

**Technická univerzita v Košiciach**  
**Fakulta elektrotechniky a informatiky**

**Ladenie hyperparametrov GRU modelu**

**2025**

**Nikita Chernysh**

**Technická univerzita v Košiciach**

**Fakulta elektrotechniky a informatiky**

**Ladenie hyperparametrov GRU modelu**

Študijný program: Inteligentné systémy  
Študijný odbor: Inteligentné systémy  
Školiace pracovisko: Katedra kybernetiky a umelej inteligencie (KKUI)  
Školiteľ: Ing. Miroslava Matejová  
Konzultant: Ing. Miroslava Matejová

**Košice 2025**

**Nikita Chernysh**

## **Abstrakt v SJ**

Táto práca sa zaobrá hľadaním, optimalizáciou a testovaním hyperparametrov modelu rekurentnej neurónovej siete (RNN) založeného na architektúre GRU (Gated Recurrent Unit). Cieľom bolo vyvinúť všeobecný algoritmus schopný pracovať s ľubovoľnými dátami, pričom všetky metriky hodnotenia a výpočty boli implementované s využitím základných matematických knižníc. Výnimku tvorí samotný GRU model. Praktické overenie navrhnutého prístupu bolo realizované na reálnom datasete o prenájmoch bicyklov v Londýne. Výsledky experimentov potvrdzujú efektivitu vybraných metód optimalizácie hyperparametrov (grid search, random search a Bayesian search).

## **Klúčové slová**

Ladenie, GRU, RNN, hyperparametre, bayesovské vyhľadávanie, náhodné vyhľadávanie, vyhľadávanie v sieti

## **Abstrakt v AJ**

This work focuses on finding, optimizing, and testing hyperparameters of a recurrent neural network (RNN) model based on the GRU (Gated Recurrent Unit) architecture. The goal was to develop a universal algorithm capable of working with arbitrary datasets, with all evaluation metrics and calculations implemented using basic mathematical libraries. The exception is the GRU model itself. Practical verification of the proposed approach was conducted on a real-world dataset of bike rentals in London. Experimental results confirm the effectiveness of selected hyperparameter optimization methods (grid search, random search, and Bayesian search).

## **Klúčové slová v AJ**

Optimization, debugging, neural networks, GRU, RNN, hyperparameters, bayesian search, random search, grid search

# Zadanie práce

## Zadanie Strojové učenie

Cieľom zadania je naprogramovať niektorý z preberaných algoritmov strojového učenia a prakticky aplikovať získané vedomosti.

- **Nie je dovolené** využívať naprogramované funkcie algoritmu z nejakej knižnice.
- Ani **metriky** ako Presnosť, Správnosť, Návratnosť, F1 miera nemôžete implementovať z knižnice, ale **musíte sami naprogramovať**.
- Pri nekontrolovanom učení môžete využiť na vyhodnotenie iné metriky, ktoré môžete implementovať z knižníc (kedže tie tie sú preberané).
- Algoritmus má byť **naprogramovaný všeobecne**. Nemá byť prispôsobený na konkrétné dátá, ale na ľubovoľnú dátovú množinu.
- **Dátová množina** pre experimenty má byť **reálna/z praxe**. Dáta z cvičenia nestačia. **Dáta majú mať aspoň 4 popisné atribúty (+ cieľový atribút) a aspoň 20 záznamov**. V odôvodnených a prekonzultovaných prípadoch to môže byť aj inak.
- Kód má obsahovať **komentáre** v slovenčine.
- Dokumentácia má obsahovať:
  - o Teoretický popis algoritmu
  - o Popis postupu a jednotlivých funkcií
  - o Popis dát (vysvetlenie, popis atribútov, počet záznamov, zdroj)
  - o Vyhodnotenie (metriky, interpretácia výsledkov, záver)
- **Dokumentácia** má byť napísaná **v šablóne** záverečných prác (aby ste sa naučili s ňou pracovať).
- Všetko čo ste robili má byť popísané v dokumentácii.
- Pozor na správne citovanie zdrojov.
- Hodnotí sa aj **množstvo vašej práce a úsilia**.
- Na obhajobu zadania máte **1 pokus**.
- Vytvorený model nasaďte v ľubovoľnej vami vytvorenej aplikácii

## **Čestné vyhlásenie**

Vyhlasujem, že som zadanie vypracoval(a) samostatne s použitím uvedenej odbornej literatúry.

