

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

Evidenčné číslo: FIIT-16768-120897

**Dávid Truhlář**

**Výskum v oblasti technológie digitálneho  
dvojčata**

Bakalárska práca

Vedúci záverečnej práce: Ing. Matej Petrík

Máj 2025



Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

Evidenčné číslo: FIIT-16768-120897

**Dávid Truhlář**

# **Výskum v oblasti technológie digitálneho dvojčata**

Bakalárska práca

Študijný program: Informatika

Študijný odbor: Informatika

Školiace pracovisko: Ústav počítačového inžinierstva a aplikovanej informatiky

Vedúci záverečnej práce: Ing. Matej Petřík

Máj 2025





## ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Dávid Truhlár**  
ID študenta: 120897  
Študijný program: informatika  
Študijný odbor: informatika  
Vedúci práce: Ing. Matej Petrík  
Vedúci pracoviska: Ing. Katarína Jelemenská, PhD.

Názov práce: **Výskum v oblasti technológie digitálneho dvojčata**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Digitálne dvojča je inovatívna technológia, ktorá mení spôsob, akým chápeme a interagujeme s fyzickými objektami, procesmi či systémami. Digitálne dvojča je virtuálna alebo digitálna kópia fyzického objektu, systému alebo procesu, pričom sa snaží zachytiť a kopírovať čo najpresnejšie jeho vlastnosti. Technológia digitálneho dvojčata zohráva kľúčovú rolu v Priemysle 4.0 kvôli možnostiam monitorovania, simulácie a automatizácie v reálnom čase. Vďaka týmto možnostiam využitie tejto technológie umožňuje nové úrovne inovácie a optimalizácie naprieč rôznymi odvetvami a taktiež má potenciál akcelerovať vývoj v týchto oblastiach. Preskúmajte využitie technológie digitálneho dvojčata a jej aplikácie. Zamerajte sa na preskúmanie konceptu digitálneho dvojčata, na spôsob jeho implementácie a taktiež aj na dostupné technológie na jeho tvorbu. Vytvorte prehľad dostupných riešení, navrhnite spôsob vytvorenia jednoduchého digitálneho dvojčata, ktoré bude kópiou fyzického objektu alebo systému. Navrhnuté riešenie implementujte a overte jeho funkčnosť. Literatúra: 1. Singh, Maulshree, et al. "Digital twin: Origin to future." Applied System Innovation 4.2 (2021): 36. 2. Crespi, Noel, Adam T. Drobot, and Roberto Minerva. The Digital Twin. Cham: Springer International Publishing, 2023. 3. Jones, David, et al. "Characterising the Digital Twin: A systematic literature review." CIRP journal of manufacturing science and technology 29 (2020): 36-52.

Rozsah práce: 40

Termín odovzdania bakalárskej práce: 12. 05. 2025  
Dátum schválenia zadania bakalárskej práce: 15. 04. 2025  
Zadanie bakalárskej práce schválil: doc. Ing. Ján Lang, PhD. – garant študijného programu

## Čestné prehlásenie

Čestne vyhlasujem, že som túto prácu vypracoval samostatne, na základe konzultácií a s použitím uvedenej literatúry.

V Bratislave, 12.5.2025

.....

Dávid Truhlář



## Pod'akovanie

Touto cestou by som rád vyjadril úprimné poďakovanie všetkým, ktorí ma podporili a pomohli mi pri vypracovaní tejto bakalárskej práce.

Na prvom mieste patrí moja vďaka školiteľovi Ing. Matejovi Petríkovi za jeho odborné vedenie, ochotu, cenné rady a usmernenia, ktoré výrazne prispeli k realizácii tejto práce. Ďalej vyjadrujem poďakovanie Ing. Matejovi Janebovi za jeho čas a technické konzultácie v oblasti 5G technológií.

Osobitné poďakovanie patrí aj mojej rodine za prejavenu dôveru, trpezlivosť a neustálu podporu počas celého štúdia.





# Obsah

<b>1</b>	<b>Technický abstrakt</b>	<b>1</b>
<b>2</b>	<b>Laický abstrakt</b>	<b>3</b>
<b>3</b>	<b>Úvod</b>	<b>5</b>
<b>4</b>	<b>Formulácia problému a riešenie</b>	<b>7</b>
4.1	Formulácia problému . . . . .	7
4.2	Technický literárny prehľad . . . . .	8
4.3	Prehľad riešenia na vysokej úrovni . . . . .	10
4.4	Hodnotenie rizík . . . . .	13
4.5	Technická reprodukovateľnosť a integrácia . . . . .	14
4.6	Udržateľnosť a environmentálny dopad . . . . .	18
4.7	Zamestnateľnosť . . . . .	19
4.8	Tímová práca, diverzita a inklúzia . . . . .	20
<b>5</b>	<b>Záver</b>	<b>21</b>
	<b>Zoznam použitej literatúry</b>	<b>25</b>
<b>A</b>	<b>Harmonogram práce</b>	
<b>B</b>	<b>Používateľská príručka</b>	

**C Obsah digitálneho média**

**D Technická dokumentácia**



# Zoznam obrázkov

4.1	Model zrkadlených priestorov . . . . .	8
4.2	Architektúra komponentov Open5GS a UERANSIM . . . . .	11
4.3	Architektúra systému: simulácia 5G siete, zber metrických dát a real-time klasifikácia s výstupom do Grafany. . . . .	12



# Zoznam tabuliek

5.1	Výkon klasifikácie tried pre syntetické a reálne dáta. . . . .	21
-----	--	----





# Zoznam použitých skratiek

<b>5G</b>	Sieť piatej generácie
<b>AMF</b>	Funkcia riadenia prístupu a mobility (Access and Mobility Management Function)
<b>BP</b>	Bakalárska práca
<b>CSV</b>	Hodnoty oddelené čiarkou (Comma-Separated Values)
<b>DoS</b>	Distribúované odmietnutie služby
<b>DT</b>	Digitálne dvojča (Digital Twin)
<b>gNB</b>	Node B novej generácie (Next Generation Node B)
<b>GPU</b>	Grafický procesor (Graphics Processing Unit)
<b>IMSI</b>	Medzinárodná identita mobilného účastníka (International Mobile Subscriber Identity)
<b>IoT</b>	Internet vecí (Internet of Things)
<b>IT</b>	Informačné technológie
<b>LSTM</b>	Dlhá krátkodobá pamäť (Long Short-Term Memory)
<b>PCAP</b>	Záznam sieťovej komunikácie (Packet Capture)
<b>PI</b>	Permutačná dôležitosť (Perutational imporatnce)
<b>QoS</b>	Kvalita služby (Quality of Service)
<b>RAN</b>	Rádiová prístupová sieť (Radio Access Network)
<b>SUCI</b>	Maskovaný identifikátor účastníka (Subscription Concealed Identifier)
<b>UC</b>	Používateľský scenár (User case)
<b>UE</b>	Užívateľské zariadenie (User Equipment)
<b>UPF</b>	Funkcia používateľskej roviny (User Plane Function)



# 1 Technický abstrakt

Siete piatej generácie (5G) predstavujú základnú infraštruktúru pre aplikácie s prísnyimi požiadavkami na odozvu (latency) a spoľahlivosť. Technológia digitálneho dvojčata (DT) má v tomto kontexte potenciál slúžiť ako adaptívna vrstva pre simuláciu, monitorovanie a klasifikáciu sieťovej prevádzky. Táto práca sa zameriava na návrh a implementáciu DT 5G siete, ktoré v reálnom čase analyzuje metriky jadra siete a klasifikuje aktuálny typ prevádzky pomocou rekurentných neurónových sietí (LSTM).

Navrhnuté DT pozostáva z kontajnerizovaného simulačného prostredia založeného na Open5GS a UERANSIM, doplneného o klasifikačný model. V rámci experimentov boli vygenerované syntetické dáta v šiestich definovaných používateľských scenároch (UC) a získané reálne dáta z fyzickej siete. Bol navrhnutý robustný výber metrík (kombináciou metód „Random Forest“, rekurzívne odstraňovaných prízna- kov a permutačnej dôležitosti) s cieľom identifikovať znaky vhodné pre klasifikáciu v oboch doménach. Modely boli trénované výhradne na syntetických dátach, pričom ich výkon na reálnych dátach bol následne vyhodnotený s jemným doladením aj bez neho.

Napriek vysokej presnosti modelov na syntetických dátach (96,3%) sa ukázalo, že ich schopnosť generalizácie na reálnu prevádzku je výrazne obmedzená. Bez akéhokoľvek doladenia dosahovali modely na reálnych dátach presnosť len 14-44%,

pričom ani dodatočné ladenie nedokázalo zvýšiť presnosť nad 50%. To zodpovedá úrovni náhodného tipovania pri šiestich triedach a naznačuje vážny problém doménového prenosu. DT však preukázalo technickú funkčnosť, je schopné v reálnom čase zbierať dáta, spúšťať klasifikáciu a aktualizovať model pomocou jemného doladenia bez prerušenia prevádzky. Výsledky ukazujú, že samotná infraštruktúra DT je technologicky udržateľná, avšak účinné správanie klasifikačného modelu si vyžaduje pokročilejšie techniky adaptácie medzi doménami.

## 2 Laický abstrakt

Digitálne dvojča predstavuje virtuálny model reálneho systému, ktorý umožňuje sledovať a analyzovať jeho správanie v reálnom čase. V posledných rokoch si táto technológia našla uplatnenie v priemysle, doprave či zdravotníctve. V tejto práci sa venujeme vytvoreniu digitálneho dvojčaťa pre 5G sieť s cieľom porozumieť, ako sa správa pri rôznych používateľských scenároch.

Pomocou voľne dostupných nástrojov sme vytvorili softvérové prostredie, ktoré simuluje správanie reálnej mobilnej siete. Následne sme zbierali dáta nielen zo simulácií, ale aj z reálnych zariadení pripojených do 5G siete. Tieto údaje zahŕňali napríklad počet pripojených používateľov či objem prenesených dát.

Hlavným cieľom bolo overiť, či je možné natrénovať model umelej inteligencie na simulovaných dátach a použiť ho na rozpoznávanie typov sieťovej prevádzky v reálnej sieti. Takýto prístup je bezpečný, flexibilný a umožňuje testovanie aj zriedkavých alebo extrémnych scenárov bez zásahu do reálnej prevádzky.

Aj keď vytvorený systém dokáže spoľahlivo zbierať dáta, priebežne ich analyzovať a v reálnom čase aktualizovať svoj model, jeho schopnosť správne rozpoznať správanie v reálnej sieti je zatiaľ obmedzená. Táto práca tak približuje nielen výhody digitálneho dvojčaťa, ale aj výzvy, ktoré je potrebné riešiť pri nasadení takýchto systémov v reálnych 5G sieťach.



## 3 Úvod

Technológia digitálneho dvojčata (DT) predstavuje perspektívny prístup v modernom softvérovom inžinierstve [1] a telekomunikácií [2]. DT umožňuje vytvárať virtuálne kópie fyzických systémov, ktoré v reálnom čase zrkadlia ich správanie pomocou obojsmernej výmeny dát [3]. DT nachádzajú praktické uplatnenie v rôznych doménach vrátane výroby [4], zdravotníctva [5] a energetiky, kde umožňujú monitorovanie v reálnom čase, prediktívne riadenie a optimalizáciu procesov [2].

