	Príklad
%s	1733586330
%Y-%m-%dT%H:%M:%S%z	2024-12-07T15:45:30+0000
%Y%m%dT%H%M%SZ	20241207T154530Z
%Y-%m-%dT%H:%M:%S.%f%z	2024-12-07T15:45:30.123456+0000
%b %d %H:%M:%S	1.12.2007 15:45
%Y-%m-%d %H:%M:%S	7.12.2024 15:45
%d/%m/%Y %H:%M:%S	7.12.2024 15:45
%d|%m|%Y%H:%M	07|12|2024 15:45
%Y.%m.%d %H:%M:%S	7.12.2024 15:45
%a %d-%b-%Y %H:%M:%S %Z	Sat 07-Dec-2024 15:45:30 UTC
%Y-%m-%d %H:%M:%S.%f	2024-12-07 15:45:30.123456
%Y-%j	2024-342
%Y-W%V-%u	2024-W49-6
%Y-%m	2024-12
%Y-%U	2024-49
%Y%m%d%H%M%S	20241207154500
%Y%m%dT%H%M%S	20241207T154530
%b %d %Y %H:%M:%S.%f	Dec 07 2024 15:45:30.123456
%b %d %Y %H:%M:%S	Dec 07 2024 15:45:30
%b %d %Y %H:%M:%S %z	Dec 07 2024 15:45:30 +0000
%Y-%m-%dT%H:%M:%S	2024-12-07T15:45:30
%Y-%m-%dT%H:%M:%S.%f	2024-12-07T15:45:30.123456
%Y-%m-%d'T'%H:%M:%S*%f%z	2024-12-07'T'15:45:30*123456+0000
%Y %b %d %H:%M:%S.%f %Z	2024 Dec 07 15:45:30.123456 UTC
%b %d %H:%M:%S %z %Y	Dec 07 15:45:30 2024 +0000
%d/%b/%Y:%H:%M:%S %z	07/Dec/2024:15:45:30 +0000
%b %d %Y %I:%M:%S %p	Dec 07 2024 03:45:30 PM
%b %d %H:%M:%S %Y	1.12.2007 15:45
%b %d %H:%M:%S %z	Dec 07 15:45:30 +0000
%b %d %H:%M:%S	1.3.2016 8:12
%Y-%m-%d'T'%H:%M:%S%z	2023-10-14T22:11:20+0000
%Y-%m-%d'T'%H:%M:%S.%f'Z'	2023-07-01T14:59:55.711+0000
%Y-%m-%d %H:%M:%S %z	19.8.2023 12:17
%Y-%m-%d %H:%M:%S%z	19.8.2023 12:17
%Y-%m-%d %H:%M:%S%f	2024-12-07 15:45:30.123456
%Y/%m/%d*%H:%M:%S	2023/04/12*19:37:50
%Y %b %d %H:%M:%S.%f*%Z	2023 Apr 13 22:08:13.211*PDT
%Y %b %d %H:%M:%S.%f	2024 Mar 10 01:44:20.392
%Y-%m-%d %H:%M:%S%f%z	2024-12-07 15:45:30.123456+0000
%Y-%m-%d %H:%M:%S.%f	27.2.2024 15:35
%Y-%m-%d %H:%M:%S.%f%z	12.3.2024 13:11
%Y-%m-%d'T'%H:%M:%S.%f	2023-07-22'T'16:28:55.444
%Y-%m-%d'T'%H:%M:%S	2023-09-08'T'03:13:10
%Y-%m-%d'T'%H:%M:%S'Z'	2024-03-12'T'17:56:22'-0700'
%Y-%m-%d'T'%H:%M:%S.%f	2023-11-22'T'10:10:15.455
%Y-%m-%d*%H:%M:%S:%f	2023-10-30*02:47:33:899
%Y-%m-%d*%H:%M:%S	2023-07-04*13:23:55
%y-%m-%d %H:%M:%S%f %z	23-12-07 15:45:30.123456 +0000
%y-%m-%d %H:%M:%S%f	23-12-07 15:45:30.123456
%y-%m-%d %H:%M:%S	23.4.2019 12:00
%y/%m/%d %H:%M:%S	23.1.2006 4:11
%y%m%d %H:%M:%S	220423 11:42:35

Tab. 4.2: Ukážka implementovaných časových pečiatok.

Poskytovatelia	Popis	Príklady hodnôt
dlp_type	Typ klasifikácie DLP podľa citlivosti dát	performance metrics
event_level	Úroveň závažnosti udalosti	network timeout
event_type	Typ alebo kategória udalosti	Access
host_platform	Typ alebo OS hostiteľského systému	virtual machine
logon_method	Spôsob prihlásenia používateľa do systému	Active Directory Integrated Authentication
os_architecture	Architektúra operačného systému	ARM Cortex-R
os_type	Typ operačného systému	AROS (AROS Research Operating System)
os_version	Verzia operačného systému	v10.0.19044
os_name	Názov operačného systému nainštalovaného na hostiteľovi	Windows 10
result	Výsledok vykonanej akcie	IP blocked
severity	Závažnosť alebo dopad udalosti	CRITICAL
tactic	MITRE ATT&CK taktika	Reconnaissance
tags	Štítky priradené k udalosti	man-in-the-middle
technique	MITRE ATT&CK technika útoku	container administration command
threat_actor	Meno alebo názov útočníka/skupiny	Mofang
threat_type	Typ kybernetickej hrozby	APT
url_category	Kategória URL adresy	supply chain attacks
user_group	Skupina používateľov v systéme	DNS Admins
user_type	Typ používateľa podľa oprávnení	Network Administrator
vuln_name	Názov zraniteľnosti	Petya
zone_dest	Cieľová sieťová zóna	external
zone_src	Zdrojová sieťová zóna	external
description	Popis poskytuje podrobné vysvetlenie alebo zhrnutie udalosti a dopĺňa kontext.	Admin login failed.
action	Akcia označuje, čo sa vykonalo alebo spustilo v systéme počas udalosti.	update
command_line	Príkaz vykonaný v systéme	echo %USERNAME%
crypto	Kryptografický algoritmus použitý v udalosti	Blum Blum Shub
network_protocol	Použitý sieťový protokol	RTSP
traffic_direction	Smer sieťovej komunikácie	outbound
rule	Pravidlo alebo detekčný mechanizmus, ktorý spustil udalosť	credential-dumping-detected
event_source	Zdroj systému, ktorý vygeneroval udalosť	Intrusion Detection System
host_group	Skupina alebo kategória hostiteľov	Public Cloud Resources
service_name	Názov zapojenej služby	Bitdefender GravityZone

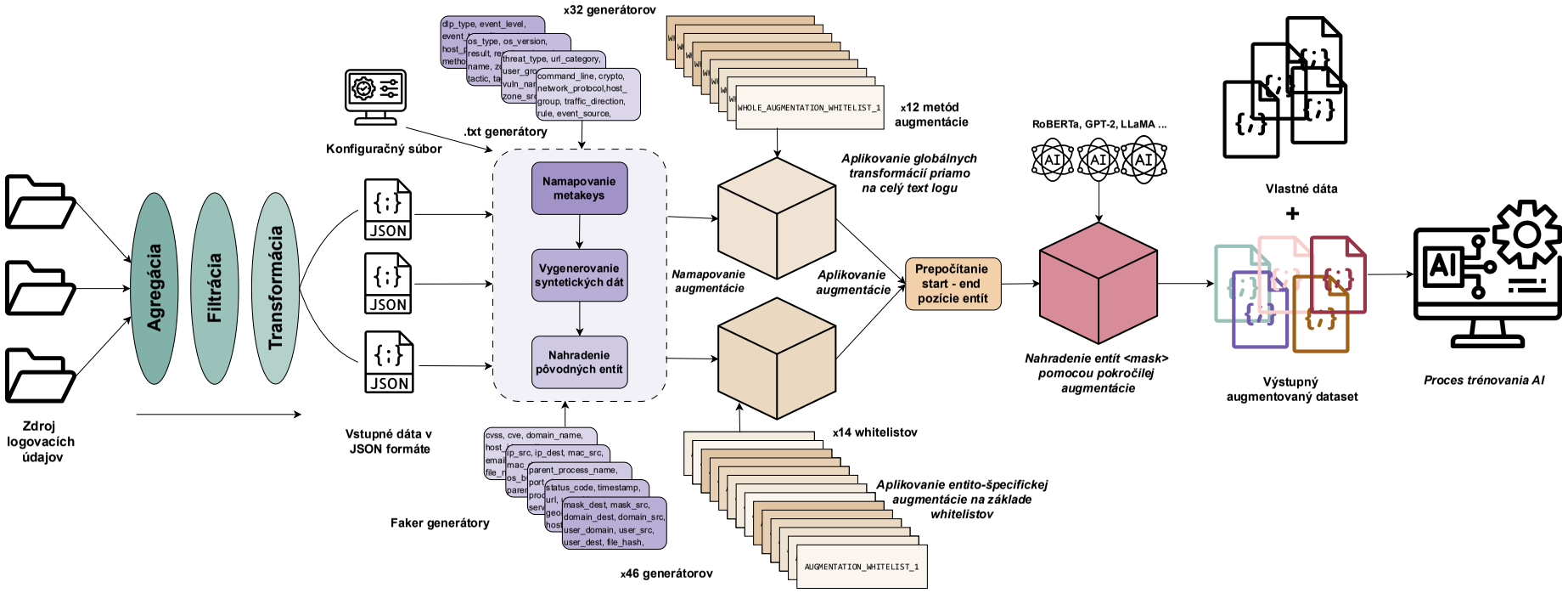
Tab. 4.3: Ukážka poskytovateľov a generovaných hodnôt z txt súborov.

4.2 Rozšírenie pomocou pokročilej augmentácie

Do projektu je integrované AI riešenie v rámci spolupráce s členom týmu [125] na generovanie maskovaných entít `<mask>` v bezpečnostných logoch, ktoré predstavuje pokročilú formu augmentácie textových dát. Tento proces využíva veľké jazykové modely (napr. RoBERTa, GPT-2, LLaMA), ktoré boli natrénované na rozsiahlej množine anonymizovaných logových záznamov. Cieľom je nahradiť chýbajúce alebo maskované hodnoty (ako IP adresy, používateľské mená, identifikátory zariadení a pod.) realistickými údajmi v kontexte ostatných častí logu. Na rozdiel od jednoduchých augmentačných techník, ktoré operujú na úrovni znakov alebo slov bez hlbšieho porozumenia textu, táto metóda využíva schopnosť modelov rozpoznať významové a štruktúrne súvislosti medzi jednotlivými časťami logu. Konkrétne sa aplikuje technika (angl. *Masked Language Modeling*) (MLM), kde model predpovedá chýbajúce slová (tokeny) v logu na základe obojstranného kontextu. V trénovacej fáze boli tokenizované logové záznamy s náhodne zamaskovanými entitami spracované modelom RoBERTa, ktorý dynamicky generoval predikcie pre jednotlivé masky. Okrem toho boli experimentálne testované aj modely ako ALBERT, MobileBERT a ELECTRA. Výsledkom tohto procesu je automatizované generovanie veľkého množstva syntetických, ale realistických logových záznamov, ktoré vykazujú vysokú variabilitu. Tieto rozšírené dáta sa následne využívajú na doladenie modelu T5, ktorý je určený na rozpoznávanie pomenovaných entít (NER) v bezpečnostných logoch. Tento prístup umožňuje efektívne zvýšiť objem a rozmanitosť trénovacích dát bez potreby manuálneho zásahu, čím sa zvyšuje presnosť a robustnosť výsledného modelu.

4.3 Štruktúra kódu nástroja pre rozšírenie záznamov

Nástroj je vyvíjaný vo vývojovom prostredí *Visual Studio Code* v programovacom jazyku Python a implementovaný ako balíček (angl. *package*), čo umožňuje jednoduchšiu integráciu do existujúcich projektov a efektívnejšiu prácu pri trénovaní neurónovej siete. Táto štruktúra uľahčuje správu kódu, jeho údržbu a konfiguráciu podľa potrieb používateľa. Pre zjednodušenie konfigurácie bola pridaná možnosť nastavenia augmentačných metód cez súbor `.env`, kde si používateľ môže zvoliť z viacerých možností konfigurácie nástroja. Celková implementácia nástroja je znázornená na obrázku 4.1, kde je zobrazený priebeh spracovania dát od počiatočného generovania realistických logových záznamov, cez použitie nástroja na augmentáciu až po ich využitie na trénovanie modelu. Kód obsahuje 5 modulov a je rozdelený do viacerých súborov pre lepšiu údržbu, prehľadnosť a prispôsobenie sa potrebám používateľa:



Obr. 4.1: Vizualizácia implementácie nástroja

- `main.py` – je hlavný spustiteľný súbor. Obsahuje inicializáciu generátorov, augmentátorov, konfigurácií a slúži na spracovanie a augmentáciu logov buď v testovacom režime, alebo na základe vstupného súboru s logmi.
- `log_config.py` – tento súbor načítava konfiguračné premenné zo súboru `.env`, ako napríklad cestu k logovacím súborom, úroveň logovania a rôzne nastavenia týkajúce sa maskovania a generovania syntetických dát.
- `.env` – konfiguračný súbor nastavuje logovanie a augmentáciu textov. Definuje cestu a veľkosť log súboru, úroveň logovania a logovanie na konzolu. Zároveň umožňuje aktivovať rôzne metódy augmentácie, ako nahrádzanie maskovaných slov, generovanie syntetických dát a úpravy textu pomocou *whitelistu*, vzápätí aj rozsah aplikácie augmentácie a ďalšie parametre.

Modul `file_manipulator`

- `file_creator.py` – vytvára súbory a adresáre na ukladanie dát, pričom uloží zadané JSON dáta do súboru `augmented_data.json` v priečinku output. Zabezpečuje, že adresár output existuje, a určuje jeho umiestnenie na základe koreňového adresára projektu.
- `file_reader.py` – číta obsah JSON súborov a vracia ich dáta ako zoznam.

Modul `log_augmentator`

- `code_functions` – tento priečinok obsahuje všetkých 44 generátorov využívajúcich Faker knihovňu, ktoré generujú nové syntetické dáta.
- `list_functions` – obsahuje všetky `.txt` súbory so syntetickými dátami, ktoré sú využívane na generovanie nových entít logov.
- `config.py` – obsahuje zoznam namapovaných poskytovateľov a *whitelisty* povolených augmentačných metód pre každého poskytovateľa.
- `custom_augmentator.py` – zabezpečuje konzistenciu medzi pôvodným textom a jeho modifikovanými verziami, pričom zachováva správne výpočty dĺžok a offsetov pre každú zmenenú entitu. Umožňuje uloženie zoznamu entít do JSON súboru, čo je užitočné pre neskoršiu analýzu alebo ďalšie spracovanie.
- `custom_generator.py` – slúži na generovanie a augmentáciu dát pomocou poskytovateľov v rámci knižnice Faker. Inicializuje poskytovateľov na základe externých zoznamov (listov), ktoré sú načítané a pridané do Faker, a následne generuje hodnoty, ktoré môžu byť augmentované rôznymi technikami textovej augmentácie.

Modul `pm_log_utils`

Umožňuje nastavovať logovanie aplikácie, zapisovať logy do súborov, logovať na konzolu a konfigurovať úroveň logovania a formát správ.

Modul `text_manipulation`

- `text_augmentation_selector.py` – spravuje zoznam dostupných a aktívnych

metód na augmentáciu textu, umožňuje ich povolenie a následné aplikovanie na text.

- `text_augmentator.py` – obsahuje rôzne metódy textovej augmentácie na rozšírenie logových záznamov.