Košice 9. 5. 2025

.....  
*Vlastnoručný podpis*

## Predhovor

Táto semestrálna práca bola vypracovaná v rámci predmetu „Strojové učenie“ na Fakulte elektrotechniky a informatiky Technickej univerzity v Košiciach. Jej hlavným cieľom bolo prakticky aplikovať nadobudnuté vedomosti z oblasti strojového učenia na implementáciu rekurentnej neurónovej siete založenej na architektúre GRU (Gated Recurrent Unit), a zároveň splniť všetky stanovené obmedzenia – najmä zákaz využívania hotových knižničných implementácií algoritmov a metrík.

Počas práce bol kladený dôraz na všeobecnú aplikovateľnosť algoritmu na akýkoľvek dataset, nezávisle od jeho štruktúry či pôvodu. Navyše, implementácia všetkých metrík, výpočtov a pomocných nástrojov bola realizovaná výlučne pomocou základných matematických knižníc ako `numpy` a `pandas`. Výnimku tvorila iba samotná architektúra GRU, ktorá bola implementovaná s využitím knižnice `PyTorch`, keďže vytvorenie GRU blokov od základov by výrazne presahovalo rozsah zadania.

Reálny dataset použitý na experimenty predstavuje historické záznamy o prenájme bicyklov v Londýne. Práca zahŕňa kompletný životný cyklus vývoja modelu – od predspracovania dát, cez návrh modelu, optimalizáciu hyperparametrov (s využitím viacerých stratégií), až po vyhodnotenie výsledkov a ich vizualizáciu. Cieľom tejto práce bolo nielen technicky zvládnuť implementáciu, ale aj získať hlbšie porozumenie fungovaniu GRU modelov v kontexte sekvenčných dát.

Toto cestou by som sa chcel podakovať vyučujúcim predmetu za odborné vedenie a podporu počas semestra.

# **Obsah**

<b>Úvod</b>	<b>1</b>
<b>1 Formulácia úlohy</b>	<b>2</b>
<b>2 Analýza existujúcich riešení a metód</b>	<b>3</b>
2.1 Metódy optimalizácie hyperparametrov . . . . .	3
2.2 Analýza dostupných metrík hodnotenia . . . . .	4
2.3 Analýza realizácií GRU modelov . . . . .	5
<b>3 Experimenty</b>	<b>6</b>
3.1 Dátová množina (Dataset) [1] . . . . .	6
3.1.1 Nájdenie dátovej množiny . . . . .	6
3.1.2 Realizácia EDA . . . . .	6
3.1.3 Predspracovanie a unifikácia dát . . . . .	9
3.1.4 Vytvorenie univerzálneho dátového načítača . . . . .	10
3.2 Implementácia flexibilného GRU modelu . . . . .	10
3.3 Realizácia metrík . . . . .	11
3.4 Priebeh experimentov . . . . .	12
3.4.1 Počiatočná siet hyperparametrov . . . . .	12
3.4.2 Výsledky Random Search . . . . .	13
3.4.3 Výsledky Bayesian Search . . . . .	14
3.4.4 Analýza najlepších sád hyperparametrov . . . . .	15
3.4.5 Grid Search optimalizácia . . . . .	16
<b>4 Analýza a vizualizácia výsledkov</b>	<b>17</b>
4.1 Vizualizácia výsledkov Random Search . . . . .	17
4.1.1 Progres vyhľadávania . . . . .	18
4.1.2 Vztah medzi epochami a stratou (Loss) . . . . .	19
4.1.3 Vplyv veľkosti skrytých vrstiev (Hidden size) . . . . .	20