V oblasti sietí, najmä sietí piatej generácie (5G), poskytujú DT nové možnosti v simulácii sieťového správania, včasnej detekcii anomálií a optimalizácii alokácie zdrojov [6]. 5G infraštruktúry sa vyznačujú vysokou heterogenitou, dynamikou a nárokmi na kvalitu služieb (QoS), čo komplikuje ich správu a testovanie [2]. DT v tomto kontexte umožňuje testovanie rôznych scenárov vo virtuálnom prostredí bez rizika výpadku služby či narušenia integrity dát. Kľúčovou výzvou pri návrhu DT pre 5G siete je však zabezpečenie dostatočnej vernosti simulácie. Otázna zostáva najmä generalizovateľnosť modelov trénovaných iba na syntetických dátach voči reálnym podmienkam, ktoré sú komplexnejšie a menej predvídateľné. Cieľom tejto práce je preskúmať túto oblasť prostredníctvom návrhu a implementácie jednoduchého DT pre 5G sieť, postaveného na voľne dostupných nástrojoch Open5GS [7] a UERANSIM [8], a vyhodnotiť schopnosť modelov klasifikovať správanie siete v reálnom čase na základe synteticky generovaných metrík.





## 4 Formulácia problému a riešenie

### 4.1 Formulácia problému

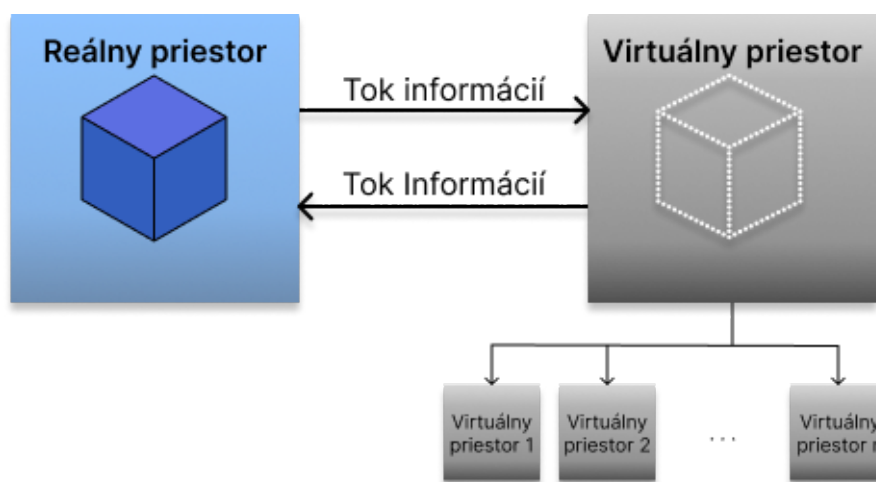
V oblasti 5G sietí predstavujú DT perspektívny nástroj na bezpečné testovanie, monitorovanie a optimalizáciu správania siete. Vysoké nároky 5G na nízku odozvu, spoľahlivosť a flexibilitu riadenia zároveň vyžadujú schopnosť rýchlo identifikovať typické aj neštandardné správanie vrátane anomálií a útokov. Aby bolo možné efektívne trénovať klasifikačné modely bez rizika pre produkčnú infraštruktúru, je nevyhnutné simulovať rôzne scenáre v kontrolovanom prostredí digitálneho dvojčaťa.

Kľúčovým problémom však zostáva otázka, do akej miery môžu modely trénované výhradne na syntetických dátach zo simulovaných prostredí, ako Open5GS a UERANSIM, generalizovať na reálne siete, keďže výrazný nesúlad medzi syntetickými a reálnymi dátami môže obmedziť praktickú využiteľnosť takto vytvorených modelov.

Cieľom tejto bakalárskej práce je preto navrhnúť a implementovať jednoduché DT 5G siete a preskúmať, do akej miery možno synteticky generované metriky využiť na tréning modelov schopných klasifikovať správanie v reálnej sieti.

## 4.2 Technický literárny prehľad

DT predstavuje koncept virtuálnej repliky fyzického objektu alebo systému, ktorá je schopná v reálnom čase odrážať jeho aktuálny stav prostredníctvom obojsmernej výmeny dát (pozri Obr. 4.1) [9]. Na rozdiel od tradičných modelov alebo simulácií je DT charakteristické dynamickým aktualizovaním svojho správania na základe dát zo sledovaného objektu [10]. Tento koncept nachádza uplatnenie naprieč viacerými odvetvami vrátane výroby, zdravotníctva, dopravy či energetiky [6].



Obr. 4.1: Model zrkadlených priestorov (Mirrored Spaces Model) tak, ako ho vo svojej práci navrhol Grieves [11]. Tento model sa skladá z troch komponentov - reálny priestor (Real Space), virtuálny priestor (Virtual space) a spájací mechanizmus (Linking Mechanism) [9], ktorý prúdi automatizovane oboma smermi medzi týmito priestormi. Virtuálny priestor vytvára digitálnu reprezentáciu reálnych objektov a podporuje viacero virtuálnych systémov na analýzu, simuláciu alebo predikciu správania fyzických objektov.

Vo svojom výskume Enders a Hořbachová [1] identifikovali sektory, kde je používanie DT najrozšírenejšie. Patria sem výroba [12], letecký priemysel [13], energetika [14], automobilový priemysel [15], námorníctvo [16], petrochemický priemysel [17],

poľnohospodárstvo [18], zdravotníctvo [19], verejný sektor [20] a ťažba [21].

Taktiež identifikovali tri hlavné využitia DT v týchto oblastiach [2]: ovládanie, simulovanie a monitorovanie. To však nepokrýva všetky možnosti a spôsoby využitia. DT dnes nájde uplatnenie aj pri dizajnovaní, validácii, predchádzaní chýb, trénoch a optimalizácii.

Vývoj DT je podporený množstvom softvérových nástrojov líšiacich sa funkcionalitou a cenou. ANSYS Twin Builder je zameraný na presné fyzikálne modelovanie procesov (mechanika, elektromagnetizmus) a je vhodný pre tvorbu inžinierskych DT, pričom patrí medzi najnákladnejšie riešenia [22]. Siemens MindSphere ponúka cloudovú platformu na správu dát a monitoring zariadení v priemyselných aplikáciách, s cenou závislou od rozsahu nasadenia [23]. MATLAB a Simulink ponúkajú flexibilnú platformu pre tvorbu DT s dôrazom na simuláciu dynamických systémov, modelovanie riadiacich algoritmov a predikciu správania [24]. Alternatívou bez licenčných nákladov je vlastná implementácia DT prostredím, kde je možné vytvoriť simulované kópie vybraných častí systému. Napríklad v oblasti 5G možno pomocou Open5GS a UERANSIM zostaviť základné virtuálne repliky siete, ktoré v kombinácii s vhodne naprogramovanými skriptami umožňujú budovanie DT.

Ak sa chceme pozrieť na reálne aplikácie DT, Huawei implementoval DT na monitorovanie výrobných liniek [25], zatiaľ čo mestá ako Bristol [26] či Singapur [27] používajú DT na efektívne riadenie inteligentných mestských systémov. V týchto scenároch DT umožňuje predikciu zlyhaní, optimalizáciu zdrojov a minimalizáciu prestojov.

V oblasti telekomunikácií umožňuje DT monitorovanie siete v reálnom čase, predikciu bezpečnostných incidentov a optimalizáciu konfigurácie bez zásahu do produkčnej infraštruktúry. Príkladom sú práce, ktoré implementovali DT do experimentálnych 5G jadier s cieľom analyzovať sieťové toky, detegovať anomálie a

klasifikovať typ prevádzky pomocou strojového učenia [28].

Významnú úlohu tu zohráva výber príznakov, ktorý umožňuje redukovať výpočtovú náročnosť klasifikácie a zároveň zvyšuje robustnosť modelov v prostredí s vysokým objemom dát. Rôzne štúdie ukazujú, že výber príznakov výrazne ovplyvňuje výkon modelov najmä v prípade detekcie útokov distribuovaného odmietnutia služby (DoS) alebo neštandardného správania zariadení internetu vecí (IoT) v 5G jadre [29]. Použité techniky zahŕňajú metódy ako Random Forest, k-najbližších susedov, rekurzívne odstraňovanie príznakov, permutačnú dôležitosť (PI) a ich kombinácie.

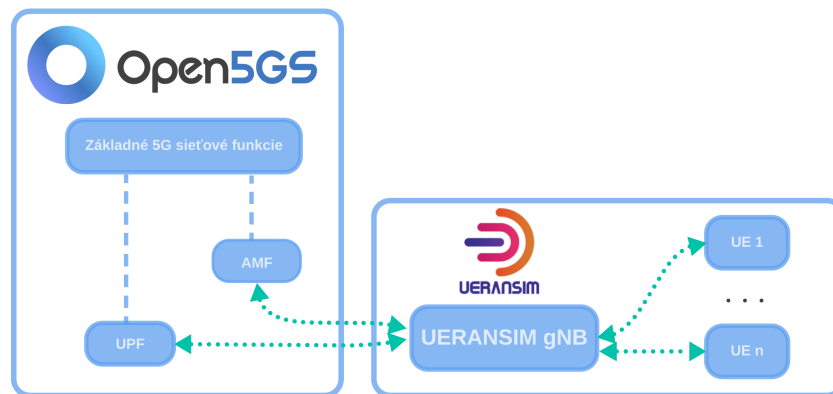
Z prehľadu literatúry vyplýva, že realistická simulácia sieťového správania a efektívna klasifikácia v reálnom čase si vyžadujú prepojenie viacerých nástrojov a metodík. Nasledujúca kapitola predstavuje konkrétny návrh a implementáciu DT systému, ktorý tieto poznatky aplikuje v kontexte 5G siete.

### 4.3 Prehľad riešenia na vysokej úrovni

Navrhnuté riešenie pozostáva z modulárneho systému na báze DT, ktorého cieľom je analyzovať a klasifikovať správanie 5G siete v kontrolovanom prostredí. Vytvorené DT simuluje vybrané komponenty reálnej siete a umožňuje generovanie syntetických dát, ich zber, spracovanie a následnú aplikáciu modelov strojového učenia. Architektúra systému bola navrhnutá s dôrazom na rozšíriteľnosť, modularitu a experimentálnu reprodukovateľnosť.

Open5GS je voľne dostupná implementácia 5G jadra siete, ktorá slúži ako základná riadiaca infraštruktúra pre prichádzajúce spojenia. V navrhnutom riešení zodpovedá za autentifikáciu zariadení, správu relácií a prideľovanie IP adries.

UERANSIM je emulátor Rádiovo-prístupovej siete (RAN) prístupu, ktorý generuje simulované používateľské zariadenia (UE), ktoré sa pripájajú do siete (pozri Obr. 4.2) cez definované UC. Umožňuje flexibilne konfigurovať počet zariadení, typy prevádzky a časovanie spojení.