Modul `pm_deep_augmentation`
Zabezpečuje pokročilú augmentáciu textu pomocou modelov hlbokého učenia. Umožňuje automatické dopĺňanie maskovaných slov <mask> v logových správach na základe predikcií jazykového modelu.

4.4 Implementácia čiastkových funkcií

Pre implementáciu čiastkových funkcií bolo zvolené riešenie založené na *whitelistoch*, ktoré sú definované v súbore `config.py`. V rámci týchto *whitelistov* môže používateľ explicitne špecifikovať vhodné augmentačné metódy prislúchajúce jednotlivým poskytovateľom entít. Augmentačné metódy je možné kombinovať alebo ponechať prázdne pole. Výber vhodnej augmentačnej metódy a jej následná aplikácia je komplexný proces, ktorý si vyžaduje detailné porozumenie štruktúre vstupných dát a sémantickému významu jednotlivých entít. V prípade nevhodne zvolených metód môže dôjsť k viacerým negatívnym dôsledkom, od narušenia štruktúr integrity logových záznamov, cez stratu významovej presnosti údajov, až po porušenie vzájomnej konzistencie medzi entitami. Okrem entity-špecifickej augmentácie program podporuje aj globálne transformácie aplikované priamo na celý text logu (*payload*). V tomto prípade nie sú transformácie viazané na konkrétne entity, ale sú aplikované na celý reťazec ako celok. Aj pri tomto type však hrozia riziká, ako napríklad generovanie nesprávnych údajov alebo narušenie logického kontextu logu. Samotné augmentačné metódy sú definované v triede `TextAugmentator`, ktorá bola vytvorená na základe analýzy uvedenej v kapitole 2.2. Táto trieda poskytuje základ pre rozširovanie logových záznamov, pričom vývoj bol orientovaný na minimalizáciu vyššie uvedených problémov. Tabuľka 4.4 poskytuje prehľad aplikovaných metód augmentácie spolu s ukážkami výstupu.

Globálne metódy

Tieto metódy sa aplikujú na celý log ako celok, čím ovplyvňujú štruktúru, syntax alebo formátovanie záznamov:

- `replace_spaces_with_underscore` – nahrádza medzery znakom `_`, čím simuluje alternatívne formátovanie textu.
- `replace_spaces_with_dash` – nahrádza medzery znakom `-`, často používaným v logoch na oddelenie reťazcov.

- `replace_spaces_with_double_underscore` – zvyšuje zložitosť výrazu použitím dvoch znakov `__`.
- `replace_spaces_with_double_dash` – obdobné ako vyššie, ale používa `-`, čo simuluje rozdiely v zápise medzi systémami.
- `replace_spaces_with_double_punctuations` – medzery sa nahradia dvoma náhodnými interpunkčnými znakmi (napr. `??`, `::`).
- `replace_spaces_with_single_punctuation` – používa jeden náhodný interpunkčný znak namiesto medzery.
- `wrap_text` – vloží logové reťazce do špecifických zátvoriek alebo značiek (napr. `<log>...<\log>`).
- `normalize_unicode` – konvertuje znaky na štandardný Unicode formát, čo eliminuje diakritiku a odchýlky.
- `remove_punctuation_symbols` – odstráni všetky interpunkčné znaky, čím sa testuje odolnosť tokenizácie.
- `SHUFFLE_ENTITIES` – zmení poradie entít v rámci logu, čím testuje závislosť na poradí.
- `entity_replacement` – nahrádza celé entity inými validnými príkladmi z rovnakého typu (napr. IP adresu inou IP adresou).

Entitovo-špecifické metódy

Tieto metódy sa aplikujú len na konkrétne entity v logu, ako sú IP adresy, používateľské mená, alebo iné tokeny:

- `to_upper_case` – prevedie entitu na veľké písmená.
- `to_lower_case` – prevedie entitu na malé písmená.
- `random_case` – náhodne kombinuje malé a veľké písmená v rámci jednej entity.
- `random_swap_words` – prehodí poradie slov v entite.
- `random_swap` – prehodí náhodné znaky v rámci slova.
- `random_insertion` – vloží náhodný znak do entity.
- `random_deletion` – odstráni náhodný znak z entity.
- `remove_stop_words` – odstráni bežné „zbytočné“ slová (napr. „the“, „is“), ak sa v entite nachádzajú.
- `word_dropout` – vynechá celé slovo.
- `blank_noising` – nahradí časť entity prázdnym znakom.
- `ocr_augmentation` – simuluje chyby spôsobené OCR.
- `keyboard_augmentation` – nahrádza znaky podľa fyzickej blízkosti na klávesnici.
- `character_substitution` – nahrádza znaky podľa definovaného zoznamu substitúcií.

- `modify_punctuation` – nahrádza, odstraňuje alebo vkladá interpunkčné znaky.

Metóda augmentácie	**Whitelist**	**Príklad výstupu**
Bez augmentácie		INFO User 'Šimčáková' logged in from IP 192.168.1.45 via web interface.
to_upper_case	AUGMENTATION_WHITELIST_1	INFO USER 'ŠIMČÁKOVÁ' LOGGED IN FROM IP 192.168.1.45 VIA WEB INTERFACE.
random_case	AUGMENTATION_WHITELIST_3	inFo uSeR 'ŠimčáKová' lOGGed in FroM iP 192.168.1.45 Via wEb InterfaCE.
to_lower_case	AUGMENTATION_WHITELIST_5	info user 'šimčáková' logged in from ip 192.168.1.45 via web interface.
ocr_augmentation	AUGMENTATION_WHITELIST_7	INFO User 'Šimčáková' lOgged in fr0rn IP 192.168.1.45 vla we6 interface.
random_deletion	AUGMENTATION_WHITELIST_8	INFO User 'Šimčáová' logged in fom IP 192.168.1.45 via web interface.
random_insertion	AUGMENTATION_WHITELIST_9	INFO User 'Šimčákhová' logged in fr*o*m IP 192.168.1.45 via web interface.
random_swap	AUGMENTATION_WHITELIST_10	INFO Uesr 'Šmičáková' logegd in from IP 192.168.1.45 via web interface.
keyboard_augmentation	AUGMENTATION_WHITELIST_11	INFO Uder 'Šimčálová' loggrd on frim IP 192.168.1.45 via web interfave.
character_substitution	AUGMENTATION_WHITELIST_12	INFO Us3r 'Šimčáková' lOgg3d in fr0m IP 192.168.1.45 v1a w3b 1nt3rfac3.
random_swap_words	AUGMENTATION_WHITELIST_2	INFO IP 'Šimčáková' logged in from User 192.168.1.45 via web interface.
word_dropout	AUGMENTATION_WHITELIST_6	INFO User logged from IP via interface.
blank_noising	AUGMENTATION_WHITELIST_14	INFO User 'Šimčáková' _ in from IP 192.168.1.45 via web interface.
remove_stop_words	AUGMENTATION_WHITELIST_4	INFO User 'Šimčáková' logged IP 192.168.1.45 via web interface.
modify_punctuation	AUGMENTATION_WHITELIST_13	INFO User „Šimčáková" logged in from IP 192.168.1.45 via web interface!
replace_spaces_with_ underscore	WHOLE_LOG_AUGMENTATION_1	INFO_User_'Šimčáková'_logged_in_from_IP_192.168.1.45 _via_web_interface.
replace_spaces_with_dash	WHOLE_LOG_AUGMENTATION_2	INFO-User-'Šimčáková'-logged-in-from-IP-192.168.1.45 -via-web-interface.
wrap_text	WHOLE_LOG_AUGMENTATION_3	[ĬNFO"] ["Šimčáková'"] "logged in"from IP (192.168.1.45) {via web interface}
normalize_unicode	WHOLE_LOG_AUGMENTATION_4	INFO User 'Simcakova' logged in from IP 192.168.1.45 via web interface.
replace_spaces_with _double_underscore	WHOLE_LOG_AUGMENTATION_5	INFO__User__'Šimčáková'__logged__in__from__IP__ 192.168.1.45__via__web__interface.
replace_spaces_with _double_dash	WHOLE_LOG_AUGMENTATION_6	INFO-User-'Šimčáková'-logged-in-from-IP-192.168 .1.45-via-web-interface.
replace_spaces_with _double_punctuations	WHOLE_LOG_AUGMENTATION_7	INFO..User"Šimčáková";;logged,:in-from„IP;;192.168 .1.45:;via/,web..interface.
remove_punctuation_ symbols	WHOLE_LOG_AUGMENTATION_8	INFO User Šimčáková logged in from IP 192168145 via web interface
replace_spaces_with _single_punctuation	WHOLE_LOG_AUGMENTATION_9	INFO.User.'Šimčáková'.logged.in.from.IP.192.168.1.45 .via.web.interface.
SHUFFLE_ENTITIES	SHUFFLE_ENTITIES	Šimčáková' 192.168.1.45 logged interface [2025-04-11 14:32:08] INFO via web in User from IP

Tab. 4.4: Ukážka poskytovateľov a generovaných hodnôt z txt súborov.

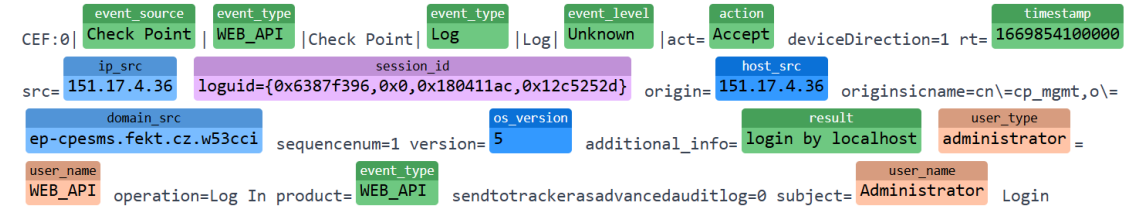
4.5 Priebeh spracovania logov v nástroji

Nástroj je navrhnutý na spracovanie logových záznamov vo formáte JSON, pričom každý záznam obsahuje samotný payload a zoznam entít, ktoré je potrebné identifikovať a neskôr modifikovať. Ukážka vstupných dát je zobrazená na obrázku 4.2. Spracovanie prebieha v niekoľkých po sebe idúcich fázach, ktoré spolu zabezpečujú rozšírenie dát a zvýšenie ich variability pre potreby ďalšieho využitia, napríklad pri trénovaní modelov strojového učenia.

Obr. 4.2: Ukážka vstupných dát.

Pred samotným spustením programu má používateľ možnosť prostredníctvom konfiguračného súboru detailne definovať správanie systému vrátane aplikácie augmentačných techník na vybrané segmenty logu, aktivácie alebo deaktivácie generovania syntetických dát, výberu špecifickej augmentačnej metódy, ako aj zapnutia mechanizmu generovania masiek pre identifikované entity.

Celý proces funguje na princípe detailného rozkladu vstupného logu, ktorý je štruktúrovaný ako JSON objekt. Na obrázku 4.3 je znázornený log pred aplikáciou augmentácií a generovania syntetických dát, na ktorý neskôr budú aplikované zmeny po augmentácií.[1]

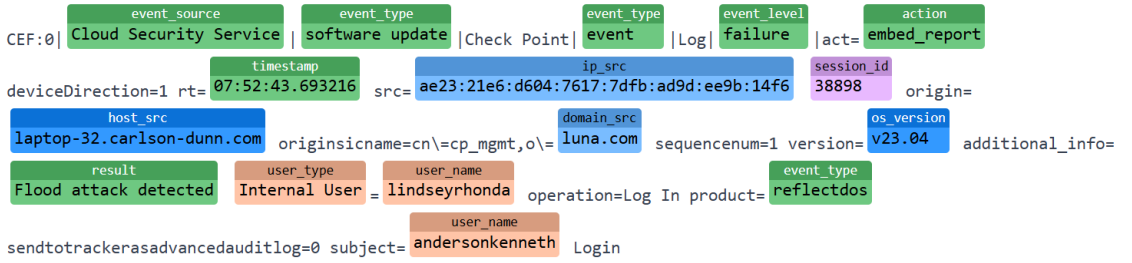


Obr. 4.3: Ukážka logu bez použitia nástroja.

Tento objekt obsahuje jednak reťazec payload, ktorý predstavuje pôvodný reťazec logu, ako aj zoznam entít pod kľúčom entities, ktoré reprezentujú významové jednotky identifikované v texte logu. Tento objekt obsahuje informácie o jednotlivých entitách, kde každá entita má definovaný svoj názov (`event_source`), hodnotu (`word`) a pozíciu výskytu v rámci pôvodného reťazca pomocou dvojice hodnôt `start` a `end`. Na základe názvu entity je následne vyvolaný zodpovedajúci poskytovateľ dát, ktorý má na starosti generovanie novej hodnoty. Títo poskytovatelia sú implementovaní buď ako zoznamy statických hodnôt (listy), alebo ako volania funkcií

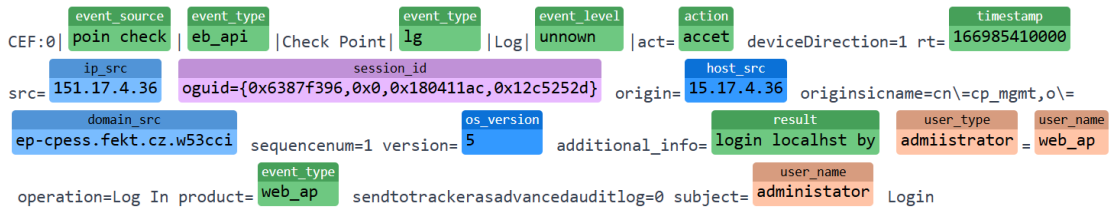
[1]Vizualizácia logu bola umožnená prostredníctvom aplikácie: Platforma pro adaptivní dolování znalostí z logových záznamů pomocí technik umělé inteligence (VB02000059). Aplikácia bola vyvinutá v rámci výzkumného projektu na VUT.

knižnice `Faker`, ktorá generuje realisticky vyzerajúce údaje ako sú emaily, IP adresy, mená, názvy domén a podobne. Ukážku logu je možné vidieť na obrázku 4.4, kde sú pôvodné entity označené a nahradené synteticky vygenerovanými hodnotami.



Obr. 4.4: Ukážka logu po vygenerovaní nových syntetických dát.

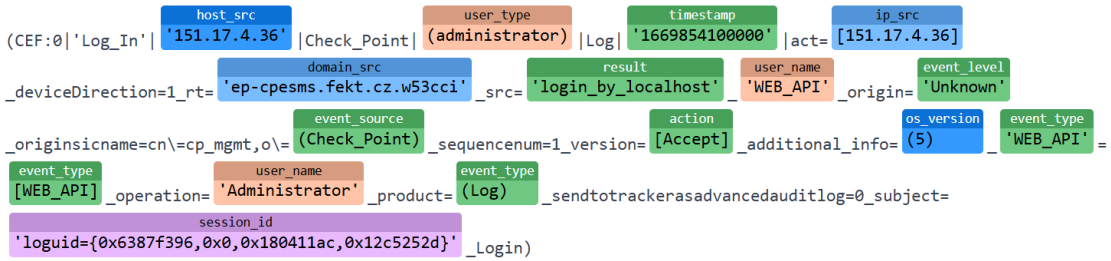
Na vygenerované hodnoty sú následne aplikované rôznorodé augmentačné techniky, ktoré boli navrhnuté na základe dôkladnej analýzy a slúžia na maximalizáciu variability vstupných údajov. Cieľom týchto techník je simulovať rozličné podoby zápisu logu, čím sa zabezpečí robustnosť a generalizovateľnosť modelov pri následnom spracovaní dát. Jednotlivé augmentačné metódy sú cielene aplikované na vybrané entity pomocou tzv. *whitelistov*, ktoré definujú konkrétne entity vhodné na augmentáciu. Tento prístup minimalizuje riziko zavádzania šumu do dát prostredníctvom modifikácie nevhodných alebo citlivých entít, ktoré by mohli negatívne ovplyvniť kvalitu trénovania modelov umelej inteligencie. Augmentácia pomocou enitovo-špecifických metód je znázornená na obrázku 4.5, kde na pôvodný log boli aplikované metódy `random_deletion`, `to_lower_case` a `random_swap_words`.