4.1.4	Krivky učenia (Learning curves) . . . . .	21
4.1.5	Korelácia medzi parametrami a výkonom . . . . .	22
4.2	Vizualizácia výsledkov Bayesian Search . . . . .	23
4.2.1	Progres vyhľadávania (Search Progress) . . . . .	23
4.3	Vizualizácia výsledkov Grid Search . . . . .	24
4.3.1	Progres vyhľadávania (Search Progress) . . . . .	24
4.3.2	Krivky učenia (Learning Curves) . . . . .	25
4.3.3	Vztah medzi epochami a stratou (Epochs vs Loss) . . . . .	26
4.3.4	Vplyv skrytej vrstvy (Hidden Size Influence) . . . . .	27
4.3.5	Korelácia parametrov s výkonom (Parameter Correlation) . .	28
4.4	Vizualizácia výsledkov modelu natrénovaného na najlepších paramet- roch . . . . .	29
4.4.1	Evolúcia chýb a dynamika tréningu . . . . .	29
4.4.2	Dynamické prispôsobovanie učiacej rýchlosťi . . . . .	30
4.4.3	Porovnanie tréningových a validačných metrík . . . . .	31
4.4.4	Komplexná metrická vizualizácia . . . . .	32
4.4.5	Porovnanie výkonu modelu v priebehu epoch . . . . .	33
<b>5</b>	<b>Záver</b>	<b>34</b>
5.1	Zhrnutie dosiahnutých výsledkov . . . . .	34
5.2	Hodnotenie kvality riešenia . . . . .	34
5.3	Odporúčania a návrhy na zlepšenie . . . . .	35
5.4	Záverečné zhodnotenie . . . . .	35
<b>Zoznam príloh</b>		<b>37</b>
<b>Príloha A</b>		<b>38</b>

## Zoznam obrázkov

3–1	Distribution of Bike Rentals . . . . .	8
3–2	Box Plot of Bike Rentals . . . . .	9
3–3	Average Daily Bike Rentals . . . . .	9
4–1	Progres výhľadávania . . . . .	18
4–2	Epochs vs Loss . . . . .	19
4–3	Hidden size influence . . . . .	20
4–4	Learning curves . . . . .	21
4–5	Parameter correlation . . . . .	22
4–6	Search progress . . . . .	23
4–7	Search progress . . . . .	24
4–8	Learning curves . . . . .	25
4–9	Epochs vs Loss . . . . .	26
4–10	Hidden size influence . . . . .	27
4–11	Parameter correlation . . . . .	28
4–12	Best model error metrics evolution . . . . .	29
4–13	Best model learning rate schedule . . . . .	30
4–14	Best model training and validation metrics comparison . . . . .	31
4–15	Best model radar chart . . . . .	32
4–16	Metrics comparison over epochs . . . . .	33

## **Zoznam tabuľiek**

3–1 Niekolko prvých riadkov datasetu . . . . .	7
3–2 Prehľad typov dát v datasete . . . . .	7
3–3 Štatistické zhrnutie datasetu . . . . .	8
3–4 Počiatočná siet hyperparametrov . . . . .	12
3–5 Výsledky Random Search . . . . .	13
3–6 Výsledky Bayesian Search . . . . .	14
3–7 Najlepšie sady hyperparametrov z Random a Bayesian Search . . . . .	15
3–8 Grid Search optimalizačná siet . . . . .	16
3–9 Najlepšie hyperparametre z Grid Search . . . . .	17

## Slovník termínov

**Strojové učenie (Machine Learning)** – Oblast umelej inteligencie, ktorá sa zaoberá tvorbou algoritmov schopných učiť sa z dát a zlepšovať svoje výstupy bez nutnosti explicitného programovania všetkých krokov.

**Rekurentná neurónová sieť (RNN)** – Typ neurónovej siete navrhnutý na spracovanie sekvenčných dát. Umožňuje uchovávanie informácií z predchádzajúcich stavov prostredníctvom vnútorných pamäťových štruktúr.

**GRU (Gated Recurrent Unit)** – Variant rekurentnej neurónovej siete, ktorý využíva mechanizmy brán (update a reset gate) na efektívnejšie učenie dlhodobých závislostí a zmierňuje problém miznúcich gradientov.