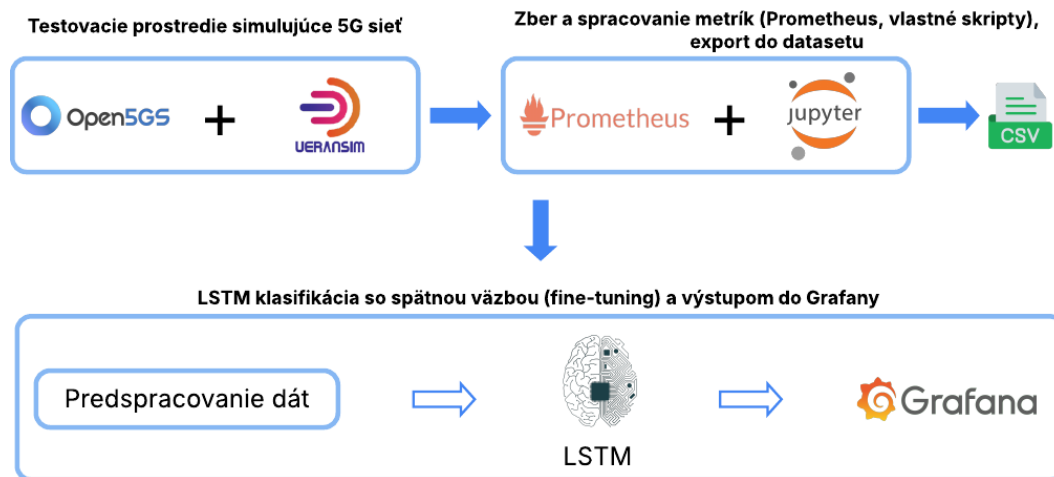
Obr. 4.2: Architektúra jednotlivých komponentov Open5GS a UERANSIM a ich vzájomné prepojenie. Open5GS je rozdelené na riadiacu rovinu a rovinu užívateľských dát a obe tieto roviny sú napojené na rádiový prístupový uzol (gNB), ku ktorým sa môžu pripájať zariadenia.

Monitorovacia infraštruktúra pozostáva z nástrojov Prometheus, Grafana, Promtail a network\_watcher. Prometheus zberá metriky zo siete a systémových zdrojov, ktoré sú vizualizované pomocou Grafany. Networky\_watcher je vlastný Python skript, ktorý analyzuje výpisy z Open5GS a exportuje vlastné metriky ako počet aktívnych UE či maskovaný identifikátor účastníka (SUCI).

Skripty na UC slúžia na spúšťanie preddefinovaných testovacích situácií. Každý UC definuje počet UE, objem prenesených dát a časovanie spojení, čím vytvára realistické syntetické zaťaženie siete.

Predspracovanie dát (preprocessing) a výber metrík (feature selection) boli vykonané raz pred trénovaním modelu, s cieľom optimalizovať vstup pre model. Výber metrík bol podporený metódami založenými na rozhodovacích stromoch a PI.

Modely strojového učenia boli implementované pomocou LSTM modelov, ktoré sú vhodné pre sekvenčné dáta. Model bol natrénovaný na syntetických dátach a neskôr testovaný na dátach z reálnej siete (pozri Obr. 4.3).



Obr. 4.3: Zariadenia generované pomocou UERANSIM sa pripájajú k jadru siete v Open5GS, ktoré loguje udalosti a vystavuje metriky. Tieto údaje sú zbierané pomocou Promethea a `network_watcher.py` skriptu. Výstupy sú agregované do datasetu, ktorý sa ďalej spracúva v prostredí Jupyter Notebook. Po spracovaní sa dáta využívajú na tréning a testovanie modelov strojového učenia. Vizualizácia metrických údajov prebieha súbežne v Grafane.

Modularita kontajnerizovaného riešenia založeného na Docker Compose umožňuje jednoduché nasadenie a replikáciu prostredia. Voľne dostupné komponenty boli zvolené kvôli dostupnosti a možnosti detailnej konfigurácie siete. Použitie syntetických scenárov poskytuje kontrolu nad testovaným správaním bez rizika ovplyvnenia reálnej prevádzky. LSTM siete boli zvolené vďaka schopnosti modelovať časové závislosti, ktoré sú kľúčové pri klasifikácii sieťového správania. Kombinácia reálneho a syntetického vstupu umožňuje overiť možnosti generalizácie modelov v telekomunikačnom prostredí.

## 4.4 Hodnotenie rizík

Implementácia DT 5G siete v kombinácii s klasifikačným modelom umelej inteligencie prináša niekoľko potenciálnych rizík, ktoré by mohli ovplyvniť presnosť, stabilitu a praktickú využiteľnosť riešenia [30].

Jedným z najvýznamnejších rizík je nedostatočná kvalita vstupných dát. Vzhľadom na to, že systém sa bude spoliehať na syntetické metriky generované v simulovanom prostredí, existuje riziko, že tieto dáta nebudú dostatočne reprezentatívne pre správanie reálnej siete. To by mohlo negatívne ovplyvniť schopnosť modelu generalizovať. Ako mitigácia sa navrhuje testovanie viacerých variantov simulácií, aplikácia metód validácie [31] (napr. krížová validácia) a neskôr prípadné doplnenie reálnych dátových vstupov [32].

Ďalším rizikom je obmedzený čas dostupný na zber a spracovanie dát. Experimentálne simulácie si vyžadujú značné množstvo iterácií a manuálnej prípravy scenárov, čo môže limitovať rozsah a kvalitu datasetu pre modelovanie [33]. Mitigáciou je dôsledné plánovanie simulácií, skriptovanie opakovaných činností a prioritizácia najrelevantnejších scenárov.

V oblasti softvérovej architektúry je potrebné počítať s možnou nekompatibilitou medzi jednotlivými komponentmi ako Open5GS, UERANSIM, Prometheus, Docker a vlastné skripty. Vzhľadom na to, že ide o voľne dostupné riešenia vyvíjané nezávisle, ich integrácia môže byť nestabilná alebo nedostatočne zdokumentovaná [34]. Odporúčanou stratégiou je modulárne nasadzovanie systémov, priebežné testovanie a dôsledná kontrola konfigurácií v menších izolovaných častiach.

Ďalším rizikom je nesynchronizácia časových pečiatok medzi komponentmi, čo môže ovplyvniť presnosť dátových okien pre LSTM model. Tento problém je možné

mitigovať precíznym logovaním časových pečiatok a synchronizáciou systémového času medzi kontajnermi.

Z pohľadu bezpečnosti a etiky sa pri použití simulovanej siete s fiktívnymi údajmi a v uzavretom prostredí eliminuje riziko úniku osobných údajov či konfiguračných súborov. Napriek tomu je vhodné minimalizovať citlivosť konfigurácií pomocou `.env` súborov [35] a testovať výhradne na syntetických identitách a medzinárodných identitách mobilného účastníka (IMSI).

Všetky uvedené riziká boli zohľadnené pri návrhu architektúry a experimentálnych scenárov tak, aby bola zabezpečená čo najvyššia spoľahlivosť riešenia.

### 4.5 Technická reprodukovateľnosť a integrácia

Riešenie DT 5G siete bolo vyvinuté s dôrazom na experimentálnu reprodukovateľnosť a jednoduché nasadenie. Celý projekt je dostupný vo verejnom repozitári na platforme GitHub<sup>1</sup>, vrátane kompletného zdrojového kódu, Docker konfigurácie a skriptov pre klasifikáciu.

K spusteniu základného systému stačí klonovanie repozitára pomocou `git clone`, pričom je potrebné, aby bol na cieľovom systéme nainštalovaný Git (2.39.5) a Docker (28.0.1). Všetky závislosti sú zabudované do kontajnerov a celý systém je orchestrovaný pomocou `docker compose`. Po nakonfigurovaní premenných prostredia v súbore `.env` a inicializácii MongoDB databázy s preddefinovanými UE, sa systém spustí jedným príkazom:

```
docker compose -f deploy-all.yaml up -build -d
```

Po nasadení komponentov je možné cez webové rozhranie Open5GS (port 9999)

---

<sup>1</sup><https://github.com/xtruhlar/5GDigitalTwin>



overiť funkčnosť jadra siete a cez Grafanu (port 3000) vizualizovať stav siete v reálnom čase. Konfigurácia UERANSIM gNB a UE je súčasťou repozitára a umožňuje simuláciu dynamického sieťového zaťaženia.

### Zber dát

Zber dát pre tréning a testovanie klasifikačných modelov bol realizovaný samostatne pre syntetickú a reálnu prevádzku. V prípade syntetických dát boli experimenty plne automatizované: systém bežal v Docker Compose prostredí. Skript každú sekundu extrahoval metriky z nástroja Prometheus, údaje z logov Open5GS a informácie o práve aktívnom UC zo sprievodného súboru. Každý riadok dát tak obsahoval časovú pečiatku, sieťové metriky, informácie o logoch a aktuálne označenie podľa spusteného UC.

Pri zbere reálnych dát bola situácia odlišná. Merania prebiehali v laboratórnom prostredí so skutočnými zariadeniami pripojenými do nezávislej 5G siete. Presný čas začiatku a konca každého používateľského scenára bol zaznamenaný manuálne. Po experimente boli všetky relevantné logy skopírované a vložené do preddefinovanej zložky, ktorú čítal ten istý skript. Na základe časového okna experimentu boli metriky zozbierané z Promethea a synchronizované s logmi. Pridávanie stĺpca s číslom UC do jednotlivých riadkov prebiehalo podľa známeho začiatku a konca testovaného UC. Tým bola zabezpečená konzistentnosť medzi stavom siete a cieľovou premennou modelu.

Výsledné datasety boli pre každý experiment exportované vo formáte CSV, pričom štruktúra bola zhodná pre syntetické aj reálne dáta.

## Trénovanie modelov a použitie výstupov

Na zabezpečenie rovnakých výsledkov tréningu boli dodržané nasledovné podmienky:

- vo všetkých notebookoch sú fixované počiatočné hodnoty (seed), veľkosť dávky (batchu), počet epôch, architektúra modelu a stratifikované rozdelenie datasetu,
- krok 1: spustenie `eda_feature_selection.ipynb`, ktorý vytvorí súbor `selected_features.json`,
- krok 2: spustenie `dataset_processing.ipynb`, ktorý vygeneruje `X_scaled.npy`, `y_labels.npy` a `scaler.joblib`,
- krok 3: spustenie `lstm_preprocessing.ipynb`, ktorý vytvára sekvencie a exportuje ich vo formáte `X_train.npy`, `y_train.npy`, `X_test.npy`, `y_test.npy`,
- krok 4: tréning modelov pomocou `lstm*_model.ipynb` (napr. `lstm_attention_model.ipynb`) na syntetických dátach,
- krok 5: vyhodnotenie na reálnych dátach v `eval_results_real_data.ipynb`,

Predtrénované modely sú uložené v

`/5GDigitalTwin/Implementation/data/Model/trained_models`.

## Klasifikácia v reálnom čase a ladenie modelu

Kľúčovým komponentom riešenia je vlastný skript `network_watcher.py`, ktorý slúži nielen na spracovanie logov z Open5GS, ale aj na priebežnú predikciu aktuálneho UC pomocou LSTM modelu a jeho ladenia na najnovších dátach. Skript beží

ako samostatný proces v jednom z Docker kontajnerov a je automaticky spúšťaný pri štarte systému.