Obr. 4.5: Ukážka logu po aplikácií entitovo-špecifických augmentácií.

Augmentácia nie je obmedzená iba na úroveň entít, súčasťou systému sú aj techniky, ktoré modifikujú celý log ako celok, nezávisle od konkrétnych entít, čím sa dosahuje komplexnejšia a rozmanitejšia transformácia vstupného datasetu. Navyše, systém podporuje kombináciu viacerých *whitelistov* naraz, čo umožňuje vyššiu flexibilitu pri konfigurácii augmentačných scenárov a ich väčšiu variabilitu. Medzi implementované metódy patrí napríklad zmena veľkosti písmen (transformácia na malé, veľké alebo náhodné písmená), zámena medzier za znaky ako podtržítko, pomlčka

alebo ich duálne varianty, či náhodná úprava znakov, napríklad ich zámenná náhrada inými znakmi, vynechanie alebo vloženie znakov. Tieto techniky umožňujú generovať rozmanité varianty logov, čím sa zvyšuje variabilita a robustnosť modelov pri analýze logových dát. Ďalšou implementovanou metódou je technika „shuffle entity“, ktorá náhodne premiešava pozície entít v logoch. Tento proces zachováva význam entít, pričom mení ich pozíciu v payloade, čím sa dosahuje vyššia variabilita v spracovaní logu. Na obrázku 4.6 je znázornená ukážka globálnych augmentácií, kde boli na pôvodný log aplikované techniky ako `wrap_text`, `replace_spaces_with_underscore` a `shuffle_entities`.



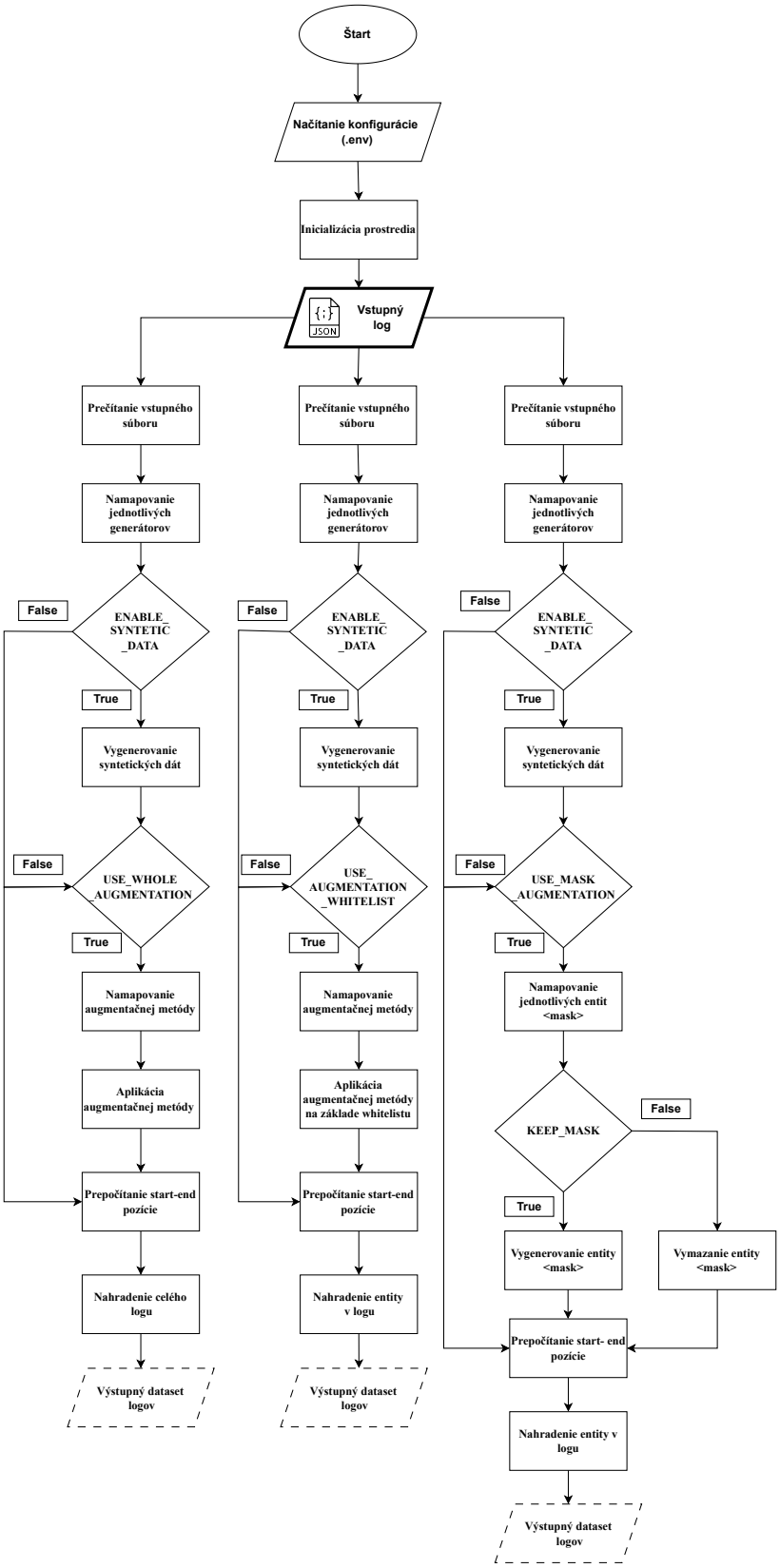
Obr. 4.6: Ukážka logu po aplikácii globálnych augmentácií.

Niektoré techniky navyše simulujú bežné chyby pri OCR spracovaní alebo zámeny spôsobené susednými znakmi na QWERTY klávesnici, čo zlepšuje schopnosť modelov prispôsobiť sa reálnym chybám v texte. Tieto metódy sa aplikujú na vybraný percentuálny podiel entít, čo umožňuje flexibilné nastavenie úrovne variácie v závislosti od požiadaviek.

Po aplikácii augmentačných metód na text dochádza k zmene jeho obsahu, čo následne ovplyvňuje pozície entít v pôvodnom texte. Každá entita v texte je definovaná svojimi počiatočnými a konečnými pozíciami, ktoré označujú jej začiatok a koniec v rámci textového payloadu. Ak je text entít nahradený alebo zmenený, tieto pozície sa môžu stať nepresnými, preto je potrebné ich správne upravit, aby stále zodpovedali skutočnému umiestneniu entít v novo upravenom texte. Pri zmene textu entity, ako napríklad pri jej nahradení za novú hodnotu, je potrebné vypočítať rozdiel v dĺžke medzi pôvodným a novým textom. Tento rozdiel, sa následne aplikuje na pozície všetkých nasledujúcich entít, aby sa ich pozície správne posunuli podľa zmeny dĺžky nahradeného textu. Týmto spôsobom sa zabezpečí, že všetky entity, ktoré sa nachádzajú po upravenej entite, budú mať aktualizované svoje pozície v texte. V prípade globálnych zmien v texte, ako je zámena poradia entít alebo komplexné transformácie celého textu, nie je možné jednoducho vychádzať zo zmeny pozícií na základe dĺžky. V takýchto prípadoch sa využíva metóda vyhľadávania entít v texte pomocou regulárnych výrazov. Každá entita sa vyhľadáva podľa svojho obsahu, ktorý je definovaný v poli `word`, a následne sa určuje jej nová pozícia v

upravenom texte. Tento postup zaručuje, že entity budú vždy presne lokalizované v textovom *payloade*, aj keď došlo k významným zmenám v jeho štruktúre.

Na záver je zavolaný modul pokročilej augmentácie, doplňania maskovaných entít pomocou jazykových modelov, ako sú RoBERTa, GPT-2 a LLaMA, ktoré na základe kontextu predikujú realistické hodnoty nahrádzajúce tokeny `<mask>`. Po spracovaní logov všetkými vyššie spomenutými metódami nasleduje výstupná fáza, v ktorej nástroj vytvorí nový súbor obsahujúci augmentované logové záznamy. Výstupné dáta si zachovávajú pôvodnú štruktúru, ale obsahujú nové, rozšírené hodnoty entít, pripravené na ďalšie spracovanie alebo použitie v experimentoch so strojovým učením.



Obr. 4.7: Vývojový diagram nástroja.

5 Testovanie nástroja

Testovanie nástroja prebiehalo na dvoch rôznych datasetoch. Prvý dataset *D1* obsahuje logy z rôznych zariadení a systémov (napr. Linux, Windows, Fortinet, Synology, Cisco ASA, Snort, Apache), v rôznych formátoch (Syslog, CEF, CLF, atď.), čo zaručuje vysokú variabilitu a rôznorodosť dát. Naproti tomu druhý dataset *D2* je úzko zameraný na bezpečnostné logy zo zariadení Cisco ASA a poskytuje konzistentné a špecifické dáta. Ako model bol zvolený obojsmerný LSTM (biLSTM), ktorý je osvedčenou architektúrou pre spracovanie sekvenčných údajov typických pre logové dáta. Výhodou tejto architektúry v porovnaní s komplexnejšími modelmi, ako napríklad T5, je dosiahnutie porovnateľnej presnosti pri podstatne nižších výpočtových nárokoch. Tým sa biLSTM stáva efektívnou a praktickou voľbou pre aplikáciu v reálnych prostrediach, kde sú obmedzené výpočtové zdroje. V rámci analýzy výsledkov sa prihliadalo aj na jednotlivé entity, keďže ich zastúpenie v datasetoch nebolo rovnomerné, niektoré entity výrazne prevyšovali ostatné. Z dôvodu tejto nevyváženosti a prítomnosti entít, ktoré sa nepodarilo korektne namapovať, nebolo možné plnohodnotne overiť účinnosť nástroja na rozširovanie logov.

Tréningová sada bola rozdelená v pomere 80 % na trénovanie a 20 % na validáciu. Počas trénovania bolo 30 % dát augmentovaných, aby sa model vystavil väčšej variabilite a potenciálne lepšie generalizoval na menej zastúpené entity. Výsledky boli sledované pomocou nástroja MLflow, ktorý umožnil prehľadnú analýzu a porovnanie experimentov. Počas testovania sa využívali rôzne augmentačné metódy, samostatne aj v kombinácii s generátorom syntetických dát, s cieľom posúdiť ich prínos. Výsledky sa následne porovnávali s *baseline* modelom, trénovaným na neaugmentovanom datasete.

5.1 Analýza výsledkov

Počas analýzy výsledkov bolo potrebné sa zamerať na rôzne aspekty testovania a identifikovať hlavné faktory ovplyvňujúce výkon modelov. Výsledky boli porovnávané na základe F1 skóre, čo je metrika používaná na hodnotenie výkonu klasifikačných modelov, najmä v prípadoch, keď je rozdelenie tried nerovnomerné. Zachytáva kompromis medzi presnosťou (angl. *precision*) a úplnosťou (angl. *recall*). F1 skóre bolo porovnávané ako pre jednotlivé entity, tak aj celkovo, aby sa vyhodnotila účinnosť modelu aj na úrovni jednotlivých entít. V rámci testovania boli metódy aplikované na prvom datasete *D1* bez generovania syntetických dát, ako je znázornené v tabuľke 5.1, kde *baseline* dosahuje celkové skóre 0.7985. Najlepšie výsledky v tomto prípade dosahovali metódy `remove_stop_words` (+0,0108),

`random_swap` (+0,0060) a `to_lower_case` (+0,0048), no aj ostatné metódy prispeli ku zlepšeniu presnosti modelu. Ďalšia tabuľka 5.2 uvádza výsledky pre prvý dataset *D1* s generovaním syntetických dát, kde skóre bolo o nižšie ako v prípade bez generovaných dát. Najvyššie skóre v tomto prípade dosahovala metóda `word_dropout` (+0,0013). Testovanie na druhom datasete *D2* je znázornené v tabuľke 5.3, kde *baseline* dosahuje veľmi vysoké overall F1 skóre 0.9989. Tento výsledok prekonala iba metóda `character_substitution` (+0,0004), pričom porovnateľný výkon s *baseline* dosiahla metóda `blank_noising` (+0,0000). V tabuľke 5.4 sú zaznamenané výsledky pre druhý dataset *D2* s generovaním syntetických dát, kde sa skóre pri niektorých metódach zlepšilo a prekonalo *baseline*, konkrétne pri metódach `random_swap_words` (+0,0004), `to_lower_case` (+0,0004), `random_case` (+0,0002) a `random_insertion` (+0,0000). Neskôr boli testované aj kombinácia najlepších metód, ktorá však nedosahovala lepšie výsledky oproti aplikácií týchto metód samostatne. Táto skutočnosť naznačuje, že ani kombinovanie viacerých úspešných augmentácií nemusí automaticky viesť k lepšiemu výkonu modelu, a preto je potrebné venovať pozornosť vhodnej konfigurácii a optimalizácii týchto metód. V rámci testovania boli hodnotené aj jednotlivé globálne metódy, ktoré sa však nepreukázali ako vhodná forma augmentácie. Zmeny spôsobené týmito metódami viedli k tomu, že model stratil schopnosť správne identifikovať jednotlivé entity. Napriek tomu však tieto metódy a aj ostatné testované augmentačné prístupy, ktoré viedli len k miernemu zlepšeniu alebo udržaniu presnosti, predstavujú dôležitý prínos z viacerých dôvodov. Predovšetkým, tieto metódy prispievajú k rozšíreniu variability tréningových dát a vystavujú model novým vzorom, ktoré mu umožňujú lepšie generalizovať na dáta z reálneho sveta, kde sa logy môžu vyskytovať v rôznych formátoch a obsahovať neočakávané šumové alebo chybové prvky. Aj malé prírastky v skóre môžu byť indikátorom toho, že model je odolnejší voči variabilite a lepšie pripravený na nasadenie v produkčnom prostredí. Okrem toho, aj metódy, ktoré nedosiahli výrazné zvýšenie F1 skóre, pomáhajú testovať robustnosť modelu a jeho schopnosť vysporiadať sa s rôznymi druhmi vstupného šumu. V praxi to znamená, že model nemusí byť závislý len na presne štruktúrovaných logoch, ale dokáže pracovať aj s dátami, ktoré sú neúplné, šumové alebo pochádzajú z iných zdrojov. Tieto experimenty sú dôležité aj pre budúci vývoj, pretože ukazujú, ktoré metódy majú potenciál a ktoré je vhodné ďalej optimalizovať alebo kombinovať s inými prístupmi.

Záverom možno konštatovať, že napriek tomu, že niektoré augmentačné metódy nepriniesli významné zlepšenie presnosti alebo F1 skóre, tieto metódy sú stále cenným nástrojom na zvyšovanie robustnosti a generalizácie modelu a mali by byť súčasťou procesu testovania a ladenia modelov pre spracovanie logových dát. V budúcnosti bude potrebné zamerať sa na viaceré vylepšenia datasetov, ktoré môžu výrazne prispieť k zlepšeniu účinnosti augmentačných metód a celkovej výkonnosti

modelu. V prvom rade bude vhodné znížiť množstvo metakľúčov, ktoré sú aktuálne namapované, pretože niektoré z nich sa často prekrývajú svojimi hodnotami a môžu tým spôsobovať nejednoznačnosť v interpretácii logových záznamov. Taktiež bude potrebné zjednodušiť a štandardizovať názvy jednotlivých metakľúčov, aby sa predišlo nejasnostiam pri anotácii a vyhodnocovaní výsledkov. Okrem toho bude dôležité zabezpečiť, aby datasety obsahovali dostatočné množstvo príkladov zo všetkých relevantných kategórií entít. Nedostatočné zastúpenie niektorých kategórií môže totiž viesť k skresleným výsledkom, preto by bolo vhodné pri budovaní datasetov dbať na vyváženosť a reprezentatívnosť všetkých dôležitých entít.