**Hyperparameter** – Parameter, ktorý nie je učený modelom, ale je nastavený pred samotným tréningom. Zahŕňa napríklad veľkosť skrytého stavu, počet vrstiev alebo rýchlosť učenia.

**Optimalizácia hyperparametrov** – Proces hľadania najvhodnejšej kombinácie hyperparametrov, ktorý vedie k zlepšeniu výkonnosti modelu na validačných dátach.

**MAE (Mean Absolute Error)** – Priemerná absolútна chyba medzi predikovanými a skutočnými hodnotami. Udáva sa v rovnakých jednotkách ako cieľová premenná.

**MSE (Mean Squared Error)** – Priemer štvorcov rozdielov medzi predikciami a skutočnými hodnotami. Zvýrazňuje väčšie odchýlky.

**RMSE (Root Mean Squared Error)** – Druhá odmocnina z MSE. Interpretovateľná v rovnakých jednotkách ako cieľová premenná, citlivejšia na veľké chyby.

**R<sup>2</sup> (Koeficient determinácie)** – Miera vysvetlenej variability cieľovej premennej modelom. Hodnoty blízke 1 indikujú vysokú presnosť modelu.

**MAPE (Mean Absolute Percentage Error)** – Priemerná absolútна percentuálna chyba.

Vyjadruje chybu v percentách voči skutočným hodnotám.

**SMAPE (Symmetric Mean Absolute Percentage Error)** – Symetrická verzia MAPE, ktorá redukuje skreslenie pri malých hodnotách.

**Explained Error (Explained Variance Score)** – Ukazovateľ, do akej miery variabilita predikcií vysvetluje variabilitu skutočných hodnôt.

**Peak Error** – Maximálna absolútna chyba v rámci celej testovacej množiny. Používa sa na odhalenie najväčších extrémnych odchýlok.

**Sekvenčné dátá** – Typ dát, kde je dôležité poradie záznamov. Typickými príkladmi sú časové rady alebo textové reťazce.

**Tréningová množina** – Časť dát použiteľná na učenie modelu.

**Testovacia množina** – Časť dát určená na nezávislé vyhodnotenie výkonnosti modelu po trénovaní.

**Validácia** – Proces priebežného vyhodnocovania modelu na samostatnej množine za účelom výberu najvhodnejších hyperparametrov a predchádzania pretrénovania.

## Úvod

Význam strojového učenia neustále narastá, čo vedie k vzniku množstva rôznych modelov a architektúr schopných riešiť široké spektrum úloh, od predikcie časových radov po spracovanie prirodzeného jazyka. Medzi najpopulárnejšie modely patria rekurentné neurónové siete (RNN), ktoré sú vhodné na spracovanie sekvenčných dát vďaka schopnosti pamätať si predchádzajúce vstupy. Jednou z efektívnych a rozšírených architektúr RNN je Gated Recurrent Unit (GRU).

Táto práca sa zameriava na problematiku optimalizácie hyperparametrov GRU modelov. Správny výber a nastavenie hyperparametrov má zásadný vplyv na výkonnosť a presnosť modelu. Implementácia musí byť dostatočne univerzálna na použitie s rôznymi dátovými množinami bez nutnosti zásadných úprav kódu. V rámci zadania bolo nutné všetky metriky hodnotenia implementovať manuálne, čo umožňuje hlbšie pochopenie a kontrolu nad celým procesom učenia.

## 1 Formulácia úlohy

Hlavným cieľom tejto práce bolo navrhnúť, implementovať a otestovať algoritmus pre vyhľadávanie, optimalizáciu a hodnotenie hyperparametrov modelu RNN využívajúceho GRU architektúru. Konkrétnie boli stanovené nasledovné ciele:

- Implementovať univerzálny algoritmus pre optimalizáciu hyperparametrov GRU, schopný aplikácie na ľubovoľný dataset.
- Vytvoriť manuálne implementované metriky hodnotenia pomocou knižníc numpy a pandas.
- Overiť algoritmus na reálnych dátach o prenájme bicyklov v Londýne.
- Porovnať efektívnosť rôznych metód optimalizácie hyperparametrov, konkrétnie grid search, random search a Bayesian search.