Každú sekundu skript načíta nové riadky z logov pomocou `pygtail`, extrahuje stav jednotlivých UE zariadení (registrácia, deregistrácia, trvanie relácií) a z týchto údajov vypočítava vlastné metriky. Zároveň načíta posledných 60 záznamov z generovaného CSV súboru, normalizuje ich, a použije ako vstup do už natrénovaného LSTM modelu. Výsledná predikcia je exportovaná ako vlastná Prometheus metrika spolu s pravdepodobnosťou predikovanej triedy (confidence score) a stratou (loss) počas tréningu.

### Simulácia prípadov použitia

Na automatizovanú simuláciu UC bol vytvorený skript `running_network.py`, ktorý náhodne vyberá a spúšťa jednotlivé scenáre zo súborov `uc1.py` až `uc6.py`. Každý z týchto šiestich skriptov reprezentuje špecifický typ zaťaženia 5G siete, ako napríklad opakovanú registráciu používateľov, sťahovanie veľkého objemu dát, dlhodobé pripojenie s minimálnou aktivitou či anomálnu autentizáciu.

Skript `running_network.py` funguje ako orchestrátor – každých niekoľko minút náhodne vyberie niektorý z UC skriptov, spustí ho v pozadí ako samostatný proces a následne ho po skončení ukončí. Informáciu o aktuálne bežiacom scenári zapisuje do súboru, ktorý slúži ako jediný zdroj pravdy pre označovanie zberaných metrických dát počas trvania experimentu.

Každý `ucX.py` skript (kde  $X \in \{1, \dots, 6\}$ ) obsahuje definíciu počtu zariadení, konfiguráciu ich sieťového zaťaženia a dĺžku trvania simulácie. Scenáre boli navrhnuté tak, aby pokrývali reprezentatívne prípady bežnej aj abnormálnej prevádzky v 5G sieti. Táto architektúra umožňuje nielen spoľahlivý zber syntetických dát, ale

aj jednoduché rozšírenie systému o ďalšie prípady použitia bez zásahu do zvyšku infraštruktúry.

Celá infraštruktúra tak poskytuje replikovateľné, modulárne a škálovateľné prostredie vhodné na výskum DT v oblasti 5G, s podporou klasifikácie, monitorovania a experimentálnej validácie modelov v reálnom čase.

### 4.6 Udržateľnosť a environmentálny dopad

DT predstavujú účinný nástroj na optimalizáciu prevádzkových procesov – v porovnaní s fyzickým testovaním umožňujú výrazne znížiť spotrebu zdrojov, produkciu odpadu aj operačné riziká [36]. V kontexte návrhu DT 5G siete, implementovaného v rámci tejto práce, bola environmentálna udržateľnosť reflektovaná v niekoľkých rovinách.

Z pohľadu výpočtovej náročnosti bola celá infraštruktúra navrhnutá s cieľom minimalizovať energetickú záťaž. Celý systém bol prevádzkovaný na lokálnom výpočtovom zariadení bez využitia centralizovaných cloudových služieb či grafických akceleratorov (GPU). Kombinácia kontajnerizovaných služieb (Docker Compose) a voľne dostupných softvérových nástrojov (Open5GS, UERANSIM, Prometheus, Grafana) umožnila zabezpečiť nízku spotrebu zdrojov [37] pri zachovaní dostatočnej funkcionality pre experimentálne overenie konceptu [38].

V širšom zmysle sú DT považované za technológiu s potenciálom znižovať uhlíkovú stopu prostredníctvom presnejšej správy sietí, prediktívnej údržby a minimalizácie potreby fyzických zásahov do infraštruktúry [39, 40]. Aj keď prediktívne modelovanie nebolo predmetom tejto práce, implementovaný systém poskytuje základ, na ktorom je možné budovať systémy podporujúce rozhodovanie s pozitívnym environmentálnym dopadom [41].

Navyše, architektúra riešenia je modularizovaná, čo umožňuje selektívnu údržbu a výmenu komponentov bez nutnosti reinštalácie celého systému. Tento návrhový prístup je zároveň v súlade s princípmi zeleného softvérového inžinierstva, ktoré zdôrazňujú energetickú efektivitu a podporujú dlhodobú udržateľnosť softvérových riešení [42].

Navrhnutý systém teda nielen zohľadňuje technologickú efektivitu a experimentálnu hodnotu, ale zároveň reflektuje environmentálne požiadavky pri návrhu a testovaní moderných komunikačných systémov.

### 4.7 Zamestnateľnosť

Táto bakalárska práca prepája rozvoj teoretických poznatkov v oblasti DT s praktickou implementáciou a analýzou správania 5G sietí, vedie k rozšíreniu zručností vo viacerých kľúčových oblastiach technologického sektora. Dôležitou súčasťou riešenia je aj aplikácia strojového učenia na klasifikáciu správania siete, čo rozširuje záber práce do oblasti dátovej vedy a analytiky.

Formát práce umožňuje autorom prejsť celým životným cyklom softvérového projektu – od technologického prieskumu cez návrh riešenia až po jeho implementáciu a testovanie. Týmto získajú ucelený a komplexný pohľad na vývoj a riadenie softvérových projektov, ako aj na plánovanie, organizáciu a efektívnu komunikáciu.

Takáto kombinácia technických, projektových a komunikačných schopností môže významne zvýšiť hodnotu autorov na trhu práce, keďže problematika DT a 5G sietí je stále viac žiadaná a nachádza uplatnenie v rôznych odvetviach.

## 4.8 Tímová práca, diverzita a inklúzia

Hoci bola bakalárska práca vypracovaná individuálne, viaceré jej časti by nebolo možné realizovať bez odbornej spolupráce s ďalšími osobami a komunitami.

Počas celého vývoja zohrával významnú úlohu školiteľ Ing. Matej Petrík, ktorý poskytoval metodické usmernenia v oblasti DT, návrhu experimentov, spracovania dát a validácie výsledkov. V technických aspektoch súvisiacich s 5G sieťami, konfiguráciou Open5GS a zberom metrických údajov zo siete prebiehala konzultácia s Ing. Matejom Janebom, ktorý sa aktívne podieľal na verifikácii experimentálnej infraštruktúry a asistoval pri získavaní reálneho datasetu v laboratórnych podmienkach.

Dôležitým prvkom riešenia bola aj interakcia s komunitou vývojárov voľne dostupného nástroja UERANSIM. V rámci otvorenej komunikácie bol kontaktovaný hlavný autor projektu Ali Güngör so žiadosťou o objasnenie spracovania RRC Release signálov, ktoré absentovali v oficiálnom repozitári. Spolupráca napomohla lepšiemu pochopeniu obmedzení dostupných nástrojov a zdôraznila význam komunitne vyvíjaného softvéru pri budovaní výskumných riešení.

Riešenie zároveň reflektuje princípy inklúzie tým, že všetky vytvorené skripty a výstupy sú otvorene dokumentované, konfigurovateľné a dostupné bez obmedzenia technologickej platformy či licencie. Vďaka modulárnej architektúre, využitiu voľne dostupných technológií a kontajnerizovanému nasadeniu je výstup práce plne replikovateľný a potenciálne využiteľný aj v ďalších výskumných kontextoch.

## 5 Záver

V rámci tejto bakalárskej práce bol navrhnutý a implementovaný základný model DT pre 5G sieť, ktorý umožňuje simuláciu správania siete a zber dát z fyzickej i simulovanej infraštruktúry. Použitím nástrojov Open5GS a UERANSIM bol vytvorený kontrolovaný experimentálny priestor, v ktorom bolo možné testovať rôzne UC bez zásahu do reálnej siete.

Výsledky ukazujú, že klasifikačné modely trénované výhradne na syntetických dátach dosahovali výrazne horšiu presnosť pri aplikácii na reálne dáta (44%), v porovnaní so 96.3% presnosťou na dátach zo simulácie (pozri Tab. 5.1). Tento pokles jasne signalizuje rozdiel medzi reálnym a simulovaným správaním a zdôrazňuje potrebu spoľahlivých metód prenosu znalostí medzi týmito dvoma doménami.

Tabuľka 5.1: Výkon klasifikácie tried pre syntetické a reálne dáta.

Model	Presnosť	Presnosť tried	F1-Skóre	Úplnosť
<b>Syntetický dataset</b>				
<i>Základný Model</i>	0.949	0.950	0.949	0.949
<i>Robustný Model</i>	0.900	0.905	0.900	0.900
<i>Model s Normalizáciou Dávky</i>	0.963	0.964	0.963	0.963
<i>Model s Pozornosťou</i>	0.914	0.918	0.914	0.914
<b>Reálny dataset</b>				
<i>Základný Model</i>	0.14	0.06	0.04	0.14
<i>Robustný Model</i>	0.44	0.25	0.29	0.44
<i>Model s Normalizáciou Dávky</i>	0.21	0.16	0.17	0.21
<i>Model s Pozornosťou</i>	0.16	0.03	0.05	0.16

Tieto zistenia poukazujú na viaceré limity implementovaného prístupu, ktoré je potrebné otvorene pomenovať. Aj napriek systematickej exploratívnej analýze a dôslednému výberu metrík so stabilným významom naprieč oboma doménami sa ukázalo, že syntetické metriky generované pomocou Open5GS majú obmedzenú schopnosť zachytiť komplexitu reálneho sieťového správania. Chýba im detailná reprezentácia QoS mechanizmov, realistický prenos na aplikačnej vrstve, ako aj podpora mobility, čo výrazne znižuje použiteľnosť týchto simulácií ako základ pre tréňovanie modelov s dobrou generalizačnou schopnosťou. Hoci boli testované viaceré architektúry klasifikačných modelov (vrátane mechanizmu s pozornosťou a normalizácie dávky), výkonnosť na reálnych dátach ostávala nízka, čo naznačuje, že problém spočíva predovšetkým v obmedzenej kvalite dát.

Navyše, použité metriky sa týkali výlučne jadrovej vrstvy, pričom chýbali detailnejšie údaje z RAN vrstvy, ako napríklad parametre rádiového signálu alebo využitie fyzických kanálov. Z pohľadu reálneho experimentálneho zberu bola práca obmedzená na malý počet UE, krátke trvanie meraní, čo mohlo ovplyvniť spoľahlivosť výstupov. Samotné UC boli však navrhnuté tak, aby pokrývali reprezentatívne spektrum bežnej aj hraničnej prevádzky siete: od opakovaných krátkych spojení (napr. periodické získavanie predpovede počasia), cez stabilné dátové toky (streamovanie), až po málo aktívne, no perzistentné relácie (držanie spojenia) či autentizačné chyby. Každý UC tak reprezentuje špecifickú sieťovú záťaž a jeho úspešná klasifikácia poskytuje informáciu o charaktere prevádzky v danom časovom okne.

Práca ukazuje, že klasifikácia v reálnom čase a ladenie modelov je technicky realizovateľné aj v rámci jednoduchého DT, hoci zatiaľ len pri použití syntetických dát. Výsledky tejto práce zároveň naznačujú, že DT môžu nájsť uplatnenie v oblasti riadenia a optimalizácie 5G sietí – predovšetkým ako nástroj na bezpečné



testovanie sieťových zmien, včasné odhaľovanie anomálií a tréovanie systémov umelej inteligencie mimo produkčného prostredia. Do budúcnosti by mal byť výskum orientovaný na zvýšenie vernosti simulácií a zlepšenie kvality dát použitých na tréovanie modelov. Za účelom premostenia medzi syntetickými a reálnymi dátami sa ako perspektívna javí kombinácia metrických údajov z Open5GS s dátami zachytenými v podobe paketov (PCAP), čo umožní extrahovať dotatočné charakteristiky sieťového správania a vrstviť informácie naprieč L2–L7. Zároveň je potrebné preskúmať možnosti aplikácie pokročilých doménovo adaptačných techník, ktoré by mohli znížiť závislosť na rozsiahlych reálnych datasetoch.