Metóda augmentácie	**user_name**	**device_ip**	**crypto**	**description**	**Overall F1**	ΔF_1
Bez augmentácie	0,6667	0,7222	0,7673	0,9364	0,7985	0,0000
remove_stop_words	0,6571	0,8581	0,7826	0,9337	0,8093	+0,0108
random_swap	0,6933	0,7733	0,7308	0,9392	0,8045	+0,0060
modify_punctuation	0,6296	0,7785	0,8444	0,9326	0,8031	+0,0046
to_upper_case	0,6933	0,8182	0,6897	0,9358	0,7992	+0,0007
keyboard_augmentation	0,6038	0,7733	0,7075	0,9341	0,7990	+0,0005
random_deletion	0,6933	0,7483	0,8144	0,9306	0,7941	-0,0044
ocr_augmentation	0,6757	0,6269	0,6809	0,9289	0,7925	-0,0060
random_swap_words	0,5600	0,6324	0,7750	0,9325	0,7225	-0,0760
random_insertion	0,6571	0,7310	0,8263	0,9365	0,7210	-0,0775
word_dropout	0,5600	0,6212	0,7123	0,9205	0,7168	-0,0817
character_substitution	0,0000	0,6154	0,7799	0,9330	0,7047	-0,0938
random_case	0,6761	0,7310	0,8313	0,9261	0,6826	-0,1159
blank_noising	0,6269	0,7843	0,8295	0,9274	0,8029	+0,0044
to_lower_case	0,6479	0,7703	0,7750	0,9343	0,8033	+0,0048

Tab. 5.1: Výsledky modelu na datasete *D1* bez generovania syntetických dát.

Metóda augmentácie	**user_name**	**device_ip**	**crypto**	**description**	**Overall F1**	ΔF_1
Bez augmentácie	0,6667	0,7222	0,7673	0,9364	0,7985	0,0000
blank_noising	0,4681	0,6619	0,7643	0,9335	0,7291	-0,0694
character_substitution	0,7059	0,7483	0,7702	0,9335	0,7964	-0,0021
keyboard_augmentation	0,6571	0,7133	0,7702	0,9357	0,7925	-0,0060
modify_punctuation	0,6296	0,7347	0,7105	0,9345	0,7962	-0,0023
ocr_augmentation	0,6571	0,6809	0,7561	0,9333	0,7883	-0,0102
random_case	0,5600	0,7517	0,7750	0,9290	0,7914	-0,0071
random_deletion	0,6757	0,6619	0,8391	0,9323	0,7730	-0,0255
random_insertion	0,6667	0,6222	0,6857	0,9311	0,7813	-0,0172
random_swap	0,6571	0,6806	0,7722	0,9366	0,6390	-0,1595
random_swap_words	0,6667	0,6316	0,6906	0,9320	0,7901	-0,0084
remove_stop_words	0,6933	0,7432	0,6861	0,9234	0,7801	-0,0184
to_lower_case	0,5600	0,6165	0,6714	0,9295	0,7806	-0,0179
to_upper_case	0,6761	0,6471	0,8242	0,9359	0,7358	-0,0627
word_dropout	0,6571	0,7682	0,7879	0,9327	0,7998	+0,0013

Tab. 5.2: Výsledky modelu na datasete *D1* s generovaním syntetických dát.

Metóda augmentácie	**domain_src**	**device_ip**	**action**	**description**	**Overall F1**	ΔF_1
Bez augmentácie	0,9962	1,0000	0,9964	1,0000	0,9989	0,0000
blank_noising	1,0000	1,0000	0,9989	1,0000	0,9989	0,0000
character_substitution	0,9943	1,0000	0,9982	1,0000	0,9993	+0,0004
keyboard_augmentation	0,9423	0,9792	0,9324	0,9997	0,8365	-0,1624
modify_punctuation	0,0000	0,7785	0,7415	0,9326	0,8031	-0,1958
random_case	0,9276	0,9462	0,9575	0,9998	0,9178	-0,0811
random_deletion	0,8825	0,6000	0,8875	0,9992	0,8210	-0,1779
random_insertion	0,9693	0,7273	0,9143	0,9984	0,9129	-0,0860
random_swap	0,0000	0,7733	0,7528	0,9392	0,8045	-0,1944
random_swap_words	0,9924	0,9574	0,9949	1,0000	0,9983	-0,0006
remove_stop_words	0,9904	1,0000	0,9953	1,0000	0,9978	-0,0011
to_lower_case	0,9962	1,0000	0,9935	1,0000	0,9987	-0,0002
to_upper_case	0,9721	0,9574	0,8272	0,9997	0,7630	-0,2359
word_dropout	0,9437	0,8736	0,9407	0,9996	0,8602	-0,1387

Tab. 5.3: Výsledky modelu na datasete *D2* bez generovania syntetických dát.

Metóda augmentácie	**domain_src**	**device_ip**	**action**	**description**	**Overall F1**	ΔF_1
Bez augmentácie	0,9962	1,0000	0,9964	1,0000	0,9989	0,0000
blank_noising	0,0000	0,6619	0,7231	0,9335	0,7291	-0,2698
character_substitution	1,0000	1,0000	0,9942	0,9997	0,9988	-0,0001
keyboard_augmentation	0,9624	0,9792	0,9686	0,9995	0,9793	-0,0196
modify_punctuation	0,9962	0,9684	0,9686	0,9996	0,9221	-0,0768
ocr_augmentation	0,9933	0,9574	0,9763	0,9995	0,9870	-0,0119
random_case	1,0000	1,0000	0,9924	0,9999	0,9991	+0,0002
random_deletion	0,9835	0,9792	0,9670	0,9995	0,9243	-0,0746
random_insertion	1,0000	1,0000	0,9917	1,0000	0,9989	0,0000
random_swap	0,9253	0,4923	0,9008	0,9977	0,7786	-0,2203
random_swap_words	1,0000	1,0000	0,9956	0,9998	0,9993	+0,0004
remove_stop_words	0,9981	1,0000	0,9427	1,0000	0,9980	-0,0009
to_lower_case	1,0000	1,0000	0,9971	1,0000	0,9993	+0,0004
to_upper_case	0,9127	0,8471	0,8770	0,9991	0,8629	-0,1360
word_dropout	0,9715	1,0000	0,9935	0,9999	0,9265	-0,0724

Tab. 5.4: Výsledky modelu na datasete *D2* s generovaním syntetických dát.

Záver

Táto bakalárska práca sa zameriava na návrh a implementáciu nástroja na rozšírenie záznamov bezpečnostných udalostí s cieľom zlepšiť trénovanie neurónových sietí. V rámci práce boli podrobne analyzované súčasné techniky textovej augmentácie, ich možnosti aplikácie v oblasti kybernetickej bezpečnosti a výzvy súvisiace s ich implementáciou. Výsledkom je nástroj, ktorý umožňuje generovanie syntetických logov a ich rozšírenie pomocou moderných metód textovej augmentácie.

Teoretická časť tejto práce sa venuje analýze problematiky bezpečnostného monitoringu a umelej inteligencie. Podrobne opisuje kľúčové bezpečnostné pojmy, ktoré sú dôležité pre pochopenie základov kybernetickej bezpečnosti, a analyzuje kategorizáciu a formáty logov, ich spracovanie a význam v kontexte informačných systémov. Súčasťou teoretickej časti je aj prehľad technológií využívaných na bezpečnostný monitoring, ako aj metód umelej inteligencie a strojového učenia, ktoré sú aplikované pri detekcii hrozieb. Kľúčovým bodom je podrobná analýza techník textovej augmentácie a ich aplikácia na rozšírenie logových záznamov, pričom sa zdôrazňuje ich význam pre zlepšenie kvality a variability dát. V rámci analýzi bolo zanalyzovaných viac ako 50 článkov textovej augmentácie a posúdenie jednotlivych metod vramci logových záznamov.

Praktická časť nadväzovala na teoretické poznatky a realizovala návrh a implementáciu nástroja, ktorý umožňuje generovanie syntetických logov a ich následnú textovú augmentáciu. V rámci riešenia bolo vytvorených 75 generátorov využívajúcich moderných knižnic na generovanie syntetických dát a textových súborov. Tieto generátory boli navrhnuté tak, aby pokrývali rôzne typy entít vyskytujúcich sa v bezpečnostných logoch, ako sú IP adresy, používateľské mená, časové pečiatky či názvy služieb. Na základe analytickej časti bolo následne implementovaných 24 rôznych techník augmentácie, ktoré významne prispeli k zvýšeniu objemu a rozmanitosti trénovacích dát. Tieto techniky pokrývali všetky úrovne textovej augmentácie, od manipulácie so znakmi a slovami, cez štruktúrne zmeny viet, až po pokročilé modelovo riadené transformácie. Kombináciou generovania syntetických záznamov a ich následnej augmentácie vznikol nástroj, ktorý umožňuje vytváranie variabilných, realistických a účelovo prispôsobených dátových sád pre trénovanie modelov neurónových sietí.

Na základe testovania implementovaného nástroja bolo možné vyhodnotiť jeho efektivitu pri rozširovaní bezpečnostných logov pomocou moderných metód textovej augmentácie. Testovanie prebehlo na dvoch datasetoch, *D1* (obsahujúci rôznorodé logy z rôznych zariadení a systémov) a *D2* (špecifický pre bezpečnostné logy Cisco ASA). Výsledky testov potvrdili, že navrhnutý nástroj dokáže generovať syntetické dáta s vysokou variabilitou a štruktúrnou konzistenciou, čo vedie k zlepšeniu gene-

ralizácie neurónových sietí. Výsledky testovania teda potvrdzujú potenciál navrhnutého nástroja pri generovaní rozšírených logových datasetov a jeho použiteľnosť pri praktickej aplikácii v oblasti kybernetickej bezpečnosti. Zároveň naznačujú potrebu ďalšieho doladenia procesu mapovania entít a doplnenia techník na vyrovnanie ich zastúpenia pre ešte lepšie výsledky v budúcnosti.

V budúcnosti by bolo možné rozšíriť návrh nástroja o pokročilejšie augmentačné metódy, najmä metódy založené na využití predtrénovaných jazykových modelov (napríklad metódy typu synonymickú náhradu slov alebo generatívne modely), ktoré umožňujú meniť logové správy spôsobom, ktorý zachováva ich význam a štruktúru. Tým by sa ešte viac zvýšila variabilita trénovacích dát a súčasne by sa minimalizovalo riziko narušenia kľúčových entít, čo by pomohlo modelu naučiť sa robustnejšie reprezentácie. Ďalším krokom by mohlo byť aj vylepšenie samotných datasetov, konkrétne úpravou ich jednotných názvov metakľúčov. V súčasnosti nie je vždy zabezpečené, že všetky logy obsahujú jednotné pomenovanie pre rovnaké typy entít, čo môže viesť k nerovnomernému zastúpeniu jednotlivých kategórií počas trénovania modelu. Vyrovnanie počtu identifikovaných metakľúčov a ich konzistentné označovanie by významne prispelo k zlepšeniu kvality a reprezentatívnosti trénovacích dát, a tým aj k vyššej presnosti výsledných modelov.

Literatúra

1. ILLUSION PICTURES S.R.O. *Jakou roli hraje AI v kybernetické obraně?* [Online]. Algotech.cz, [b.r.]. Dostupné tiež z: https://www.algotech.cz/novinky/ai-a-kyberbezpecnost. [cit. 2025-03-28].
2. CHECK POINT SOFTWARE TECHNOLOGIES LTD. *A Closer Look at Q3 2024: 75% Surge in Cyber Attacks Worldwide* [Online]. Check Point Software Technologies Ltd., 2024. Dostupné tiež z: https://blog.checkpoint.com/research/a-closer-look-at-q3-2024-75-surge-in-cyber-attacks-worldwide/. [cit. 2025-02-19].
3. DUTT, Vinay. The Rise of The Machines: AI-Driven SIEM User Experience for Enhanced Decision-Making. *International Journal of Computer Engineering and Technology* [Online]. 2021. [cit. 2025-02-25].
4. IBM CORPORATION. *Cost of a Data Breach Report 2024* [Online]. IBM, 2024. Dostupné tiež z: https://www.ibm.com/reports/data-breach. [cit. 2025-02-25].
5. SANDERS, Chris. Chapter 1 - The Practice of Applied Network Security Monitoring. In: SANDERS, Chris; SMITH, Jason (ed.). *Applied Network Security Monitoring*. Boston: Syngress, 2014, s. 1–24. ISBN 978-0-12-417208-1. Dostupné z DOI: https://doi.org/10.1016/B978-0-12-417208-1.00001-5. [cit. 2024-10-07].
6. SANS INSTITUTE. *Glossary of Security Terms | SANS Institute* [Online]. SANS Institute, [b.r.]. Dostupné tiež z: https://www.sans.org/security-resources/glossary-of-terms/. [cit. 2024-10-07].
7. PJG. *CyberSecurity.CZ* [Online]. CyberSecurity.cz, 2024-2025. Dostupné tiež z: https://www.cybersecurity.cz/glossary.html. [cit. 2024-10-10].
8. NATIONAL INITIATIVE FOR CYBERSECURITY CAREERS AND STUDIES (NICCS). *Vocabulary | NICCS* [Online]. NICCS, 2024. Dostupné tiež z: https://niccs.cisa.gov/cybersecurity-career-resources/vocabulary. [cit. 2025-10-12].
9. SACRED HEART UNIVERSITY. *Cybersecurity and the Importance of Log Files* [Online]. Sacred Heart University, 2024. Dostupné tiež z: https://www.sacredheart.edu/academics/colleges--schools/school-of-computer-science--engineering/computer-science--cybersecurity-blog/cybersecurity-and-the-importance-of-log-files/. [cit. 2024-11-03].

10. SHARIF, ARFAN. *Log Files: Definition, Types, and Importance | CrowdStrike* [Online]. CrowdStrike, 2022. Dostupné tiež z: https://www.crowdstrike.com/en-us/cybersecurity-101/next-gen-siem/log-file/. [cit. 2024-11-23].

11. COBB, MICHAEL. *Security log management and logging best practices | TechTarget* [Online]. Search Security, 2023. Dostupné tiež z: https://www.techtarget.com/searchsecurity/tip/Security-log-management-and-logging-best-practices. [cit. 2024-11-03].

12. SEMATEXT GROUP, INC. *What Is a Log File: Definition & Types Explained* [Online]. Sematext, 2023. Dostupné tiež z: https://sematext.com/glossary/log-file/. [cit. 2024-11-23].

13. SEMATEXT GROUP, INC. *What Is Structured Logging and Why You Should Use It* [Online]. Sematext, [b.r.]. Dostupné tiež z: https://sematext.com/glossary/structured-logging/. [cit. 2024-11-28].

14. RAVI, Janani. *Structured Logging: Definition, Format, Benefits, and More* [Online]. Atatus, 2022. Dostupné tiež z: https://www.atatus.com/glossary/structured-logging/. [cit. 2024-11-28].

15. THE GRAYLOG TEAM. *Log Formats – a (Mostly) Complete Guide* [Online]. Graylog, 2020. Dostupné tiež z: http://graylog.org/post/log-formats-a-complete-guide/. [cit. 2024-11-29].