## 2 Analýza existujúcich riešení a metód

V tejto kapitole sa zaoberáme analýzou existujúcich prístupov k optimalizácii hyperparametrov rekurentných neurónových sietí s dôrazom na architektúru GRU. Podrobne skúmame hlavné metódy optimalizácie hyperparametrov a analyzujeme ich výhody a nevýhody v kontexte použitia na rôznych dátových množinách.

### 2.1 Metódy optimalizácie hyperparametrov

Optimálne nastavenie hyperparametrov je klíčové pre dosiahnutie vysokej presnosti modelov strojového učenia. Existujú rôzne prístupy na riešenie tejto úlohy:

**Grid Search** predstavuje systematické skúmanie vopred definovanej množiny hyperparametrov, kde každý parameter je testovaný v rôznych kombináciach s ostatnými. Táto metóda je jednoduchá na implementáciu, ale neefektívna pri veľkom počte hyperparametrov, keďže počet potrebných výpočtov rastie exponenciálne.

**Random Search** vykonáva náhodný výber kombinácií hyperparametrov z definovaného rozsahu. Táto metóda je efektívnejšia ako grid search v prípadoch s veľkým počtom parametrov, pretože poskytuje širšie pokrytie priestoru hyperparametrov s menším počtom iterácií.

**Bayesian Search** využíva Bayesovské metódy na riadenie výberu kombinácií hyperparametrov. Táto metóda postupne zlepšuje svoj výber na základe predchádzajúcich výsledkov a je efektívna najmä pri zložitých modeloch, kde tradičné metódy optimalizácie môžu byť neefektívne.

Postup optimalizácie bol nasledovný: najprv bol vykonaný random a Bayesian search na rovnakom rozsahu hyperparametrov. Následne boli tri najlepšie sady hyperparametrov z oboch metód kombinované a pomocou grid search bola nájdená najlepšia výsledná kombinácia. Tento prístup umožnil efektívne zúženie priestoru hľadania a zvýšenie presnosti finálneho výberu hyperparametrov.

## 2.2 Analýza dostupných metrík hodnotenia

Pre vyhodnotenie výkonu modelov strojového učenia sa používajú rôzne metriky. V rámci tejto práce sme manuálne implementovali niekoľko štandardných metrík (3.3) s využitím knižníc numpy a pandas:

- **MSE (Mean Squared Error)**: Priemerná štvorcová chyba meria priemernú veľkosť chyby medzi predikovanými a reálnymi hodnotami.
- **MAE (Mean Absolute Error)**: Priemerná absolútна chyba, robustnejšia metrika voči odľahlým hodnotám.
- **RMSE (Root Mean Squared Error)**: Odmocnina z MSE, poskytuje chybu v jednotkách pôvodnej veličiny, čím zlepšuje interpretovateľnosť výsledkov.
- **R-squared (koeficient determinácie)**: Udáva mieru zhody medzi modelom a dátami, pričom vyššie hodnoty naznačujú lepšiu predikčnú schopnosť.
- **MAPE (Mean Absolute Percentage Error)**: Priemerná absolútna percentuálna chyba, vhodná na vyhodnotenie relatívnej veľkosti chyby.
- **sMAPE (Symmetric Mean Absolute Percentage Error)**: Symetrická verzia MAPE, vhodná pre symetrické posúdenie chyby.
- **Explained Variance**: Vyjadruje podiel vysvetlenej variancie modelom oproti celkovej variancii dát.
- **Peak Error**: Udáva najväčšiu chybu medzi predikciou a reálnymi hodnotami, dôležitá pre analýzu kritických chýb.

Tieto metriky poskytujú jasný pohľad na presnosť a efektivitu vyvinutého algoritmu v rámci tejto práce. Pri použití na reálnych dátach poskytujú objektívne a konzistentné hodnotenie výkonu modelu.