Výzvou do budúcnosti ostáva rozšíriť prístup v reálnom čase aj na reálne sieťové prostredie a uzavrieť spätnú regulačnú slučku – teda umožniť systému nielen detegovať anomálie v reálnom čase, ale aj autonómne reagovať vhodnou rekonfiguráciou siete alebo upozornením správcu. Takýto obojsmerný tok dát medzi fyzickým a digitálnym prostredím predstavuje ďalší krok smerom k autonómnym, adaptívnym 5G sieťam.



# Zoznam použitej literatúry

- [1] Martin Enders a Nadja Hoßbach. “Dimensions of Digital Twin Applications - A Literature Review”. In: aug. 2019.
- [2] Maulshree Singh et al. “Applications of Digital Twin across Industries: A Review”. In: *Applied Sciences* 12 (jún 2022), s. 5727. DOI: 10.3390/app12115727.
- [3] Alberto Mozo et al. “B5GEMINI: Digital Twin Network for 5G and Beyond”. In: *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. 2022, s. 1–6. DOI: 10.1109/NOMS54207.2022.9789810.
- [4] Feng Xiang et al. “Digital Twin Driven Green Material Optimal-Selection towards Sustainable Manufacturing”. In: *Procedia CIRP* 81 (2019). 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019, s. 1290–1294. ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2019.04.015>. URL: <https://www.sciencedirect.com/science/article/pii/S2212827119306298>.
- [5] *From Digital Twin to Improved Patient Experience — siemens-healthineers.com*. <https://www.siemens-healthineers.com/sk/news/mso-digital-twin-mater.html>. [Accessed 29-12-2024].
- [6] Huan X. Nguyen et al. “Digital Twin for 5G and Beyond”. In: *IEEE Communications Magazine* 59.2 (2021), s. 10–15. DOI: 10.1109/MCOM.001.2000343.
- [7] Open5GS. <https://github.com/open5gs/open5gs>. 2017.

- [8] Ali Güngör. *UERANSIM*. <https://github.com/aligungr/UERANSIM>. 2020.
- [9] Maulshree Singh et al. “Digital Twin: Origin to Future”. In: *Applied System Innovation* 4 (máj 2021), s. 36. DOI: 10.3390/asi4020036.
- [10] Roberto Verdecchia et al. “Network Digital Twins: A Systematic Review”. In: *IEEE Access* 12 (2024), s. 145400–145416. DOI: 10.1109/ACCESS.2024.3453034.
- [11] Michael Grieves. “Origins of the Digital Twin Concept”. Aug. 2016. DOI: 10.13140/RG.2.2.26367.61609.
- [12] Guodong Shao a Moneer Helu. “Framework for a digital twin in manufacturing: Scope and requirements”. In: *Manufacturing Letters* 24 (2020), s. 105–107. ISSN: 2213-8463. DOI: <https://doi.org/10.1016/j.mfglet.2020.04.004>. URL: <https://www.sciencedirect.com/science/article/pii/S2213846319301312>.
- [13] Eric Tuegel et al. “Reengineering Aircraft Structural Life Prediction Using a Digital Twin”. In: *International Journal of Aerospace Engineering* 2011 (jan. 2011). DOI: 10.1155/2011/154798.
- [14] Anton Rassõlkin et al. “Implementation of Digital Twins for electrical energy conversion systems in selected case studies”. In: *Proceedings of the Estonian Academy of Sciences* 70 (jan. 2021), s. 19–39. DOI: 10.3176/proc.2021.1.03.
- [15] P.K. Rajesh et al. “Digital Twin of an Automotive Brake Pad for Predictive Maintenance”. In: *Procedia Computer Science* 165 (2019). 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019, s. 18–24. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2020.01.061>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050920300697>.

- [16] DNV GL Maritime. *Digital Twin Report for DMA: Digital Twins for Blue Denmark*. Tech. spr. [Accessed 29-12-2024]. Danish Maritime Authority, 2018.
- [17] Sophie Menard. *3 ways digital twins are going to help improve oil and gas maintenance and operations - linkedin.com*. <https://www.linkedin.com/pulse/3-ways-digital-twins-going-help-improve-oil-gas-sophie-menard/>. [Accessed 29-12-2024]. 2017.
- [18] José Monteiro et al. “Towards Sustainable Digital Twins for Vertical Farming”. In: *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*. 2018, s. 234–239. DOI: 10.1109/ICDIM.2018.8847169.
- [19] Carlos Miskinis. *Disrupting The Healthcare Industry Using Digital Twin Technology — challenge.org*. <https://www.challenge.org/insights/digital-twin-in-healthcare/>. [Accessed 29-12-2024]. 2018.
- [20] Joe David, Andrei Lobov a Minna Lanz. “Learning Experiences Involving Digital Twins”. In: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. Okt. 2018, s. 3681–3686. DOI: 10.1109/IECON.2018.8591460.
- [21] Carlos Miskinis. *The Impact of Digital Twins Technology in Mining and Mineral Extraction — challenge.org*. <https://www.challenge.org/insights/digital-twin-in-mining/>. [Accessed 29-12-2024]. 2018.
- [22] ANSYS Inc. *ANSYS Twin Builder: Simulation-driven Digital Twins*. Accessed: 2024-04-29. 2023. URL: <https://www.ansys.com/products/digital-twin/ansys-twin-builder>.
- [23] Siemens AG. *MindSphere: Industrial IoT as a service solution*. Accessed: 2024-04-29. 2023. URL: <https://siemens.mindsphere.io/>.
- [24] MathWorks. *What Is a Digital Twin?* Accessed: 2024-04-29. 2023. URL: <https://www.mathworks.com/discovery/digital-twin.html>.

- [25] Huawei. *Site Digital Twins Based 5G Digital Engineering Solution*. <https://www.huawei.com/en/news/2020/2/site-digital-twins-based-5g-digital-engineering-solution>. Accessed: 2024-12-03. 2020.
- [26] Paul Wilson. “State of smart cities in UK and beyond”. In: *IET Smart Cities* 1.1 (2019), s. 19–22. DOI: <https://doi.org/10.1049/iet-smc.2019.0024>. eprint: <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-smc.2019.0024>. URL: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-smc.2019.0024>.
- [27] Deuk-Young Jeong et al. “Digital Twin: Technology Evolution Stages and Implementation Layers With Technology Elements”. In: *IEEE Access* 10 (2022), s. 52609–52620. DOI: 10.1109/ACCESS.2022.3174220.
- [28] Mario Sanz Rodrigo et al. “Digital Twins for 5G Networks: A Modeling and Deployment Methodology”. In: *IEEE Access* 11 (2023), s. 38112–38126.
- [29] Ye-Eun Kim, Yea-Sul Kim a Hwankuk Kim. “Effective Feature Selection Methods to Detect IoT DDoS Attack in 5G Core Network”. In: *Sensors* 22 (máj 2022), s. 3819. DOI: 10.3390/s22103819.
- [30] Nour Alqudah a Qussai Yaseen. “Machine Learning for Traffic Analysis: A Review”. In: *Procedia Computer Science* 170 (2020). The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops, s. 911–916. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2020.03.111>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050920305494>.
- [31] Giang Nguyen Thu. *Introduction to data science*. Vydavateľstvo SPEKTRUM STU, 2022. ISBN: 978-80-227-5193-3. URL: [https://kis.cvt.stuba.sk/ar1-stu/sk/detail-stu\\_us\\_cat-0095495-Introduction-to-data-science/](https://kis.cvt.stuba.sk/ar1-stu/sk/detail-stu_us_cat-0095495-Introduction-to-data-science/).

- [32] Chengliang Chai et al. “Mitigating Data Scarcity in Supervised Machine Learning Through Reinforcement Learning Guided Data Generation”. In: *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 2024, s. 3613–3626. DOI: 10.1109/ICDE60146.2024.00278.
- [33] Timothy D. West a Mark Blackburn. “Is Digital Thread/Digital Twin Affordable? A Systemic Assessment of the Cost of DoD’s Latest Manhattan Project”. In: *Procedia Computer Science* 114 (2017). Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA, s. 47–56. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2017.09.003>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050917317970>.
- [34] Sumit Singh et al. “Challenges of Digital Twin in High Value Manufacturing”. In: *SAE International - Aerospace Systems and Technology Conference 2018, London*. Okt. 2018. DOI: 10.4271/2018-01-1928.
- [35] Marc-Roger Gagné. *Digital Twins, Another Reason to Worry About the IoT and Data Security*. <https://irishtechnews.ie/digital-twins-iot-and-data-security/>. 2022.
- [36] Rui Carvalho a Alberto Rodrigues da Silva. “Sustainability Requirements of Digital Twin-Based Systems: A Meta Systematic Literature Review”. In: *Applied Sciences* 11.12 (2021). ISSN: 2076-3417. DOI: 10.3390/app11125519. URL: <https://www.mdpi.com/2076-3417/11/12/5519>.
- [37] Christopher Tozzi. *Why Docker containers are good for the environment*. <https://cloudnativenow.com/features/docker-containers-good-environment/>. 2017.
- [38] Edrian Blasquino. *Docker forensics for sustainability: Unraveling the environmental impact of containerized systems*. <https://eforensicsmag.com/>

- docker-forensics-for-sustainability-unraveling-the-environmental-impact-of-containerized-systems/. 2023.
- [39] Yunlong Lu et al. “Communication-Efficient Federated Learning for Digital Twin Edge Networks in Industrial IoT”. In: *IEEE Transactions on Industrial Informatics* 17.8 (2021), s. 5709–5718. DOI: 10.1109/TII.2020.3010798.
- [40] Diego M. Botín-Sanabria et al. “Digital Twin Technology Challenges and Applications: A Comprehensive Review”. In: *Remote Sensing* 14.6 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14061335. URL: <https://www.mdpi.com/2072-4292/14/6/1335>.
- [41] Umar Danjuma Maiwada, Kamaluddeen Usman Danyaro a Aliza Bt Sarlan. “Enhancing 5G energy efficiency through digital twin networks: A comprehensive review”. In: *Digital Twins and Applications* 1.1 (2024), s. 4–25. DOI: <https://doi.org/10.1049/dgt2.12008>. eprint: <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/dgt2.12008>. URL: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/dgt2.12008>.
- [42] James Durand, Cassandra Telenko a Carolyn Seepersad. “How Does Modularity Affect Green Design?” In: *Proceedings of the ASME 2010 International Design Engineering Technical Conferences Computers and Information in Engineering Conference IDETC/CIE 2010*. Zv. 6. Aug. 2010. DOI: 10.1115/DETC2010-28760.