16. LOGICMONITOR. *What is syslog?* [Online]. LogicMonitor, 2024. Dostupné tiež z: https://www.logicmonitor.com/blog/what-is-syslog. [cit. 2024-11-29].

17. DARRINGTON, Jeff. *Syslog Protocol: A Reference Guide* [Online]. Graylog, 2025. Dostupné tiež z: https://graylog.org/post/syslog-protocol-a-reference-guide/. [cit. 2024-10-19].

18. MODERN SECOPS. *How to parse network messages with KQL in Sentinel* [Online]. Modern SecOps, 2024. Dostupné tiež z: https://modernsecops.com/p/network-message-parsing-kql. [cit. 2025-03-28].

19. SIDDIQUI, LAIBA. *What Is Syslog?* [Online]. Splunk, 2024. Dostupné tiež z: https://www.splunk.com/en_us/blog/learn/syslog.html. [cit. 2024-10-19].

20. OPENOBSERVE TEAM. *Understanding JSON Logging and Analysis* [Online]. 2024. Dostupné tiež z: https://openobserve.ai/blog/json-logging-guide-examples/. [cit. 2024-10-19].

21. MICRO FOCUS. *Micro Focus Security ArcSight Common Event Format* [Online]. Micro Focus, 2017. Dostupné tiež z: https://www.microfocus.com/documentation/arcsight/arcsight-smartconnectors/pdfdoc/common-event-format-v25/common-event-format-v25.pdf. [cit. 2024-11-02].

22. KENT, K.; SOUPPAYA, M. P. *Guide to Computer Security Log Management* [Online]. Gaithersburg, MD, 2006. Tech. spr., NIST SP 800-92. National Institute of Standards a Technology. Dostupné z DOI: 10.6028/NIST.SP.800-92. [cit. 2024-11-28].

23. SHARIF, ARFAN. *What is Log Management? 4 Best Practices & More | CrowdStrike* [Online]. CrowdStrike, 2022. Dostupné tiež z: https://www.crowdstrike.com/en-us/cybersecurity-101/next-gen-siem/log-management/. [cit. 2025-01-12].

24. ALANATA. *Bezpečnostný monitoring* [Online]. Alanata, 2022. Dostupné tiež z: https://www.alanata.sk/riesenia/bezpecnostny-monitoring/. [cit. 2024-11-15].

25. SHEERAZ, Muhammad; PARACHA, Muhammad Arsalan; HAQUE, Mansoor Ul; DURAD, Muhammad Hanif; MOHSIN, Syed Muhammad; BAND, Shahab S.; MOSAVI, Amir. Effective Security Monitoring using Efficient SIEM Architecture. 2023. Dostupné z DOI: 10.22967/HCIS.2023.13.023. [cit. 2024-11-15].

26. CISCO. *What Is SIEM? - Security Information and Event Management* [Online]. Cisco, 2024. Dostupné tiež z: https://www.cisco.com/c/en/us/products/security/what-is-siem.html. [cit. 2024-11-15].

27. SWANAGAN, Michael. *What Is A SIEM? Benefits, Tools, & Strategies* [Online]. PurpleSec, 2024. Dostupné tiež z: https://purplesec.us/learn/siem-solutions/. [cit. 2025-03-28].

28. MICROSOFT. *What Is EDR? Endpoint Detection and Response | Microsoft Security* [Online]. Microsoft, [b.r.]. Dostupné tiež z: https://www.microsoft.com/en-us/security/business/security-101/what-is-edr-endpoint-detection-response. [cit. 2024-11-20].

29. ARFEEN, Asad; AHMED, Saad; KHAN, Muhammad Asim; JAFRI, Syed Faraz Ali. Endpoint Detection & Response: A Malware Identification Solution. In: *2021 International Conference on Cyber Warfare and Security (ICCWS)*. Islamabad, Pakistan: IEEE, 2021, s. 1–8. Dostupné z DOI: 10.1109/ICCWS53234.2021.9703010. [cit. 2025-01-28].

30. GEORGE, A. Shaji; GEORGE, A. S. Hovan; BASKAR, T.; PANDEY, Digvijay. XDR: The Evolution of Endpoint Security Solutions – Superior Extensibility and Analytics to Satisfy the Organizational Needs of the Future. *Zenodo* [Online]. 2021. Dostupné z DOI: 10.5281/ZENODO.7028219. [cit. 2024-11-17].

31. MICROSOFT. *What Is SOAR? Technology and Solutions | Microsoft Security* [Online]. Microsoft, [b.r.]. Dostupné tiež z: https://www.microsoft.com/en/security/business/security-101/what-is-soar. [cit. 2024-11-06].

32. CYVATAR. *What is SOAR security orchestration and why is it important? | CYVATAR* [Online]. Cyvatar, 2022. Dostupné tiež z: https://cyvatar.ai/soar-security-orchestration-automation-response/. [cit. 2024-11-08].

33. IBM CORPORATION. *What is SOAR (security orchestration, automation and response)? | IBM* [Online]. IBM, 2023. Dostupné tiež z: https://www.ibm.com/think/topics/security-orchestration-automation-response. [cit. 2024-11-11].

34. PALO ALTO NETWORKS. *What Is SOAR?* [Online]. Palo Alto Networks, [b.r.]. Dostupné tiež z: https://www.paloaltonetworks.com/cyberpedia/what-is-soar. [cit. 2025-03-28].

35. IBM CORPORATION. *What is AI security? | IBM* [Online]. IBM, 2024. Dostupné tiež z: https://www.ibm.com/think/topics/ai-security. [cit. 2025-01-18].

36. CYBLE INC. *Real-Time Threat Detection Using The Power Of AI - Cyble* [Online]. Cyble, 2024. Dostupné tiež z: https://cyble.com/knowledge-hub/real-time-threat-detection-with-ai/. [cit. 2025-01-18].

37. NUNNAGUPPALA, Laxmi Sarat Chandra. A Future-Proof Approach to Cybersecurity Compliance: The Power of AI and ML in SIEM, SOAR, and Cloud SOC [Online]. 2023. Dostupné tiež z: https://www.researchgate.net/publication/381902931_A_FUTURE-PROOF_APPROACH_TO_CYBERSECURITY_COMPLIANCE_THE_POWER_OF_AI_AND_ML_IN_SIEM_SOAR_AND_CLOUD_SOC. [cit. 2025-05-07].

38. BINHAMMAD, Mohammad; ALQAYDI, Shaikha; OTHMAN, Azzam; ABULJADAYEL, Laila Hatim. The Role of AI in Cyber Security: Safeguarding Digital Identity. *Journal of Information Security* [Online]. 2024, roč. 15, č. 2, s. 245–278. Dostupné z DOI: 10.4236/jis.2024.152015. [cit. 2025-04-18].

39. ZSCALER, INC. AND CYBERSECURITY INSIDERS. *Unveiling the Potential: Artificial Intelligence in Cybersecurity* [Online]. Zscaler, 2023. Dostupné tiež z: https://info.zscaler.com/resources-industry-reports-role-of-artificial-intelligence-in-cybersecurity-2023-report-thank-you. [cit. 2025-04-19].

40. HASHEMI-POUR, Cameron; BARNEY, Nick. *What Is Named Entity Recognition (NER)?* [Online]. 2024-10. Dostupné tiež z: https://www.techtarget.com/whatis/definition/named-entity-recognition-NER. [cit. 2025-05-17].

41. SAP SE. *Strojové učenie | Definícia, typy a príklady* [Online]. SAP, [b.r.]. Dostupné tiež z: https://www.sap.com/sk/products/artificial-intelligence/what-is-machine-learning.html. [cit. 2025-03-28].

42. ŠŤASTNÁ, Ariela. *Security Log Anonymization Tool Focusing on Artificial Intelligence Techniques* [Online]. 2023. Dostupné tiež z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=255254. [cit. 2024-11-03].

43. IBM CORPORATION. *What Is Machine Learning (ML)? | IBM* [Online]. IBM, 2021. Dostupné tiež z: https://www.ibm.com/think/topics/machine-learning. [cit. 2025-03-28].

44. CLOUDFLARE. *What is a neural network?* [Online]. Cloudflare, [b.r.]. Dostupné tiež z: https://www.cloudflare.com/en-gb/learning/ai/what-is-neural-network/. [cit. 2024-11-17].

45. AMAZON WEB SERVICES, INC. *What is a Neural Network? - Artificial Neural Network Explained - AWS* [Online]. Amazon Web Services, [b.r.]. Dostupné tiež z: https://aws.amazon.com/what-is/neural-network/. [cit. 2025-03-28].

46. MUMUNI, Alhassan; MUMUNI, Fuseini. Data augmentation: A comprehensive survey of modern approaches. *Array*. 2022, roč. 16, s. 100258. Dostupné z DOI: 10.1016/j.array.2022.100258. [cit. 2025-03-28].

47. SHARIF, ARFAN. *Log Parsing: What Is It and How Does It Work? | CrowdStrike* [Online]. CrowdStrike, 2022. Dostupné tiež z: https://www.crowdstrike.com/en-us/cybersecurity-101/next-gen-siem/log-parsing/. [cit. 2025-03-28].

48. NOVOGRODER, Idan. *Data Preprocessing in Machine Learning: Steps & Best Practices* [Online]. lakeFS, 2024. Dostupné tiež z: https://lakefs.io/blog/data-preprocessing-in-machine-learning/. [cit. 2025-05-03].

49. MOESKER, Nick. *What is Data Preparation for Machine Learning?* [Online]. DataNorth, 2024. Dostupné tiež z: https://datanorth.ai/blog/what-is-data-preparation. [cit. 2025-05-01].

50. MAHARANA, Kiran; MONDAL, Surajit; NEMADE, Bhushankumar. A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*. 2022, roč. 3, č. 1, s. 91–99. Dostupné z DOI: 10.1016/j.gltp.2022.04.020. [cit. 2025-03-28].

51. IBM CORPORATION. *What is Data Transformation? | IBM* [Online]. IBM, 2024. Dostupné tiež z: https://www.ibm.com/think/topics/data-transformation. [cit. 2025-03-28].

52. AIRBYTE. *A Deep Dive Into Data Transformation for Data Engineers | Airbyte* [Online]. Airbyte, 2025. Dostupné tiež z: https://airbyte.com/data-engineering-resources/data-transformation. [cit. 2025-03-18].

53. IBM CORPORATION. *What is data augmentation? | IBM* [Online]. IBM, 2024. Dostupné tiež z: https://www.ibm.com/think/topics/data-augmentation. [cit. 2025-02-01].

54. NGUYEN, Quan. *Understanding Training, Validation, and Testing Data in ML* [Online]. Eastgate Software, 2024. Dostupné tiež z: https://eastgate-software.com/understanding-training-validation-and-testing-data-in-ml/. [cit. 2025-02-01].

55. DOWLING, Jim. *Guide to File Formats for Machine Learning: Columnar, Training, Inferencing, and the Feature Store* [Online]. TDS Archive, 2019. Dostupné tiež z: https://medium.com/data-science/guide-to-file-formats-for-machine-learning-columnar-training-inferencing-and-the-feature-store-2e0c3d18d4f9. [cit. 2025-05-01].

56. RAND, Chaim. *Data Formats for Training in TensorFlow: Parquet, Petastorm, Feather, and More* [Online]. TDS Archive, 2021. Dostupné tiež z: https://medium.com/data-science/data-formats-for-training-in-tensorflow-parquet-petastorm-feather-and-more-e55179eeeb72. [cit. 2025-02-10].

57. CHEN, Michael. *What is AI model training and why is it important?* [Online]. Oracle, 2023. Dostupné tiež z: https://www.oracle.com/ca-en/artificial-intelligence/ai-model-training/. [cit. 2025-05-01].

58. TANISHA.DIGITAL. *Key Evaluation Metrics For AI Model Performance* [Online]. Gen AI Adventures, 2024. Dostupné tiež z: https://medium.com/gen-ai-adventures/key-evaluation-metrics-for-ai-model-performance-8e372f17a0a2. [cit. 2025-05-01].

59. AMAZON WEB SERVICES. *What is Data Augmentation? - Data Augmentation Techniques Explained - AWS* [Online]. Amazon Web Services, [b.r.]. Dostupné tiež z: https://aws.amazon.com/what-is/data-augmentation/. [cit. 2025-05-07].

60. KÄPPEL, Martin; JABLONSKI, Stefan. Model-Agnostic Event Log Augmentation for Predictive Process Monitoring. In: INDULSKA, Marta; REINHARTZ-BERGER, Iris; CETINA, Carlos; PASTOR, Oscar (ed.). *Advanced Information Systems Engineering*. Cham: Springer Nature Switzerland, 2023, zv. 13901, s. 381–397. ISBN 9783031345593 9783031345609. Dostupné z DOI: 10.1007/978-3-031-34560-9_23. [cit. 2025-03-28].

61. BAYER, Markus; KAUFHOLD, Marc-André; REUTER, Christian. *A Survey on Data Augmentation for Text Classification* [Online]. arXiv, 2022. Dostupné z DOI: 10.48550/arXiv.2107.03158. [cit. 2025-03-18].

62. SHORTEN, Connor; KHOSHGOFTAAR, Taghi M.; FURHT, Borko. Text Data Augmentation for Deep Learning. *Journal of Big Data*. 2021, roč. 8, č. 1, s. 101. Dostupné z DOI: 10.1186/s40537-021-00492-0. [cit. 2025-03-18].

63. LI, Bohan; HOU, Yutai; CHE, Wanxiang. *Data Augmentation Approaches in Natural Language Processing: A Survey* [Online]. arXiv, 2022. Dostupné z DOI: 10.48550/arXiv.2110.01852. [cit. 2025-03-18].

64. BELINKOV, Yonatan; BISK, Yonatan. *Synthetic and Natural Noise Both Break Neural Machine Translation* [Online]. arXiv, 2018. Dostupné z DOI: 10.48550/arXiv.1711.02173. [cit. 2025-03-18].

65. FENG, Steven Y.; GANGAL, Varun; KANG, Dongyeop; MITAMURA, Teruko; HOVY, Eduard. GenAug: Data Augmentation for Finetuning Text Generators. In: AGIRRE, Eneko; APIDIANAKI, Marianna; VULIĆ, Ivan (ed.). *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Online: Association for Computational Linguistics, 2020, s. 29–42. Dostupné z DOI: 10.18653/v1/2020.deelio-1.4. [cit. 2025-03-18].

66. DAI, Haixing; LIU, Zhengliang; LIAO, Wenxiong; HUANG, Xiaoke; CAO, Yihan; WU, Zihao; ZHAO, Lin; XU, Shaochen; LIU, Wei; LIU, Ninghao; LI, Sheng; ZHU, Dajiang; CAI, Hongmin; SUN, Lichao; LI, Quanzheng; SHEN, Dinggang; LIU, Tianming; LI, Xiang. *AugGPT: Leveraging ChatGPT for Text Data Augmentation* [Online]. arXiv, 2023. Dostupné z DOI: 10.48550/arXiv.2302.13007. [cit. 2025-03-13].

67. EBRAHIMI, Javid; RAO, Anyi; LOWD, Daniel; DOU, Dejing. *HotFlip: White-Box Adversarial Examples for Text Classification* [Online]. arXiv, 2018. Dostupné z DOI: 10.48550/arXiv.1712.06751. [cit. 2025-03-18].

68. COULOMBE, Claude. *Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs* [Online]. arXiv, 2018. Dostupné z DOI: 10.48550/arXiv.1812.04718. [cit. 2025-03-28].