## 2.3 Analýza realizácií GRU modelov

Architektúra GRU (3.2) môže byť implementovaná pomocou rôznych knižníc strojového učenia, najpopulárnejšie sú TensorFlow a PyTorch. TensorFlow poskytuje širokú podporu, veľkú komunitu a množstvo integrovaných nástrojov, avšak PyTorch je častejšie preferovaný pre výskumné účely vdaka svojej jednoduchosti, flexibilite a dynamickému výpočtovému grafu.

Pre implementáciu bol zvolený práve PyTorch, pretože umožňuje jednoduché ladenie, poskytuje lepšiu transparentnosť pri experimentovaní a jednoduchšie nasadenie modelov v prostredí Jupyter notebook, ktoré bolo použité na interaktívnu analýzu a vizualizáciu výsledkov experimentov. Výber PyTorch bol strategickým rozhodnutím s cieľom maximalizovať produktivitu a efektivitu procesu vývoja.

## 3 Experimenty

### 3.1 Dátová množina (Dataset) [1]

Táto sekcia sa venuje popisu dátovej množiny použitej pre experimenty, procesu hľadania vhodnej dátovej množiny, realizácie exploratórnej dátovej analýzy (EDA), predspracovaniu dát a vytvoreniu univerzálneho dátového načítača.

#### 3.1.1 Nájdenie dátovej množiny

Pre realizáciu experimentov bola vybraná reálna dátová množina z oblasti prenájmu bicyklov v Londýne. Dáta sú dostupné verejne a poskytujú dostatočný objem a rôznorodosť pre overenie navrhnutých metód. Dataset obsahuje záznamy o prenájmoch bicyklov, ktoré pokrývajú obdobie od 4. januára 2015 do 3. januára 2017, s celkovou dĺžkou 730 dní.

#### 3.1.2 Realizácia EDA

Exploratórna dátová analýza (EDA) bola realizovaná pomocou Python notebooku. Cieľom EDA bolo podrobne preskúmať charakteristiky dát, ich distribúciu a identifikovať potenciálne problémy alebo chyby v dátach.

Dataset má nasledujúcu štruktúru:

- Veľkosť datasetu: 17 414 záznamov, 10 atribútov
- Časový rozsah dát: od 2015-01-04 00:00:00 do 2017-01-03 23:00:00
- Celková dĺžka sledovaného obdobia: 730 dní

Nasledujúce tabuľky poskytujú detailný pohľad na charakter dát:

**Tabuľka 3 – 1:** Niekolko prvých riadkov datasetu

Nº	timestamp	cnt	t1	t2	hum	wind	weather	holiday	weekend	season
1	2015-01-04 00:00:00	182	3.0	2.0	93	6.0	3	0	1	3
2	2015-01-04 01:00:00	138	3.0	2.5	93	5.0	1	0	1	3
3	2015-01-04 02:00:00	134	2.5	2.5	96	0.0	1	0	1	3
4	2015-01-04 03:00:00	72	2.0	2.0	96	0.0	1	0	1	3
5	2015-01-04 04:00:00	47	2.0	0.0	93	6.5	1	0	1	3

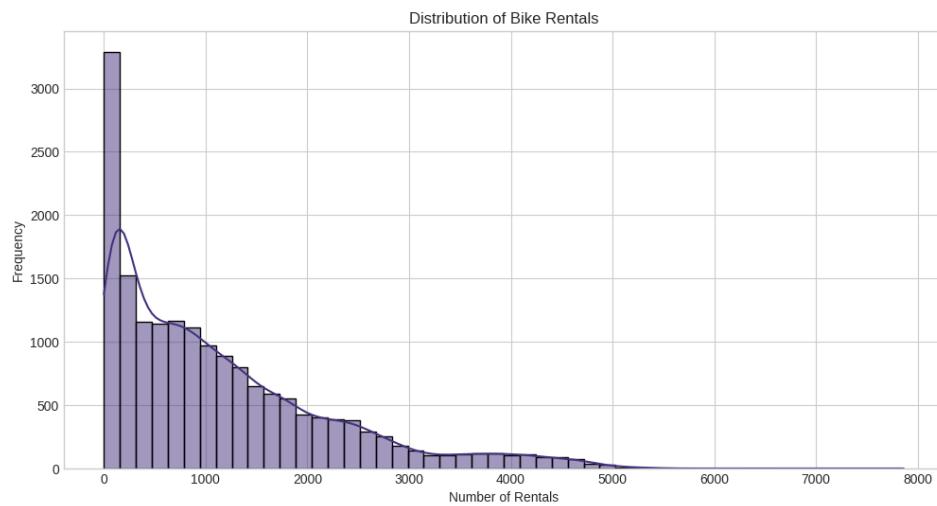
**Tabuľka 3 – 2:** Prehľad typov dát v datasete