# A Harmonogram práce

## A.1 Zimný semester

1. - 5. týždeň	Konzultácie, rešerš problematiky
6. týždeň	Formulácia problému
7. týždeň	Technický literárny prehľad
8. týždeň	Konzultovanie, Zamestnateľnosť
9. - 10. týždeň	Implementovanie pripomenutých zmien, Hodnotenie rizík, Udržateľnosť a environmentálny dopad
11. týždeň	Návrh riešenia na vysokej úrovni
12. týždeň	Úvod, Konzultovanie, Odovzdávanie BP 1

### A.1.1 Vyjadrenie k harmonogramu

Harmonogram sa dodržal, čo prispelo k systematickému postupu pri spracovaní BP. Pravidelné konzultácie s vedúcim práce zohrali kľúčovú úlohu pri jej realizácii. Vedúci poskytoval pripomienky a odporúčania, na základe ktorých sa jednotlivé časti práce mohli upraviť do súčasnej podoby. Tento proces mi umožnil efektívne riešiť prípadné nedostatky a zabezpečiť súčasnú kvalitu výsledného dokumentu.

## A.2 Letný semester

Týždeň	Bakalárska práca	Článok
1. - 2. týždeň	Tímová práca, diverzita a inklúzia	Súvisiaca práca
3. - 4. týždeň	Open5GS, UERANSIM, Zber dát	Úvod, Metodológia, Kľúčové slová
5. - 7. týždeň	Zber dát, Štúdium LSTM	Opis Modelu
8. - 10. týždeň	ML model, tréning, Reproduk- ovateľnosť a integrácia	Evaluácia a Diskusia
11. týždeň	Technický abstrakt, laický abs- trakt	Záver, Abstrakt
12. týždeň	Zhrnutie, úpravy, konzultovanie	Budúci výskum

### A.2.1 Vyjadrenie k harmonogramu

Harmonogram sa vo všeobecnosti podarilo dodržať, hoci práca na niektorých úlohách bola náročnejšia, než sa pôvodne očakávalo. Najmä zber dát a implementácia riešenia si vyžadovali viac času a experimentovania. Vďaka systematickému postupu, pravidelnej práci a priebežným konzultáciám sa však podarilo udržať plánovaný smer a postupne napredovať. Tento prístup umožnil priebežné identifikovanie problémov, ich efektívne riešenie a priebežné zlepšovanie kvality výstupov, čo sa pozitívne prejavilo aj na výslednom spracovaní bakalárskej práce a sprievodného článku.

# B Používateľská príručka

## Používateľská príručka

### Platformy a kompatibilita

- macOS Sequoia 15.4
- Ubuntu 22.04.3 LTS
- Windows 11 Home s nainštalovaným WSL2

### Požiadavky

Git (verzia 2.39.5), Docker (verzia 28.0.1)

### Inštalácia a spustenie systému

1. Klonujte repozitár a prejdite do adresára projektu:

```
git clone https://github.com/xtruhlar/5GDigitalTwin.git  
cd 5GDigitalTwin/Implementation
```

2. Vytvorte Docker obrazy:

```
cd ./base
docker build -t docker_open5gs .

cd ../ueransim
docker build -t docker_ueransim .

cd ..
```

3. Nastavte premenné prostredia:

```
cp .env.example .env

set -a
source .env
set +a
```

4. Spustite celý systém pomocou Docker Compose:

```
docker compose -f deploy-all.yaml up --build -d
```

5. Naimportujte preddefinovaných účastníkov do MongoDB:

```
docker exec -it mongo mkdir -p /data/backup
docker cp ./mongodb_backup/open5gs mongo:/data/backup/open5gs
docker exec -it mongo mongorestore \
  --uri="mongodb://localhost:27017" \
  --db open5gs /data/backup/open5gs
```

6. Overté funkčnosť jadra siete cez Open5GS WebUI:

Otvorte `http://localhost:9999` Prihlásenie:

Meno: `admin`

Heslo: `1423`

7. Spustite UERANSIM gNB:

```
docker compose -f nr-gnb.yaml -p gnodeb up -d && \
docker container attach nr_gnb
```

8. Pripojte registrované zariadenie (UE):

```
docker compose -f nr-UEs/nr-ue1.yaml -p ues up --build -d
```

9. Vizualizácia klasifikovaného stavu 5G siete: Otvorte Grafanu na `http://localhost:3000` Prihlásenie:

Meno: `open5gs`

Heslo: `open5gs`

Potom kliknite na `Dashboards` → `Current state Dash`



# C Obsah digitálneho média

Evidenčné číslo práce v informačnom systéme: FIIT-16768-120897

Obsah digitálnej časti práce (archív ZIP):

Zložka	Obsah
/data	Dataseť, Jupyter notebooky, a LSTM
→ modely	
/backup	Pravidelné ukladanie dát
/datasets	Dataseť syntetických a reálnych dát
/Jupyter Notebooks Output	HTML súbory
/logs_real_5G	Zložka s log súborom z reálnej 5G siete
/Model	Modely, konfigurácie, škálovače a
→ predspracované dáta	
/json	Konfiguračné súbory pre modely
/preprocessed_data	
/scaler	
/trained_models	Modely uložené po tréningu
current_uc.txt	Aktuálny UC podľa bežiacieho skriptu
log_execution.log	
main.ipynb	Notebook, ktorý zberá metriky, spája
→ ich s logmi a UC	

running_data.csv	Data za poslednú minútu
/log	Logy zo sieťových funkcií
amf.log	
ausf.log	
bsf.log	
nrf.log	
nssf.log	
pcf.log	
scp.log	
smf.log	
udm.log	
udr.log	
upf.log	
/loki	Spojenie logov a Grafany pomocou Loki
loki-config.yaml	
/nr-UEs	Konfiguračné súbory pre UE
/open5gs	Open5GS
/amf	Konfigurácia AMF
amf_init.sh	
amf.yaml	
/ausf	Konfigurácia AUSF
ausf_init.sh	
ausf.yaml	
/base	Dockerfile pre Open5GS, WebUI a MongoDB
Dockerfile	
open5gs_init.sh	
/bsf	Konfigurácia BSF



```
bsf_init.sh
bsf.yaml
/grafana          Konfigurácia Grafany, Dashboard, Zdroje
  /dashboards     Dashboard
    current_state.json
    current_state.yaml
  /datasources
    loki.yaml
    prometheus_open5gs.yaml
/metrics          Konfigurácia Prometheusa, Dockerfile
  Dockerfile
  metrics_init.sh
  prometheus.yaml
/mongodb_backup   Uložené DB s registrovanými UE
  /open5gs        DB
    accounts.bson
    accounts.metadata.json
    sessions.bson
    sessions.metadata.json
    subscribers.bson
    subscribers.metadata.json
/nrf              Konfigurácia NRF
  nrf_init.sh
  nrf.yaml
/nssf             Konfigurácia NSSF
  nssf_init.sh
  nssf.yaml
```

/pcf	Konfigurácia PCF
pcf_init.sh	
pcf.yaml	
/scp	Konfigurácia SCP
scp_init.sh	
scp.yaml	
/smf	Konfigurácia SMF
ip_utils.py	
make_certs.sh	
smf_4g.yaml	
smf_init.sh	
smf.conf	
smf.yaml	
/udm	Konfigurácia UDM
curve25519-1.key	
secp256r1-2.key	
udm_init.sh	
udm.yaml	
/udr	Konfigurácia UDR
udr_init.sh	
udr.yaml	
/upf	Konfigurácia UPF
ip_utils.py	
tun_if.py	
upf_init.sh	
upf.yaml	
/webui	Konfigurácia WebUI

webui_init.sh	
/promtail	Konfigurácia Promtail
promtail-config.yaml	
/ueransim	Konfigurácia UERANSIM, UE súbory,
↪ Dockerfile	
/init_scripts	UE skripty
/yaml_configs	UE konfigurácie
Dockerfile	
ueransim_image_init.sh	
ueransim_gnb_init.sh	
ueransim_gnb.yaml	
.env.example	Premenné prostredia
deploy-all.yaml	Hlavný yaml pre Docker compose
Dockerfile.networkwatcher	Dockerfile pre skript sledujúci
↪ aktuálny stav siete	
network_watcher.py	Skript - Aktuálny stav, klasifikácia,
↪ fine-tuning	
nr-gnb.yaml	Konfiguračný súbor pre gNodeB
running_network.py	Skript, ktorý náhodne spúšťa UC{1 - 6}
uc1.py	
uc2.py	
uc3.py	
uc4.py	
uc5.py	
uc6.py	

Názov odovzdaného archívu: BP\_DavidTruhlar.zip.

## D Technická dokumentácia

Táto kapitola obsahuje podrobnú technickú dokumentáciu projektu Digital Twin of 5G Network. Obsah dokumentu bol vytvorený automaticky pomocou nástroja **Sphinx** a slúži ako referenčný manuál pre používateľov a vývojárov. Dokumentácia je vyhotovená v anglickom jazyku, keďže projekt využíva množstvo externých knižníc, nástrojov a terminológie, ktorá je bežne dostupná v angličtine. Zároveň to uľahčuje prípadnú integráciu projektu do medzinárodného výskumu či tímovej spolupráce. Okrem samotnej dokumentácie dokoženej v tejto prílohe, sa technická dokumentácia nachádza aj na webstránke<sup>1</sup> repozitára tohoto projektu.

---

<sup>1</sup><https://xtruhlar.github.io/5GDigitalTwin/>

## GETTING STARTED

<b>1</b>	<b>Getting Started</b>	<b>1</b>
1.1	Overview . . . . .	1
1.2	Prerequisites . . . . .	1
<b>2</b>	<b>Installation Guide</b>	<b>2</b>
<b>3</b>	<b>Normal Surfing</b>	<b>4</b>
<b>4</b>	<b>Video Streaming</b>	<b>5</b>
<b>5</b>	<b>Periodic Keep-Alive</b>	<b>6</b>
<b>6</b>	<b>Short Burst Sessions</b>	<b>7</b>
<b>7</b>	<b>Load Registration Anomaly</b>	<b>8</b>
<b>8</b>	<b>Authentication Failure Alert</b>	<b>9</b>
<b>9</b>	<b>LSTM Model with Attention</b>	<b>10</b>
<b>10</b>	<b>LSTM BathNorm Model</b>	<b>12</b>
<b>11</b>	<b>LSTM Base Model</b>	<b>13</b>
<b>12</b>	<b>LSTM Robust Model</b>	<b>14</b>
<b>13</b>	<b>Classify the real data using LSTM</b>	<b>15</b>
<b>14</b>	<b>Exploratory Data Analysis</b>	<b>17</b>
<b>15</b>	<b>Label Real Dataset</b>	<b>20</b>
<b>16</b>	<b>Preprocess data for LSTM</b>	<b>21</b>
<b>17</b>	<b>Simulate a running network</b>	<b>22</b>
<b>18</b>	<b>Main orchestrator</b>	<b>23</b>

## GETTING STARTED

Welcome to the documentation for the **Digital Twin of a 5G Network** developed as part of a bachelor thesis at the Slovak University of Technology. This section will guide you through setting up, running, and extending the digital twin environment.

### 1.1 Overview

This project simulates and analyzes 5G network behavior by integrating the following components:

- **Open5GS** – a full 5G core implementation.
- **UERANSIM** – emulation of multiple UE and gNBs.
- **Prometheus & Grafana** – monitoring and visualization.
- **LSTM models** – classification of network behavior using deep learning.
- **Sphinx documentation** – modular API docs.