69. XIE, Ziang; WANG, Sida I.; LI, Jiwei; LÉVY, Daniel; NIE, Aiming; JURAFSKY, Dan; NG, Andrew Y. *Data Noising as Smoothing in Neural Network Language Models* [Online]. arXiv, 2017. Dostupné z DOI: 10.48550/arXiv.1703.02573. [cit. 2025-03-20].

70. LI, Yitong; COHN, Trevor; BALDWIN, Timothy. Robust Training under Linguistic Adversity. In: LAPATA, Mirella; BLUNSOM, Phil; KOLLER, Alexander (ed.). *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, 2017, s. 21–27. Dostupné tiež z: https://aclanthology.org/E17-2004/. [cit. 2025-03-11].

71. WEI, Jason; ZOU, Kai. *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks* [Online]. arXiv, 2019. Dostupné z DOI: 10.48550/arXiv.1901.11196. [cit. 2025-03-18].

72. KAPUSTA, Jozef; DRŽÍK, Dávid; ŠTEFLOVIČ, Kirsten; NAGY, Kitti Szabó. Text Data Augmentation Techniques for Word Embeddings in Fake News Classification. *IEEE Access*. 2024, roč. 12, s. 31538–31550. Dostupné z DOI: 10.1109/ACCESS.2024.3369918. [cit. 2025-03-13].

73. RIZOS, Georgios; HEMKER, Konstantin; SCHULLER, Björn. Augment to Prevent: Short-Text Data Augmentation in Deep Learning for Hate-Speech Classification. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. Beijing, China: ACM, 2019, s. 991–1000. Dostupné z DOI: 10.1145/3357384.3358040. [cit. 2025-03-18].

74. SUN, Xiao; HE, Jiajin. A novel approach to generate a large scale of supervised data for short text sentiment analysis. *Multimed Tools Appl*. 2020, roč. 79, č. 9, s. 5439–5459. Dostupné z DOI: 10.1007/s11042-018-5748-4. [cit. 2025-03-18].

75. CHOI, Seungtaek; JEONG, Myeongho; HAN, Hojae; HWANG, Seung-won. C2L: Causally Contrastive Learning for Robust Text Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022, roč. 36, č. 10, s. 10526–10534. Dostupné z DOI: 10.1609/aaai.v36i10.21296. [cit. 2025-03-27].

76. ZHANG, Xiang; ZHAO, Junbo; LECUN, Yann. Character-level Convolutional Networks for Text Classification. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015, zv. 28. Dostupné tiež z: https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html. [cit. 2025-03-19].

77. ZAITON, Hoda; AL-ANSARY, Sameh. Natural Language Processing Approaches to Text Data Augmentation: A Computational Linguistic Analysis. *IJAES*. 2025, roč. 25, č. 1, s. 99–124. Dostupné z DOI: 10.33806/ijaes.v25i1.682. [cit. 2025-03-18].

78. WANG, William Yang; YANG, Diyi. That's So Annoying!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets. In: MÀRQUEZ, Lluís; CALLISON-BURCH, Chris; SU, Jian (ed.). *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, 2015, s. 2557–2563. Dostupné z DOI: 10.18653/v1/D15-1306. [cit. 2025-02-10].

79. MRKŠIĆ, Nikola; Ó SÉAGHDHA, Diarmuid; THOMSON, Blaise; GAŠIĆ, Milica; ROJAS-BARAHONA, Lina M.; SU, Pei-Hao; VANDYKE, David; WEN, Tsung-Hsien; YOUNG, Steve. Counter-fitting Word Vectors to Linguistic Constraints. In: KNIGHT, Kevin; NENKOVA, Ani; RAMBOW, Owen (ed.). *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, 2016, s. 142–148. Dostupné z DOI: 10.18653/v1/N16-1018. [cit. 2025-02-10].

80. ALZANTOT, Moustafa; SHARMA, Yash; ELGOHARY, Ahmed; HO, Bo-Jhang; SRIVASTAVA, Mani; CHANG, Kai-Wei. Generating Natural Language Adversarial Examples. In: RILOFF, Ellen; CHIANG, David; HOCKENMAIER, Julia; TSUJII, Jun'ichi (ed.). *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, s. 2890–2896. Dostupné z DOI: 10.18653/v1/D18-1316. [cit. 2025-02-10].

81. WU, Xing; LV, Shangwen; ZANG, Liangjun; HAN, Jizhong; HU, Songlin. *Conditional BERT Contextual Augmentation* [Online]. arXiv, 2018. Dostupné z DOI: 10.48550/arXiv.1812.06705. [cit. 2025-03-18].

82. HU, Zhiting; TAN, Bowen; SALAKHUTDINOV, Ruslan; MITCHELL, Tom; XING, Eric P. *Learning Data Manipulation for Augmentation and Weighting*

[Online]. arXiv, 2019. Dostupné z DOI: 10.48550/arXiv.1910.12795. [cit. 2025-03-18].

83. JIAO, Xiaoqi; YIN, Yichun; SHANG, Lifeng; JIANG, Xin; CHEN, Xiao; LI, Linlin; WANG, Fang; LIU, Qun. *TinyBERT: Distilling BERT for Natural Language Understanding* [Online]. arXiv, 2019. Dostupné tiež z: https://arxiv.org/abs/1909.10351v5. [cit. 2025-04-01].

84. MIN, Junghyun; MCCOY, R. Thomas; DAS, Dipanjan; PITLER, Emily; LINZEN, Tal. *Syntactic Data Augmentation Increases Robustness to Inference Heuristics* [Online]. arXiv, 2020. Dostupné z DOI: 10.48550/arXiv.2004.11999. [cit. 2025-04-02].

85. GUO, Hongyu; MAO, Yongyi; ZHANG, Richong. *Augmenting Data with Mixup for Sentence Classification: An Empirical Study* [Online]. arXiv, 2019. Dostupné tiež z: https://arxiv.org/abs/1905.08941v1. [cit. 2025-03-15].

86. FENG, Steven Y.; LI, Aaron W.; HOEY, Jesse. Keep Calm and Switch On! Preserving Sentiment and Fluency in Semantic Text Exchange. In: INUI, Kentaro; JIANG, Jing; NG, Vincent; WAN, Xiaojun (ed.). *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, s. 2701–2711. Dostupné z DOI: 10.18653/v1/D19-1272. [cit. 2025-03-12].

87. ŞAHIN, Gözde Gül; STEEDMAN, Mark. Data Augmentation via Dependency Tree Morphing for Low-Resource Languages. In: RILOFF, Ellen; CHIANG, David; HOCKENMAIER, Julia; TSUJII, Jun'ichi (ed.). *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, s. 5004–5009. Dostupné z DOI: 10.18653/v1/D18-1545. [cit. 2025-03-16].

88. HARALABOPOULOS, Giannis; TORRES, Mercedes Torres; ANAGNOSTOPOULOS, Ioannis; MCAULEY, Derek. Text data augmentations: Permutation, antonyms and negation. *Expert Systems with Applications*. 2021, roč. 177, s. 114769. ISSN 0957-4174. Dostupné z DOI: 10.1016/j.eswa.2021.114769. [cit. 2025-04-05].

89. SHI, Haoyue; LIVESCU, Karen; GIMPEL, Kevin. Substructure Substitution: Structured Data Augmentation for NLP. In: ZONG, Chengqing; XIA, Fei; LI, Wenjie; NAVIGLI, Roberto (ed.). *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational

Linguistics, 2021, s. 3494–3508. Dostupné z DOI: 10.18653/v1/2021.findings-acl.307. [cit. 2025-04-25].

90. KIM, Hazel; WOO, Daecheol; OH, Seong Joon; CHA, Jeong-Won; HAN, Yo-Sub. *ALP: Data Augmentation using Lexicalized PCFGs for Few-Shot Text Classification* [Online]. arXiv, 2021. Dostupné z DOI: 10.48550/arXiv.2112.11916. [cit. 2025-04-26].

91. YU, Adams Wei; DOHAN, David; LUONG, Minh-Thang; ZHAO, Rui; CHEN, Kai; NOROUZI, Mohammad; LE, Quoc V. *QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension* [Online]. arXiv, 2018. Dostupné z DOI: 10.48550/arXiv.1804.09541. [cit. 2025-02-06].

92. GUPTA, Parul; MAHMOOD, Maha. DATA AUGMENTATION FOR NATURAL LANGUAGE PROCESSING. *International Journal of Data Science and Advanced Analytics*. 2024, roč. 6, č. 6, s. 352–359. Dostupné z DOI: 10.69511/ijdsaa.v6i6.239. [cit. 2025-01-18].

93. XIE, Qizhe; DAI, Zihang; HOVY, Eduard; LUONG, Minh-Thang; LE, Quoc V. *Unsupervised Data Augmentation for Consistency Training* [Online]. arXiv, 2020. Dostupné z DOI: 10.48550/arXiv.1904.12848. [cit. 2025-02-18].

94. NUGENT, Tim; STELEA, Nicole; LEIDNER, Jochen L. Detecting Environmental, Social and Governance (ESG) Topics Using Domain-Specific Language Models and Data Augmentation. In: [Online]. Springer International Publishing, 2021. Dostupné z DOI: 10.1007/978-3-030-86967-0_12. [cit. 2025-03-20].

95. QU, Yanru; SHEN, Dinghan; SHEN, Yelong; SAJEEV, Sandra; HAN, Jiawei; CHEN, Weizhu. *CoDA: Contrast-enhanced and Diversity-promoting Data Augmentation for Natural Language Understanding* [Online]. arXiv, 2020. Dostupné z DOI: 10.48550/arXiv.2010.08670. [cit. 2025-03-20].

96. MIYATO, Takeru; DAI, Andrew M.; GOODFELLOW, Ian. *Adversarial Training Methods for Semi-Supervised Text Classification* [Online]. arXiv, 2021. Dostupné z DOI: 10.48550/arXiv.1605.07725. [cit. 2025-05-12].

97. KUMAR, Varun; GLAUDE, Hadrien; LICHY, Cyprien de; CAMPBELL, William. *A Closer Look At Feature Space Data Augmentation For Few-Shot Intent Classification* [Online]. arXiv, 2019. Dostupné z DOI: 10.48550/arXiv.1910.04176. [cit. 2025-05-11].

98. COMBETTES, Patrick L.; PESQUET, Jean-Christophe. Proximal Splitting Methods in Signal Processing. In: [Online]. Springer, 2011. Dostupné z DOI: 10.1007/978-1-4419-9569-8_10. [cit. 2025-05-03].

99. ZHU, Chen; CHENG, Yu; GAN, Zhe; SUN, Siqi; GOLDSTEIN, Tom; LIU, Jingjing. *FreeLB: Enhanced Adversarial Training for Natural Language Understanding* [Online]. arXiv, 2020. Dostupné z DOI: 10.48550/arXiv.1909.11764. [cit. 2025-05-03].

100. JIANG, Haoming; HE, Pengcheng; CHEN, Weizhu; LIU, Xiaodong; GAO, Jianfeng; ZHAO, Tuo. *SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization* [Online]. arXiv, 2021. Dostupné z DOI: 10.48550/arXiv.1911.03437. [cit. 2025-05-03].

101. WANG, Dilin; GONG, Chengyue; LIU, Qiang. *Improving Neural Language Modeling via Adversarial Training* [Online]. arXiv, 2019. Dostupné z DOI: 10.48550/arXiv.1906.03805. [cit. 2025-05-02].

102. LIU, Xiaodong; CHENG, Hao; HE, Pengcheng; CHEN, Weizhu; WANG, Yu; POON, Hoifung; GAO, Jianfeng. *Adversarial Training for Large Neural Language Models* [Online]. arXiv, 2020. Dostupné z DOI: 10.48550/arXiv.2004.08994. [cit. 2025-05-13].

103. SHEN, Dinghan; ZHENG, Mingzhi; SHEN, Yelong; QU, Yanru; CHEN, Weizhu. *A Simple but Tough-to-Beat Data Augmentation Approach for Natural Language Understanding and Generation* [Online]. arXiv, 2020. Dostupné z DOI: 10.48550/arXiv.2009.13818. [cit. 2025-05-13].

104. ANABY-TAVOR, Ateret; CARMELI, Boaz; GOLDBRAICH, Esther; KANTOR, Amir; KOUR, George; SHLOMOV, Segev; TEPPER, Naama; ZWERDLING, Naama. *Not Enough Data? Deep Learning to the Rescue!* [Online]. arXiv, 2019. Dostupné z DOI: 10.48550/arXiv.1911.03118. [cit. 2025-05-13].

105. QIU, Siyuan; XU, Binxia; ZHANG, Jie; WANG, Yafang; SHEN, Xiaoyu; DE MELO, Gerard; LONG, Chong; LI, Xiaolong. EasyAug: An Automatic Textual Data Augmentation Platform for Classification Tasks. In: *Companion Proceedings of the Web Conference 2020*. Taipei Taiwan: ACM, 2020, s. 249–252. Dostupné z DOI: 10.1145/3366424.3383552. [cit. 2025-03-23].

106. MALANDRAKIS, Nikolaos; SHEN, Minmin; GOYAL, Anuj; GAO, Shuyang; SETHI, Abhishek; METALLINOU, Angeliki. Controlled Text Generation for Data Augmentation in Intelligent Artificial Agents. In: *Proceedings of the 3rd Workshop on Neural Generation and Translation*. Hong Kong: Association for Computational Linguistics, 2019, s. 90–98. Dostupné z DOI: 10.18653/v1/D19-5609. [cit. 2025-03-24].

107. GUU, Kelvin; HASHIMOTO, Tatsunori B.; OREN, Yonatan; LIANG, Percy. *Generating Sentences by Editing Prototypes* [Online]. arXiv, 2018. Dostupné z DOI: 10.48550/arXiv.1709.08878. [cit. 2025-03-24].

108. RAILLE, Guillaume; DJAMBAZOVSKA, Sandra; MUSAT, Claudiu. *Fast Cross-domain Data Augmentation through Neural Sentence Editing* [Online]. arXiv, 2020. Dostupné z DOI: 10.48550/arXiv.2003.10254. [cit. 2025-03-24].

109. OLLAGNIER, Anaïs; WILLIAMS, Hywel T. P. Text Augmentation Techniques for Clinical Case Classification. In: *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum*. Thessaloniki, Greece, 2020. Dostupné tiež z: https://ceur-ws.org/Vol-2696/paper_166.pdf. [cit. 2025-03-28].

110. LI, Yang; PAN, Quan; WANG, Suhang; YANG, Tao; CAMBRIA, Erik. A Generative Model for category text generation. *Information Sciences*. 2018, roč. 450, s. 301–315. Dostupné z DOI: 10.1016/j.ins.2018.03.050. [cit. 2025-03-28].

111. WANG, Congcong; LILLIS, David. Classification for Crisis-Related Tweets Leveraging Word Embeddings and Data Augmentation. *Proceedings of the Twenty-Eighth Text REtrieval Conference (TREC 2019)*. 2019. Dostupné tiež z: http://hdl.handle.net/10197/25817. [cit. 2025-04-28].

112. YOO, Kang Min; PARK, Dongju; KANG, Jaewook; LEE, Sang-Woo; PARK, Woomyeong. *GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation* [Online]. arXiv, 2021. Dostupné z DOI: 10.48550/arXiv.2104.08826. [cit. 2025-04-28].

113. LEE, Kenton; GUU, Kelvin; HE, Luheng; DOZAT, Tim; CHUNG, Hyung Won. *Neural Data Augmentation via Example Extrapolation* [Online]. arXiv, 2021. Dostupné z DOI: 10.48550/arXiv.2102.01335. [cit. 2025-04-27].

114. KOBAYASHI, Sosuke. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In: Association for Computational Linguistics, 2018. Dostupné z DOI: 10.18653/v1/N18-2072. [cit. 2025-04-24].