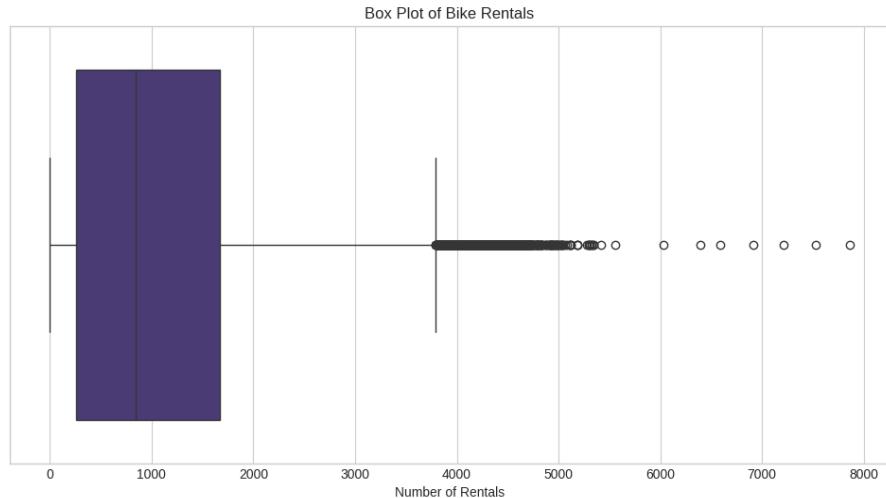
Atribút	Typ dát
timestamp	object
cnt	int64
t1	float64
t2	float64
hum	float64
wind_speed	float64
weather_code	float64
is_holiday	float64
is_weekend	float64
season	float64

**Tabuľka 3–3:** Štatistické zhrnutie datasetu

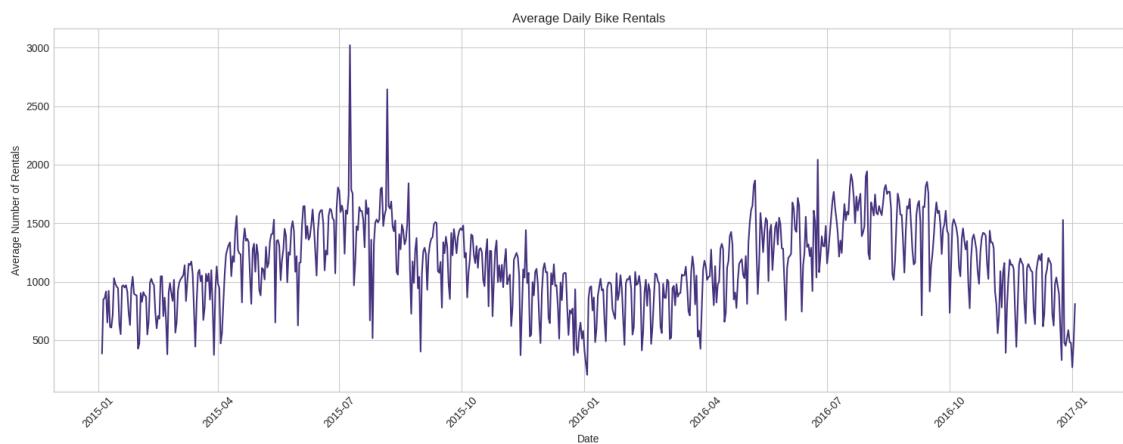
Štatistika	cnt	t1	t2	hum	wind	weather	holiday	weekend	season
count	17414	17414	17414	17414	17414	17414	17414	17414	17414
mean	1143.10	12.47	11.52	72.32	15.91	2.72	0.02	0.29	1.49
std	1085.11	5.57	6.62	14.31	7.89	2.34	0.15	0.45	1.12
min	0.00	-1.50	-6.00	20.50	0.00	1.00	0.00	0.00	0.00
25%	257.00	8.00	6.00	63.00	10.00	1.00	0.00	0.00	0.00
50%	844.00	12.50	12.50	74.50	15.00	2.00	0.00	0.00	1.00
75%	1671.75	16.00	16.00	83.00	20.50	3.00	0.00	1.00	2.00
max	7860.00	34.00	34.00	100.00	56.50	26.00	1.00	1.00	3.00