### 1.2 Prerequisites

Before running the project, ensure you have:

- Docker & Docker Compose
- Python 3.9+ and *venv*
- Git
- At least 16 GB RAM (recommended for full stack simulation)

---

CHAPTER  
TWO

---

## INSTALLATION GUIDE

First, ensure you have the following prerequisites installed:

```
Docker and Docker Compose  
Python 3.9+ and `venv`  
Git
```

Second, clone the repository:

```
git clone https://github.com/xtruhlar/5GDigitalTwin.git  
cd 5GDigitalTwin/Implementation
```

To build docker images

```
cd ./open5gs/base  
docker build -t docker_open5gs .  
  
cd ../ueransim  
docker build -t docker_ueransim .  
  
cd ..
```

To set the environment variables

```
cp .env.example .env  
  
set -a  
source .env  
set +a
```

To run the project, navigate to *Implementation/* and execute the following command

```
docker compose -f deploy-all.yaml up --build -d
```

To add subscribers to Open5GS core, run the following commands

```
docker exec -it mongo mkdir -p /data/backup  
docker cp ./open5gs/mongodb_backup/open5gs mongo:/data/backup/open5gs  
docker exec -it mongo mongorestore --uri="mongodb://localhost:27017" --db open5gs /data/  
→ backup/open5gs
```

To ensure everything works properly, open <http://localhost:9999/> in your browser and login using credentials:

## Digital Twin of 5G Network, Technical documentation

Username: admin  
Password: 1423

To connect UERANSIM gNB to Open5GS, run the following command

```
docker compose -f nr-gnb.yaml -p gnodeb up -d && docker container attach nr_gnb
```

To connect UERANSIM UE to Open5GS, run the following command

```
docker compose -f nr-ue.yaml -p ue up -d && docker container attach ue
```

Then go to Grafana, open <http://localhost:3000/> in your browser and login using credentials:

Username: open5gs  
Password: open5gs

Open menu on the left, click on *Dashboards*. Select *Current state Dash* and you can see the current state of your 5G network.

Example:



Fig. 1: Grafana dashboard



## NORMAL SURFING

`uc1.run_uc1()`

Run UC1: Normal Surfing scenario.

Simulates user behavior with intermittent UE connectivity and randomized data downloads to mimic typical mobile web browsing patterns.

### Scenario Summary

- Starts half of the UEs.
- **UEs may**
  - download random chunks of data (5–50MB),
  - randomly disconnect or reconnect during the session.
- The scenario runs for a randomized session duration (60–600 seconds).
- Active UEs are defined in *nr-ue{i}.yaml* Docker Compose files.
- Writes the current UC label to *data/current\_uc.txt*.

### Args

None

### Returns

None

## VIDEO STREAMING

`uc2.run_uc2()`

Run UC2: Video Streaming scenario.

Simulates a typical video streaming session where all UEs continuously receive data. This scenario helps to evaluate throughput and session stability under constant load.

### Scenario Summary

- Starts 4 UEs using Docker Compose.
- Each UE downloads 2MB of random data every second.
- Streaming duration is randomized between 300 and 600 seconds.
- The UC label is logged into 'data/current\_uc.txt'.

### Args

None

### Returns

None

## PERIODIC KEEP-ALIVE

`uc3.run_uc3()`

Run UC3: Periodic Keep-Alive scenario.

Simulates multiple UEs that periodically send HTTP requests (keep-alive pings) to a remote server. This pattern reflects real-world background traffic in mobile applications (e.g. chat apps, weather updates).

### Scenario Summary

- Starts 4 UEs as containers using Docker Compose.
- Each UE sends periodic HTTP GET requests (via *curl*) to a predefined URL.
- The interval between pings is randomized between 30–35 seconds.
- Simulation duration is randomly set between 300–600 seconds.
- Scenario type is logged into 'data/current\_uc.txt'.

### Args

None

### Returns

None

## SHORT BURST SESSIONS

### `uc4.run_uc4()`

Run UC4: Short Burst Sessions scenario.

Simulates multiple UEs initiating brief data sessions at random intervals. This pattern mimics sporadic, high-frequency user actions (e.g. short API requests, fast-loading web content) in mobile networks.

#### Scenario Summary

- Randomly selects a UE out of 4 available.
- Starts a short Docker container session for the selected UE.
- Simulates a 2MB data transfer using *dd* inside the container.
- After 2–4 seconds, the UE container is stopped.
- Waits 3–6 seconds and repeats until the scenario duration ends.
- Marks current use case in 'data/current\_uc.txt'.

#### Args

None

#### Returns

None

## LOAD REGISTRATION ANOMALY

`uc5.run_uc5()`

Run UC5: Load Registration Anomaly scenario.

Simulates a stress event in the 5G network by concurrently connecting multiple UEs. This tests the network's ability to handle sudden, simultaneous registration attempts — a common anomaly in overloaded environments.

### Scenario Summary

- Starts 4 UE containers simultaneously using *subprocess.Popen*.
- Waits for all containers to fully initialize.
- Holds all UE sessions active for a short time (5 seconds by default).
- Stops all containers simultaneously after the wait period.
- Logs scenario activity to 'data/current\_uc.txt'.

### Args

None

### Returns

None

## AUTHENTICATION FAILURE ALERT

`uc6.run_uc6()`

Run UC6: Authentication Failure Alert scenario.

Simulates an authentication failure event in a 5G network by repeatedly starting a misconfigured UE (User Equipment) that fails to register due to incorrect credentials or malformed configuration. This scenario is useful for testing network response to repeated failed attempts and monitoring for anomaly detection mechanisms.

### Scenario Summary

- Launches a specially prepared UE (ID=100) which is expected to fail authentication.
- Repeats the process a random number of times (3 to 6 retries).
- Between each attempt, the UE is stopped and a random interval (5–30 seconds) is observed.
- Total scenario duration is also bounded by a global timeout (120–300 seconds).
- All events are logged to the console and scenario type is saved to *data/current\_uc.txt*.

### Args

None

### Returns

None

## LSTM MODEL WITH ATTENTION

```
class lstm_attention_model.AttentionLayer(*args, **kwargs)
```

Bases: Layer

Custom attention layer compatible with LSTM outputs. Outputs a weighted sum across the time dimension.

Source: <https://www.geeksforgeeks.org/adding-attention-layer-to-a-bi-lstm/>

```
build(input_shape)
```

Build the attention layer.

**Args**

- input\_shape: Shape of the input tensor.

**Returns**

None

```
call(x)
```

Apply the attention mechanism to the input tensor.

**Args**

- x: Input tensor of shape (batch\_size, timesteps, features).

**Returns**

- output: Weighted sum of the input tensor across the time dimension.

```
compute_output_shape(input_shape)
```

Compute the output shape of the attention layer.

**Args**

- input\_shape: Shape of the input tensor.

**Returns**

- output\_shape: Shape of the output tensor.

`lstm_attention_model.build_attention_model(input_shape, num_classes)`

Build and return an attention-based LSTM model.

**Args**

- `input_shape`: tuple, shape of the input data (timesteps, features)
- `num_classes`: int, number of output classes

**Returns**

- Keras Model instance

`lstm_attention_model.train_attention_model()`

LSTM model with custom attention mechanism for multi-class classification of time-series data.

This module defines a deep learning model using TensorFlow and Keras, integrates a custom attention mechanism, and trains the model on preprocessed input data with categorical labels.

**Expected data format:**

- Input: `X_train.npy`, `X_test.npy` (shape: [samples, 60, features])
- Labels: `y_train.npy`, `y_test.npy` (categorical class indices)
- Class weights: `class_weights.json`

The trained model is saved as HDF5 and Keras formats.



## LSTM BATHNORM MODEL

`lstm_bathnorm_model.train_batchnorm_model()`

LSTM BatchNorm Model

This module defines a deep learning model using TensorFlow and Keras with Batch Normalization for multi-class classification of time-series data.

**Expected data format:**

- Input: `X_train.npy`, `X_test.npy` (shape: [samples, 60, features])
- Labels: `y_train.npy`, `y_test.npy` (categorical class indices)
- Class weights: `class_weights.json`

The trained model is saved as HDF5 and Keras formats.

## LSTM BASE MODEL

`lstm_base_model.build_base_model(X_train, y_train, X_test, y_test, class_weight_dict)`

LSTM Base Model for classification of network use case scenarios.

This script loads preprocessed training and testing data, defines and trains a baseline LSTM model, evaluates its performance, and saves the final model in HDF5 and Keras formats. Class balancing is handled using precomputed class weights.

### Usage

- This script is designed to be executed as a module. Use the function `build_base_model()` to construct and optionally train the model.

### Args

- `X_train` (numpy.ndarray): Preprocessed training data.
- `y_train` (numpy.ndarray): Labels for the training data.
- `X_test` (numpy.ndarray): Preprocessed testing data.
- `y_test` (numpy.ndarray): Labels for the testing data.
- `class_weight_dict` (dict): Class weights for handling class imbalance.

### Returns

- `model` (tensorflow.keras.Model): Trained LSTM model.

## LSTM ROBUST MODEL

`lstm_robust_model.build_robust_model()`

Trains a robust LSTM model and saves it to the *trained\_models/* directory. The model consists of 3 LSTM layers with varying dropout rates and 2 Dense layers. It is designed for classifying UC classes based on preprocessed sequential inputs.

**Args**

None

**Returns**

None

The function saves the trained model to the *trained\_models/* directory and generates visualizations for the confusion matrix and training history.

## CLASSIFY THE REAL DATA USING LSTM

Module for evaluating multiple LSTM models (base, robust, batchnorm, attention) on real-world 5G network data, with the option to fine-tune the attention model.

**This script includes**

- Definition of a custom attention layer,
- Real-data class weight computation,
- Sequence generation for LSTM input,
- Model accuracy evaluation via classification report,
- Optional fine-tuning of the attention-based LSTM model.

**class** lstm\_results\_real\_data.**AttentionLayer**(\*args, \*\*kwargs)

Bases: Layer

Custom attention layer compatible with LSTM outputs. Outputs a weighted sum across the time dimension.

Source: <https://www.geeksforgeeks.org/adding-attention-layer-to-a-bi-lstm/>

**build**(input\_shape)

Build the attention layer.

**Args**

- input\_shape: Shape of the input tensor.

**Returns**

None

**call**(x)

Apply the attention mechanism to the input tensor.

**Args**

- x: Input tensor of shape (batch\_size, timesteps, features).

**Returns**

- output: Weighted sum of the input tensor across the time dimension.

**compute\_output\_shape**(input\_shape)

Compute the output shape of the attention layer.

**Args**

- input\_shape: Shape of the input tensor.

**Returns**

### Digital Twin of 5G Network, Technical documentation

---

- output\_shape: Shape of the output tensor.

`lstm_results_real_data.create_sequences(X, y, seq_len=60)`

Converts flattened input arrays into sliding window sequences for LSTM input.

#### Args

- X (np.ndarray): Input features of shape (samples, features).
- y (np.ndarray): Target labels corresponding to input samples.
- seq\_len (int): Length of each sequence window. Default is 60.

#### Returns

- tuple: (X\_seq, y\_seq) where X\_seq has shape (samples - seq\_len, seq\_len, features) and y\_seq has shape (samples - seq\_len,).

`lstm_results_real_data.evaluate_model(model, X_seq, y_seq, name)`

Evaluates a trained model on given sequential data and prints classification report.