115. LIU, Sisi; LEE, Kyungmi; LEE, Ickjai. Document-level multi-topic sentiment classification of Email data with BiLSTM and data augmentation. *Knowledge-Based Systems*. 2020, roč. 197, s. 105918. Dostupné z DOI: 10.1016/j.knosys.2020.105918. [cit. 2025-04-24].

116. PENG, Baolin; ZHU, Chenguang; ZENG, Michael; GAO, Jianfeng. *Data Augmentation for Spoken Language Understanding via Pretrained Language Models* [Online]. arXiv, 2021. Dostupné z DOI: 10.48550/arXiv.2004.13952. [cit. 2025-04-25].

117. REGINA, Mehdi; MEYER, Maxime; GOUTAL, Sébastien. *Text Data Augmentation: Towards better detection of spear-phishing emails* [Online]. arXiv, 2021. Dostupné z DOI: 10.48550/arXiv.2007.02033. [cit. 2025-04-26].

118. LONGPRE, Shayne; WANG, Yu; DUBOIS, Christopher. *How Effective is Task-Agnostic Data Augmentation for Pretrained Transformers?* [Online]. arXiv, 2020. Dostupné z DOI: 10.48550/arXiv.2010.01764. [cit. 2025-04-26].

119. RASTOGI, Chetanya; MOFID, Nikka; HSIAO, Fang-I. *Can We Achieve More with Less? Exploring Data Augmentation for Toxic Comment Classification* [Online]. arXiv, 2020. Dostupné tiež z: https://arxiv.org/abs/2007.00875v1. [cit. 2025-04-26].

120. YU, Shujuan; YANG, Jie; LIU, Danlei; LI, Runqi; ZHANG, Yun; ZHAO, Shengmei. Hierarchical Data Augmentation and the Application in Text Classification. *IEEE Access*. 2019, roč. 7, s. 185476–185485. Dostupné z DOI: 10.1109/ACCESS.2019.2960263. [cit. 2025-04-27].

121. DAS, Manoj. *Faker: Python is Just a Fake Away!* [Online]. Medium, 2023. Dostupné tiež z: https://medium.com/@HeCanThink/faker-python-is-just-a-fake-away-ef626a0dcf8d. [cit. 2025-03-28].

122. L. *Generating mock data using Mimesis: Part I* [Online]. wemake.services, 2017. Dostupné tiež z: https://medium.com/wemake-services/generating-mock-data-using-elizabeth-part-i-ca5a55b8027c. [cit. 2025-03-28].

123. FAUXFACTORY DEVELOPERS. *API Documentation — FauxFactory 3.1.1 documentation* [Online]. 2023. Dostupné tiež z: https://fauxfactory.readthedocs.io/en/latest/api.html. [cit. 2025-03-25].

124. OPENAI. *ChatGPT (GPT-4)* [Online]. 2025. Dostupné tiež z: https://chat.openai.com/. [cit. 2025-05-09].

125. FOLTÝN, Ondřej. *Aplikace pokročilých technik rozšíření datových sad integrujících metody strojového učení pro účely syntaktické analýzy bezpečnostních logů* [Online]. Brno, 2025 [cit. 2025-05-27]. Dostupné z : https://www.vut.cz/studenti/zav-prace/detail/167295. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií.

Zoznam symbolov a skratiek

AI *Artificial Intelligence* – umelá inteligencia

ALUM *Adversarial Training with Learned Uncertainty Minimization* – robustné trénovanie pomocou optimalizácie neistoty

API *Application Programming Interface* – aplikačné programové rozhranie

APT *Advanced Persistent Threat* – pokročilá pretrvávajúca hrozba

BERT *Bidirectional Encoder Representations from Transformers* – jazykový model založený na transformeroch

C-BERT *Contextual BERT* – kontextový variant modelu BERT

CEF *Common Event Format* – formát spoločných udalostí

CNN *Convolutional Neural Network* – konvolučná neurónová sieť

CS-GAN *Conditional Sequence Generative Adversarial Network* – generatívna sieť podmienených sekvencií

CSV *Comma-Separated Values* – hodnoty oddelené čiarkou

DDoS *Distributed Denial of Service* – distribuovaný útok odmietnutia služby

DistilBERT *Distilled BERT* – zjednodušený model BERT

EDA *Easy Data Augmentation* – jednoduchá augmentácia textu

EDR *Endpoint Detection and Response* – detekcia a reakcia na koncových zariadeniach

FreeLB *Free Large-Batch Adversarial Training* – trénovanie na veľkých dávkach s nepriateľskými vstupmi

GANs *Generative Adversarial Networks* – generatívne adversariálne siete

GPT *Generative Pre-trained Transformer* – generatívny jazykový model

GUI *Graphical User Interface* – grafické užívateľské rozhranie

ID *Identifier* – identifikátor

IDS/IPS *Intrusion Detection System / Intrusion Prevention System* – systém detekcie / prevencie prienikov

IoC *Indicator of Compromise* – indikátor kompromitácie

JSON *JavaScript Object Notation* – zápis objektov v jazyku JavaScript

KL divergencia *Kullback-Leibler Divergence* – mierka rozdielu medzi pravdepodobnostnými rozdeleniami

KVP *Key-Value Pair* – dvojica kľúč-hodnota

k-NN *k-Nearest Neighbors* – algoritmus k najbližších susedov

Logfmt *Log Format* – formát logu

LSTM *Long Short-Term Memory* – pamäť dlhého a krátkeho trvania

LSTM-CNN kombinácia *Long Short-Term Memory* a *Convolutional Neural Network*

ML *Machine Learning* – strojové učenie

MNLI *Multi-Genre Natural Language Inference* – benchmark pre klasifikáciu významu viet

MTTC *Mean Time to Contain* – priemerný čas na zvládnutie incidentu

MTTI *Mean Time to Identify* – priemerný čas na identifikáciu incidentu

NDR *Network Detection and Response* – detekcia a reakcia na sieťovej vrstve

NER *Named Entity Recognition* – rozpoznávanie pomenovaných entít

NLP *Natural Language Processing* – spracovanie prirodzeného jazyka

OCR *Optical Character Recognition* – optické rozpoznávanie znakov

PID *Process Identifier* – identifikátor procesu

PGD *Projected Gradient Descent* – technika tvorby adversariálnych príkladov

POS-tagging *Part-of-Speech Tagging* – označovanie slovných druhov

PPDB *Paraphrase Database* – databáza parafráz

RBN *Rogue Botnet* – podvodná botnetová sieť

RNN *Recurrent Neural Network* – rekurentná neurónová sieť

ROC-AUC *Receiver Operating Characteristic – Area Under Curve* – metriku klasifikátorov

RoBERTa *Robustly optimized BERT approach* – robustná verzia BERT modelu

seqGAN *Sequence Generative Adversarial Network* – generatívna sieť pre sekvencie

SIEM *Security Information and Event Management* – správa bezpečnostných informácií a udalostí

SMART *Sharpness-Aware Minimization for Adversarial Robustness Training* – robustné trénovanie s ohľadom na ostrosť

SOC *Security Operations Center* – centrum bezpečnostných operácií

SOAR *Security Orchestration, Automation and Response* – orchestrácia, automatizácia a reakcia

Syslog *System Logging* – systémový logovací protokol

TCP *Transmission Control Protocol* – riadiaci prenosový protokol

TF-IDF *Term Frequency-Inverse Document Frequency* – váhovanie slov podľa výskytu

UDP *User Datagram Protocol* – užívateľský datagramový protokol

UEBA *User and Entity Behavior Analytics* – analýza správania používateľov a entít

UDA *Unsupervised Data Augmentation* – neznámkovaná dátová augmentácia

VAEs *Variational Autoencoders* – variačné autoenkódery

XDR *Extended Detection and Response* – rozšírená detekcia a reakcia

A Návod na spustenie programu

Táto kapitola slúži ako praktický návod na použitie nástroja `pm-log-augmentator`, ktorý slúži na augmentáciu logovacích záznamov, teda na ich rozšírenie, obohatenie a modifikáciu pomocou rôznych transformačných techník. Nástroj obsahuje dva ďalšie submoduly `pm_log_utils` a `pm_deep_augmentation`, sú potrebné na logovanie a augmentáciu pomocou hlbokého učenia, ktoré boli vyvinuté mimo ciele bakalárskej práce, je potrebné ich naimportovať, prosím, obráťte sa na vedúceho práce.

Import balíka

Nástroj `pm-log-augmentator` je distribuovaný ako Python balík, ktorý je možné nainštalovať z privátneho PyPI repozitára. Pre jeho úspešnú inštaláciu použite nasledujúci príkaz (na samotnú inštaláciu je potrebný prístup k projektu na adrese `https://viki.vmware.fekt.cz/mv-sectech`, prosím, kontaktujte vedúceho práce):

```
pip install --trusted-host viki.vmware.fekt.cz
--extra-index-url https://viki.vmware.fekt.cz/api/v4/projects/35/
    packages/pypi/simple sectech-log-augmentator==1.7.3
```

Spustenie hlavného skriptu

Pred spustením programu je potrebné nainštalovať všetky potrebné závislosti uvedené v súbore `requirements.txt`:

```
pip install -r requirements.txt
```

Nástroj je možné spustiť v dvoch režimoch — testovacom alebo produkčnom (s reálnym log súborom). Pri spúšťaní hlavného skriptu musí byť používateľ v rovnakom adresári, v ktorom sa nachádza súbor `.env`, aby sa nastavenia správne načítali.

- **Testovací režim:** Spustí skript s ukážkovými dátami definovanými v konfigurácii.

  ```
  python3 src/sectech_log_augmentator/main.py -t
  ```

- **Produkčný režim:** Umožňuje spracovanie vlastného log súboru a generovanie požadovaného počtu augmentovaných záznamov.

  ```
  python main.py --input_log_file <log_file> --number_of_logs <
      count>
  ```

- **Príklad použitia:**

  ```
  python3 src/sectech_log_augmentator/main.py --i src/
      sectech_log_augmentator/logy.json --n 1
  ```

Konfigurácia pomocou `.env`

Správanie nástroja je konfigurovateľné prostredníctvom súboru `.env`, v ktorom sa nastavujú parametre logovania, augmentačných techník, spôsobov generovania, ako aj rôzne systémové možnosti. Tento súbor umožňuje flexibilne prepínať medzi rôznymi režimami bez potreby úprav zdrojového kódu. Kompletný výpis dostupných parametrov nájdete v ukážkovom konfiguračnom súbore nižšie Výpis 5.1, kde je možné aj vidieť vysvetlivky k jednotlivým nastaveniam. V rámci súboru `.env` môže používateľ nastaviť tieto hlavné oblasti:

Konfigurácia logovania

Úroveň logovania, umiestnenie log súboru, formát správ, maximálna veľkosť logu, počet záložných súborov. Ukážka príkladu nastavenia konfigurácie logovania:

```
LOG_LEVEL = "INFO"
LOG_FILE = "logs/pm-log-augmentator.log"
LOG_TO_CONSOLE = True
```

Riadiace príznaky augmentácie

Používanie masiek, generovanie syntetických dát. Nastavením `ENABLE_SYNTHETIC_DATA` na `True` alebo `False` je možné zapínať alebo vypínať generovanie syntetických dát. Parameter `AUGMENTATION_PART` určuje, aká časť logu sa má augmentovať (hodnota medzi 0.0–1.0). Ukážka príkladu nastavenia riadiacich príznakov augmentácie:

```
ENABLE_SYNTHETIC_DATA = True
AUGMENTATION_PART = 0.5
```

Augmentácia na úrovni entít

Pre aktiváciu augmentácie musí byť `USE_AUGMENTATION_WHITELIST` nastavené na `True`. Potom je možné pomocou parametra `AUGMENTATION_WHITELIST` zvoliť konkrétne augmentačné techniky pomocou čísla z rozsahu (1-14). Ukážka príkladu nastavenia augmentácie na úrovni entít:

```
USE_AUGMENTATION_WHITELIST = True
AUGMENTATION_WHITELIST = 2  # random_swap_words
```

Augmentácia celých logov

Na zapnutie tejto možnosti je potrebné nastaviť `USE_WHOLE_AUGMENTATION_WHITELIST` na `True`. Potom sa parametrom `WHOLE_LOG_AUGMENTATION_WHITELIST` vyberie konkrétna metóda transformácie logov (napr. nahradenie medzier interpunkciou, odstránenie interpunkcie) pomocou čísla z rozsahu (1-9). Možnosťou je aj technika

SHUFFLE_ENTITIES, ktorá iba pri nastavení na True náhodne premieša entity v rámci logu. Ukážka príkladu nastavenia augmentácie celých logov:

```
USE_WHOLE_AUGMENTATION_WHITELIST = True
WHOLE_LOG_AUGMENTATION_WHITELIST = 3  # wrap_text
```

Ukážka príkladu nastavenia premiešania entít:

```
SHUFFLE_ENTITIES = True
```

Výpis 5.1: Konfiguračný súbor pre logovanie a augmentáciu dát.

```
#####################################
# LOGGING CONFIGURATION
#####################################

LOG_FILE = "logs/pm-log-augmentator.log"    # Log file location
LOG_LEVEL = "DEBUG"          # Log level (e.g., INFO, DEBUG, WARNING)
LOG_SIZE = 104857600           # Maximum log file size (in bytes)
LOG_FORMAT = "%(asctime)s %(levelname)-7s %(message)s" # Log format
BACKUP_LOG_COUNT = 3      # Number of backup logs to keep
LOG_TO_CONSOLE = True     # Whether to log to console
DEBUG_MODE = True      # Enable debug mode for detailed logs

#####################################
# GENERAL SETTINGS
#####################################
SEED = 2 # Random seed for reproducibility
#####################################
# AUGMENTATION CONTROL FLAGS
#####################################
KEEP_MASK = True
# Whether to keep or remove mask in the log.Keep mask=False =>remove mask from the log. Keep mask=True =>
     generate mask entity in the log.
USE_MASK_AUGMENTATION = False     # Enable or disable mask-based augmentation
ENABLE_SYNTHETIC_DATA = True              # Enable synthetic data generation

#####################################
# DEEP AUGMENTATION SETTINGS
#####################################
USE_FILL_MASK_METHOD = False    # Enable or disable deep augmentation
MODEL_NAME = "Roberta-base"
# Model name to create a model URI that can be used to fetch the model artifacts from MLflow
MODEL_VERSION_ALIAS = "newest"
# Alias for a specific version of the model to create a model URI that can be used to fetch the model artifacts
     from MLflow
USE_OLLAMA_API_GENERATION = False   # Enable or disable OLLAMA API generation
OLLAMA_API_URL = "http://192.168.40.2:11434" # OLLAMA API URL, need to make calendar reservation for GPU
OLLAMA_API_MODEL = "llama3:70b"  # OLLAMA API model name, has to be installed in the OLLAMA API
PRINT_ENTITIES = False     # Print entities after the augmentation process
#####################################
# ENTITY-LEVEL AUGMENTATION SETTINGS
#####################################

SAVE_ENTITY_NAMES_TO_JSON = False
USE_AUGMENTATION_WHITELIST = False
AUGMENTATION_PART = 0.3
AUGMENTATION_WHITELIST = 1
# 1  = to_upper_case
# 2  = random_swap_words
# 3  = random_case
# 4  = remove_stop_words
# 5  = to_lower_case
# 6  = word_dropout
# 7  = ocr_augmentation
# 8  = random_deletion
# 9  = random_insertion
# 10 = random_swap
# 11 = keyboard_augmentation
# 12 = character_substitution
# 13 = modify_punctuation
# 14 = blank_noising

#####################################
# WHOLE-LOG AUGMENTATION SETTINGS
#####################################

SHUFFLE_ENTITIES = True
USE_WHOLE_AUGMENTATION_WHITELIST = False
WHOLE_LOG_AUGMENTATION_WHITELIST = 1
# 1 = replace_spaces_with_underscore
# 2 = replace_spaces_with_dash
# 3 = wrap_text
# 4 = normalize_unicode
# 5 = replace_spaces_with_double_underscore
# 6 = replace_spaces_with_double_dash
# 7 = replace_spaces_with_double_punctuations
# 8 = remove_punctuation_symbols
# 9 = replace_spaces_with_single_punctuation
```