#### Args

- model (keras.Model): The trained Keras model to evaluate.
- X\_seq (np.ndarray): Sequential input data (samples, seq\_len, features).
- y\_seq (np.ndarray): True class labels.
- name (str): Name of the model for display purposes.

#### Returns

None

`lstm_results_real_data.load_and_preprocess_data()`

Loads the real-world labeled dataset and applies categorical mappings.

#### Args

None

#### Returns

- pd.DataFrame: Preprocessed dataset with mapped categorical columns and timestamps.

`lstm_results_real_data.run_evaluation_and_finertuning()`

Main function to evaluate four trained LSTM models and fine-tune the attention model using real labeled data.

#### Args

None

#### Returns

None

## EXPLORATORY DATA ANALYSIS

EDA module for exploratory analysis of synthetic and real 5G network datasets.

This module contains functions to load data, preprocess it, visualize it, and perform feature selection using multiple strategies including RF, RFE, RFECV, SFS and permutation importance.

Functions in this module should be called explicitly from a main script or notebook.

**eda.compute\_class\_weights(y)**

Compute class weights for imbalanced classes.

**Args**

- `y` (pd.Series): Target variable.

**Returns**

- dict: Dictionary mapping class labels to their corresponding weights.

**eda.load\_dataset(path)**

Load dataset from CSV file.

**Return type**

DataFrame

**Args**

- `path` (str): Path to the CSV file.

**Returns**

- pd.DataFrame: Loaded dataset.

**eda.load\_maps(log\_map\_path='./json/log\_map.json', app\_map\_path='./json/app\_map.json',  
uc\_map\_path='./json/uc\_map.json')**

Load mapping dictionaries from JSON files.

**Args**

- `log_map_path` (str): Path to the log type mapping JSON file.
- `app_map_path` (str): Path to the application mapping JSON file.
- `uc_map_path` (str): Path to the use case mapping JSON file.

**Returns**

- **tuple: A tuple containing three dictionaries:**
  - `log_map` (dict): Mapping of log types to integers.

### Digital Twin of 5G Network, Technical documentation

---

- `app_map` (dict): Mapping of applications to integers.
- `uc_map` (dict): Mapping of use cases to integers.

`eda.permutation_importance_stable(X, y, selected_features, n_runs=10)`

Calculate stable permutation importances over multiple runs.

#### Args

- `X` (pd.DataFrame): Feature matrix.
- `y` (pd.Series): Target variable.
- `selected_features` (list): List of selected feature names.
- `n_runs` (int): Number of runs for stability.

#### Returns

- pd.DataFrame: DataFrame containing mean and std of importances.

`eda.preprocess_data(df, log_map, app_map, uc_map)`

Preprocess dataset: fill NA, map strings to ints, scale numeric columns.

#### Args

- `df` (pd.DataFrame): Input DataFrame to preprocess.
- `log_map` (dict): Mapping of log types to integers.
- `app_map` (dict): Mapping of applications to integers.
- `uc_map` (dict): Mapping of use cases to integers.

#### Returns

- **tuple: A tuple containing:**
  - `X_scaled` (np.ndarray): Scaled feature matrix.
  - `X` (pd.DataFrame): Original feature matrix.
  - `y` (pd.Series): Target variable.

`eda.random_forest_importance(X_scaled, X, y)`

Train Random Forest and return feature importances.

#### Args

- `X_scaled` (np.ndarray): Scaled feature matrix.
- `X` (pd.DataFrame): Original feature matrix.
- `y` (pd.Series): Target variable.

#### Returns

- pd.Series: Feature importances sorted in descending order.
- RandomForestClassifier: Trained Random Forest model.

`eda.rfe_selection(X_scaled, y, X, rf)`

Recursive Feature Elimination.

#### Args

- `X_scaled` (np.ndarray): Scaled feature matrix.
- `y` (pd.Series): Target variable.

- X (pd.DataFrame): Original feature matrix.
- rf (RandomForestClassifier): Trained Random Forest model.

**Returns**

- pd.Series: Boolean mask indicating selected features.

`eda.rfecv_selection(X_scaled, y, X, rf)`

RFECV - RFE with cross-validation.

**Args**

- X\_scaled (np.ndarray): Scaled feature matrix.
- y (pd.Series): Target variable.
- X (pd.DataFrame): Original feature matrix.
- rf (RandomForestClassifier): Trained Random Forest model.

**Returns**

- pd.Series: Boolean mask indicating selected features.

`eda.sfs_selection(X_scaled, y, X, rf)`

Sequential Feature Selector.

**Args**

- X\_scaled (np.ndarray): Scaled feature matrix.
- y (pd.Series): Target variable.
- X (pd.DataFrame): Original feature matrix.
- rf (RandomForestClassifier): Trained Random Forest model.

**Returns**

- pd.Series: Boolean mask indicating selected features.



## LABEL REAL DATASET

`add_current_uc.apply_uc(row)`

Assign a Use Case (UC) label to a row based on its timestamp.

**Args**

- `row` (pd.Series): A row with a 'timestamp' field.

**Returns**

- `str`: The UC label ('uc1' to 'uc6').

`add_current_uc.compare_intervals(row, interval)`

Compare a timestamp from a row with a given time interval.

**Args**

- `row` (pd.Series): Row from the DataFrame with a 'timestamp' column.
- `interval` (dict): Dictionary with "from" and "to" datetime strings.

**Returns**

- `bool`: True if timestamp is within interval, else False.

`add_current_uc.label_realnetwork_csv(input_path='./datasets/real_network_data_before_labeling.csv',  
output_path='./datasets/real_network_data_after_labeling.csv')`

Load the CSV, assign UC labels to each row, and save the updated file.

**Args**

- `input_path` (str): Path to the input CSV file.
- `output_path` (str): Path to save the labeled CSV file.

**Returns**

None

## PREPROCESS DATA FOR LSTM

Module for preparing LSTM input data from preprocessed features. Includes functionality for loading feature arrays, creating sequences, splitting the dataset, and saving the output for training and testing.

`lstm_preprocessing.create_sequences(X, y, seq_len)`

Creates sequences of input data for LSTM using a sliding window.

**Args**

- **X** (*np.ndarray*) – Input data (features)
- **y** (*np.ndarray*) – Target values (classes)
- **seq\_len** (*int*) – Length of the sequence for LSTM

**Returns**

- tuple: (X\_seq, y\_seq) as ndarray

`lstm_preprocessing.load_data(X_path, Y_path, scaler_path, features_path, uc_map_path)`

Loads the preprocessed data from specified paths.

**Args**

- **X\_path** (str): Path to the input features (X)
- **Y\_path** (str): Path to the target labels (y)
- **scaler\_path** (str): Path to the scaler object
- **features\_path** (str): Path to the selected features JSON file
- **uc\_map\_path** (str): Path to the UC map JSON file

**Returns**

- tuple: (X, y, scaler, selected\_features, uc\_map)

`lstm_preprocessing.split_and_save_data(X_seq, y_seq, output_dir='preprocessed_data')`

Splits the dataset into training and testing sets and saves them to disk.

**Args**

- **X\_seq** (*np.ndarray*): Input features in sequence format
- **y\_seq** (*np.ndarray*): Target labels in sequence format
- **output\_dir** (str): Directory to save the split data

**Returns**

None

## SIMULATE A RUNNING NETWORK

`running_network.init_log()`

Initializes the log file and directory if needed.

**Args**

None

**Returns**

None

`running_network.run_simulation(script_name)`

Runs the selected UC simulation and logs its outcome.

**Args**

- `script_name` (str): Name of the UC script to run.

**Returns**

None

## MAIN ORCHESTRATOR

**class** network\_watcher.**AttentionLayer**(\*args, \*\*kwargs)

Bases: Layer

Custom attention layer for LSTM model. This layer computes the attention weights and applies them to the input sequence.

**Args**

- Layer (tf.keras.layers.Layer): Base class for all layers in Keras.

**Returns**

None

**build**(input\_shape)

Create the attention weights and bias.

**Args**

- input\_shape (tuple): Shape of the input tensor.

**Returns**

None

**call**(x)

Calculate the attention weights and apply them to the input sequence.

**Args**

- x (tensor): Input tensor of shape (batch\_size, sequence\_length, features).

**Returns**

- tensor: Output tensor of shape (batch\_size, features).

**compute\_output\_shape**(input\_shape)

Compute the output shape of the layer.

**Args**

- input\_shape (tuple): Shape of the input tensor.

**Returns**

- tuple: Shape of the output tensor.

network\_watcher.**clean\_old\_models**(directory='/app/data/Model', keep\_last\_n=7,  
pattern='Model\_bn\_\*.keras')

Keep only the last *keep\_last\_n* saved model files and delete older ones.

**Args**

### Digital Twin of 5G Network, Technical documentation

---

- `directory` (str): Directory containing the model files.
- `keep_last_n` (int): Number of recent models to keep.
- `pattern` (str): Pattern to match model files.

#### Returns

None

`network_watcher.load_last_sequence(csv_path, selected_features, sequence_length=60)`

Load the last sequence of records from CSV, ensuring all required features and labels are present.

#### Args

- `csv_path` (str): Path to the CSV file.
- `selected_features` (list): List of features to select from the DataFrame.
- `sequence_length` (int): Length of the sequence to load.

#### Returns

- tuple: DataFrame with selected features and the correct labels.

#### Raises

- `ValueError`: If any of the selected features are missing in the DataFrame.

`network_watcher.main_loop(interval=1, prometheus_port=9000)`

Main loop that monitors UE activity, parses logs, updates Prometheus metrics, and fine-tunes the model in real-time.

#### Args

- `interval` (int): Time interval for monitoring and updating metrics.
- `prometheus_port` (int): Port for Prometheus metrics.

#### Returns

None

`network_watcher.parse_amf(lines, previous_state)`

Parse AMF log lines to extract UE registration and deregistration events, update UE states, and compute registration/session durations.

#### Args

- `lines` (list): List of log lines to parse.
- `previous_state` (dict): Previous state of UE details and durations.

#### Returns

- tuple: Updated UE details, new registration durations, and new session durations.

`network_watcher.predict_current_uc(latest_window_df)`

Predict the current use case (UC) using the loaded LSTM model based on the latest data window.

#### Args

- `latest_window_df` (pd.DataFrame): DataFrame containing the latest data window.

#### Returns

- tuple: Predicted UC class and confidence score.

`network_watcher.remove_offset()`

Remove the offset file used by Pygtail to start reading the log file from the beginning.

**Args**

None

**Returns**

None

`network_watcher.run_main_notebook_with_backup()`

Execute the main.ipynb notebook, log its execution, truncate CSV, and periodically create CSV backups.

**Args**

None

**Returns**

None

`network_watcher.run_notebook_in_thread()`

Run the main.ipynb notebook in a separate thread to avoid blocking the main loop.

**Args**

None

**Returns**

None

`network_watcher.save_model_with_date(model, path_prefix='/app/data/Model/Model_bn_')`

Save the current model to disk with the current date as part of the filename.

**Args**

- `model` (tf.keras.Model): The model to save.
- `path_prefix` (str): Prefix for the filename.

**Returns**

None

`network_watcher.truncate_running_data(csv_path, keep_last_n=60)`

Truncate the CSV file to keep only the last `keep_last_n` records.

**Args**

- `csv_path` (str): Path to the CSV file.
- `keep_last_n` (int): Number of records to keep.

**Returns**

None