B Obsah elektronickej prílohy

- `/` — koreňový adresár projektu
  - `src`
    - `sectech_log_augmentator`
      - `file_manipulator` — modul pre manipuláciu so súbormi
        - `file_creator.py` — tvorba výstupného JSON súboru
        - `file_reader.py` — načítanie vstupného JSON súboru
      - `log_augmentator` — hlavný augmentačný modul
        - `code_functions`
          - `config.py` — konfiguračný súbor
          - `faker_functions` — Faker generátori syntetických údajov
        - `list_functions`
          - `.txt` — zoznamy pre generovanie syntetických údajov
        - `custom_augmentator.py` — augmentácia entít a celého textu v logu
        - `custom_generator.py` — namapovanie jednotlivých generátorov
        - `config.py` — nastavenia augmentačných poskytovateľov a whitelisty
      - `pm_deep_augmentation` — modul pre pokročilú augmentáciu
        - `deep_augmentator_service`
          - `AI_augmentator.py` — doplňovanie <mask> tokenov
          - `AI_Ollama_augmentator.py` — Ollama masková LLM augmentácia
          - `pipeline.py` — generovanie maskovaných hodnôt
      - `pm_log_utils` — modul pre logovanie systému
        - `src`
          - `sectech_log_utils`
            - `log_utils.py` — univerzálne logovanie systému
      - `text_manipulator` — modul pre textovú augmentáciu
        - `config.py` — konfiguračný súbor
        - `text_augmentation_selector.py` — výber a aplikácia augmentačných metód
        - `text_augmentator.py` — obsahuje jednotlivé augmentačné metódy
      - `demo_data.py.py` — testovacie dáta
      - `log_config.py` — konfigurácia prostredia a správania augmentátora
      - `main.py` — hlavný spúšťací skript pre augmentáciu logov
      - `mlflow_config.py` — získanie verzie modelu
      - `pipeline.py` — inicializácia a test augmentátora
    - `.env` — konfigurácia prostredia pre spracovanie logov a nastavovanie augmentácií
    - `.gitignore` — nastavenia pre ignorovanie súborov v systéme git
    - `.gitmodules` — súbor na správu verzií externých knižníc
    - `README.md` — základná dokumentácia k projektu
    - `pyproject.toml` — definícia závislostí a konfigurácie projektu
    - `requirements.txt` — zoznam požadovaných knižníc a verzií

C Prehľad metód augmentácie textu

Tab. 5.5: Štúdie zaoberajúce sa augmentáciou na úrovni znakov.

Augmentácia	Popis	Výsledky	Článok	Rok
Prehodenie znakov, Náhodné premiešanie vnútornej časti slova, Kompletná náhodná zmena poradia znakov, Zámenou znakov za susediace klávesy na klávesnici	Simulácia preklepov narušením slov	Vyššia odolnosť voči chybnému vstupu	Belinkov & Bisk [64]	2018
Mazanie, transpozícia, vkladanie znakov	Úprava znakov bez zmeny prvého a posledného písmena	Vyššia diverzita, plynulosť, zachovanie sentimentu	Feng et al. [65]	2020
Vkladanie znakov	Náhodné znaky vložené do textu	82,6% presnosť (BERT, PubMed20K)v porovnaní so základným modelom (79,2%)	Dai et al. [66]	2023
Prehodenie znakov	Náhodná výmena 2 znakov (simulácia preklepov)	76,2% presnosť (BERT, Amazon) porovnaní so základným modelom (73,4%)	Dai et al. [66]	2023
Vymazanie znakov	Náhodné odstránenie znakov	-	Dai et al. [66]	2023
OCR augmentácia	Simulácia chýb pri OCR rozpoznávaní	76,8% presnosť (BERT, Symptoms)v porovnaní so základným modelom (63,6%)	Dai et al. [66]	2023
Fonetické substitúcie	Zámena písmen fonetickými	Zlepšená generalizácia	Li et al. [63]	2022
Pravopisná augmentácia	Úmyselné pravopisné chyby	80,8% (BERT) v porovnaní so základným modelom (79,2%)	Dai et al. [66]	2023
Zmena veľkosti písmen	Variácia veľkých a malých písmen	+2,5% presnosť	Coulombe [68]	2018
Úprava interpunkcie	Zmeny interpunkcie pre variabilitu	+2,5% presnosť	Coulombe [68]	2018
Prírodný šum	Simulované preklepy bežnými chybami	+1,5% presnosť	Belinkov & Bisk, Coulombe [64], [68]	2018
Pravidlové transformácie	Pravopisné chyby, úpravy entít, skratky	+0,5% presnosť	Coulombe [68]	2018

Tab. 5.7: Štúdie zaoberajúce sa augmentáciou na úrovni slov.

Augmentácia	Popis	Výsledky	Článok	Rok
Unigramový šum	Nahrádzanie slov na základe pravdepodobnosti	Zlepšená klasifikačná presnosť	Xie et al. [69]	2017
Šum prázdnym znakom	Nahrádzanie slov podčiarkovníkmi	Zlepšená klasifikačná presnosť	Xie et al. [69]	2017

Augmentácia	Popis	Výsledky	Článok	Rok
Syntaktický šum	Skracovanie viet, úprava prídavných mien a štruktúr viet	+1,7 bodu presnosť	Li et al. [70]	2017
Sémantický šum	Nahrádzanie slov synonymami	+1,7 bodu presnosť	Li et al. [70]	2017
Vypadávanie slov	Náhodné maskovanie slov počas trénovania	+1,7 bodu presnosť	Li et al. [70]	2017
Náhodné vloženie	Vkladanie náhodných slov do textu	Zlepšenie výkonu pri málo dostupných dátach	Wei & Zou [71]	2019
Náhodná zámena	Náhodné prehodenie slov v rámci vety	Zlepšenie výkonu pri málo dostupných dátach	Wei & Zou [71]	2019
Náhodné vymazanie	Náhodné odstraňovanie slov z vety	Zlepšenie výkonu pri málo dostupných dátach	Wei & Zou [71]	2019
Náhrada synonymami	Nahrádzanie slov synonymami z WordNetu	+2,85% presnosť	Kapusta et al. [72]	2024
Redukcia funkčných slov	Odstránenie funkčných slov, ponechanie obsahových	+3,06% presnosť	Kapusta et al.	2024
Posun sekvencií	Posun sekvencií v rámci paddingu	5,8% zlepšenie Macro-F1	Rizos et al. [73]	2019
Zavádzanie šumu do paddingu	Pridanie bezvýznamových slov na začiatok alebo koniec	-	Sun & He [74]	2020
Náhrada podľa TF-IDF	Nahradzovanie menej významných slov podľa TF-IDF rebríčka	Zlepšená schopnosť ignorovať irelevantné slová	Xie et al. [69]	2017
Kontrafaktuálne príklady	Maskovanie kľúčových slov na generovanie kontrastných príkladov	Efektívne pri kontrastnom učení	Choi et al. [75]	2022
Synonymická náhrada pomocou tezauru	Použitie tezauru z WordNetu na rozšírenie slov	Zlepšenie presnosti	Zhang et al. [76]	2015
Náhrada z WordNet a VerbNet	Použitie synonym z WordNet aj VerbNet	Zlepšenie presnosti na základe podobnosti v embeddingu	Li et al. [63]	2022
Augmentácia pomocou WordNet	Nahrádzanie slov synonymami z WordNetu	80,5% presnosť (BERT)	Dai et al. [66]	2023
Nahrádzanie pomocou embeddingov	Nahrádzanie slov na základe podobnosti v embedding priestore	Zlepšenie F1-skóre o 2,4	Wang & Yang [78]	2015
Augmentácia counter-fitting	Úprava embeddingov na posilnenie podobnosti synonym a oslabenie antoným	Zlepšená konzistentnosť označovania	Mrkšić et al. [79]	2016
Nahrádzanie pomocou embeddingov z Google News	Nahrádzanie slov na základe embeddingov z Google News	+3,2% presnosť	Dai et al. [66]	2023
Counter-Fitted embedding augmentácia	Zlepšenie zachytávania vzťahov medzi synonymami a antonymami	75,4% presnosť na datasete Amazon	Dai et al. [66]	2023
Augmentácia pomocou c-BERT	Generovanie slov na základe kontextu pomocou BERT	Zlepšená presnosť, ale obmedzená flexibilita aplikácie	Wu et al. [81]	2018
Reinforcement Learning s c-BERT	Doladenie c-BERT pomocou posilňovacieho učenia	Výrazné zlepšenie pri malom množstve dát	Hu et al. [82]	2019

Augmentácia	Popis	Výsledky	Článok	Rok
Kontextuálna náhrada slov pomocou BERT	Kontextovo závislá náhrada slov s použitím BERT, DistilBERT a RoBERTa	Modely sú flexibilnejšie, realistickejšie generujú texty a lepšie zachovávajú kontext aj pri malom množstve dát	Dai et al. [66]	2023
Náhrada podslov	Tokenizácia slov na podslová a ich pravdepodobnostná náhrada	Efektívna augmentácia pre modely učené na maskovaní slov	Jiao et al. [83]	2019
Vkladanie slov z Google News	Náhodný výber slova zo slovníka Google News a vloženie na náhodné miesto v texte	Najlepší výkon pre model BERT	Dai et al. [66]	2023
Embeddingová substitúcia	Využíva GloVe embeddingy na výmenu slov na základe podobnosti v embeddingovom priestore	Hoci táto metóda priniesla lexikálnu variabilitu, nie vždy zachovala kontextový význam slov	Zaiton a Alansary [77]	2015

Tab. 5.8: Štúdie zaoberajúce sa augmentáciou na úrovni viet a dokumentov.

Augmentácia	Popis	Výsledky	Článok	Rok
Inverzia a pasivizácia	Preusporiadanie viet alebo použitie trpného rodu na zachovanie významu	Lepšia generalizácia modelu na datasete MNLI	Min et al. [84]	2020
MixUp augmentácia	Interpolácia medzi pármi textov na vytvorenie syntetických príkladov	Zníženie preučenia, zvýšenie presnosti klasifikácie	Guo et al. [85]	2019
Sémantická výmena textu	Cielená úprava fráz pri zachovaní sentimentu	Mierne zníženie plynulosti a konzistencie	Feng et al. [86]	2019
Orezanie a rotácia	Odstraňovanie častí viet alebo ich preusporiadanie	Významné zlepšenie v úlohách s málo dátami (napr. POS tagging)	Şahin & Steedman [87]	2018
Permutácia slov	Náhodná zmena poradia slov vo vete	Zvýšenie presnosti klasifikácie až o 4,1 %	Haralabopoulos et al. [88]	2021
Náhrada podštruktúry	Náhrada štruktúr viet medzi vzorkami tej istej triedy	Takmer dvojnásobná presnosť na datasete SST-2 a AG News	Shi et al. [89]	2021
Lexikalizované gramatické stromy	Náhrada slov podľa slovných druhov v syntaktických stromoch	Zlepšenie presnosti v few-shot a semi-supervised úlohách	Kim et al. [90]	2021
Obojsmerný preklad (round-trip)	Preklad do iného jazyka a späť na vytvorenie parafráz	Až 5,8 % zlepšenie klasifikačnej presnosti	Various studies	-
Spätný preklad	Preklad do iného jazyka a späť pri zachovaní významu	77,8 % presnosť (BERT) na datasete Symptoms v porovnaní so základným modelom (63,6%)	Dai et al. [66], Gupta & Mahmood [92]	2021, 2024
Náhodné vzorkovanie počas beam search	Vzorkovanie počas dekódovania pre rôznorodé preklady	Zvyšuje rozmanitosť prekladov	Xie et al. [93]	2020
Riadenie teploty softmaxu	Úprava náhodnosti pri dekódovaní	Lepšie zachovanie sémantickej rozmanitosti	Nugent et al. [94]	2021
Spätný preklad + adversariálne učenie	Kombinácia prekladu a adversariálnych príkladov	Vysoká kvalita a rôznorodosť dátových vzoriek	Qu et al. [95]	2020

Tab. 5.10: Štúdie zaoberajúce pokročilou augmentáciou.

Augmentácia	**Popis**	**Výsledky**	**Článok**	**Rok**
Indukcia šumu	Aditívny a multiplikatívny šum v embedding priestore	Výrazné zlepšenie v few-shot scenároch	Kumar et al. [97]	2019
Linear Delta	Pridanie rozdielu medzi dvoma vzorkami k tretej z rovnakej triedy	Väčšia rozmanitosť a generalizácia	Kumar et al. [97]	2019
PGD, FreeLB	Adverzariálne perturbácie v embedding priestore	FreeLB znižuje výpočtovú náročnosť, zvyšuje robustnosť	Combettes et al. [98]	2011
Virtuálne adverzariálne učenie (VAT)	Maximalizácia KL divergencie s perturbáciami	Zlepšenie bez potreby anotovaných dát	Miyato et al. [96]	2021
SMART	Regularizácia VAT učenia pre stabilitu	Stabilizácia a kontrola agresivity aktualizácií	Jiang et al. [100]	2021
ALUM (adversarial pretraining)	Adverzariálna optimalizácia počas pretrénovania	Zvýšená odolnosť RoBERTa voči šumu	Wang et al. [101], Liu et al. [102]	2019
Token/feature/span cutoff (vynulovanie)	Vynulovanie embeddingu slova, dimenzie alebo segmentu slov	Jednoduché, ale účinné techniky porovnateľné s PGD	Shen et al. [103]	2020
Generatívna augmentácia (GPT/BERT)	Tvorba nových dát pomocou generatívnych jazykových modelov	Výrazné zlepšenie presnosti v NLP úlohách	Anaby-Tavor et al. [104]	2019
VAEs / CVAEs	Generovanie z prior/posterior distribúcie	Vyváženie diverzity a relevancie	Qui et al. [105], Malandrakis et al. [106]	2020, 2019
NeuralEditor / Edit-transformer	Generovanie textov cez vektorové editovanie	Vhodné na modelovanie jazykov a transfer learning	Guu et al. [107], Raille et al. [108]	2018, 2020
RNN/LSTM-CNN generátory	Generovanie viet po segmentácii textu	Vyššia diverzita, lepšia generalizácia	Rizos et al. [73], Ollagnier and Williams [109]	2019, 2020
GANs (seqGAN, CS-GAN)	Reinforcement learning medzi generátorom a diskriminátorom	Zlepšenie kvality a triednej konzistencie generovaných vzoriek	Sun a He [74]	2020
GPT-2 / GPT-3 augmentácia	Tvorba celých syntetických inštancií cez prompting	Skokové zlepšenia v few-shot režimoch	Wang a Lillis [111], Anaby-Tavor et al. [104], Yoo et al. [112]	2019, 2019, 2021
Slice-based generovanie	Augmentácia málo zastúpených segmentov dát	Lepšia reprezentácia minoritných tried	Lee et al. [113]	2021
AugGPT (s ChatGPT)	Refrázovanie viet cez ChatGPT, následný fine-tuning	Zlepšenie v few-shot klasifikácii: Amazon z 73,4 % na 81,6 %, symptómy z 63,6 % na 88,9 %, PubMed20K z 79,2 % na 83,5 %	Doi et al. [66]	2023