Grafické analýzy zahŕňali nasledujúce:

**Obrázok 3–1:** Distribution of Bike Rentals



**Obrázok 3 – 2:** Box Plot of Bike Rentals



**Obrázok 3 – 3:** Average Daily Bike Rentals

### 3.1.3 Predspracovanie a unifikácia dát

Počas analýzy bolo identifikovaných niekoľko chýb a nekonzistentností v dátach, ktoré boli vyriešené predspracovaním a unifikáciou dát. Toto zahŕňalo:

- Riešenie chýbajúcich hodnôt a ich doplnenie alebo odstránenie
- Normalizáciu a škálovanie dátových atribútov

- Konverziu dátových typov a odstránenie redundantných dát

### 3.1.4 Vytvorenie univerzálneho dátového načítača

Pre zabezpečenie flexibility a všeobecnej použiteľnosti bol vytvorený univerzálny dátový načítač, ktorý umožňuje jednoducho načítať a spracovať ľubovoľný dataset bez nutnosti výraznej úpravy kódu. Tento načítač bol implementovaný v Python súbore pomocou knižníc pandas a numpy, čím je zabezpečená kompatibilita s rôznymi formátmi a štruktúrami dát.

## 3.2 Implementácia flexibilného GRU modelu

V rámci tejto práce bol implementovaný flexibilný GRU model, umožňujúci efektívne spracovanie sekvenčných dát a adaptáciu na rôzne typy úloh a dátových množín. Implementácia bola realizovaná pomocou PyTorch [3], pričom bola zabezpečená vysoká flexibilita modelu.

Kľúčové vlastnosti modelu zahŕňajú:

- Podpora jednej alebo viacerých vrstiev GRU.
- Možnosť konfigurácie veľkostí skrytých vrstiev.
- Použitie dropout vrstiev medzi GRU vrstvami na prevenciu pretrénovania.
- Možnosť volby jednosmernej alebo obojsmernej GRU architektúry.
- Podpora návratu výstupov za celú sekvenciu alebo len z posledného kroku.

Implementovaný model je navrhnutý na základe jasne štruktúrovaného a dobre komentovaného Python kódu, čo umožňuje jednoduchú údržbu a budúce rozšírenia alebo modifikácie. Model bol navrhnutý s dôrazom na všeobecnú použiteľnosť a vysokú výpočtovú efektívnosť.

### 3.3 Realizácia metrík

Pre vyhodnotenie modelov boli implementované nasledujúce metriky:

**Mean Absolute Error (MAE):**

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Kde  $y_i$  je skutočná hodnota a  $\hat{y}_i$  je predikovaná hodnota.

**Mean Squared Error (MSE):**

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

**Root Mean Squared Error (RMSE):**

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

**R-squared (Coefficient of Determination):**

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Kde  $\bar{y}$  je priemerná hodnota skutočných výstupov.

**Mean Absolute Percentage Error (MAPE):**

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

**Symmetric Mean Absolute Percentage Error (sMAPE):**

$$SMAPE = \frac{100\%}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|) / 2}$$

**Explained Variance (EV):**

$$EV = 1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)}$$

Kde  $\text{Var}(\cdot)$  označuje štatistickú varianciu.

**Peak Error (Maximum Absolute Error):**

$$PE = \max_i |y_i - \hat{y}_i|$$

Tieto metriky boli implementované pomocou knižníc numpy a pandas, čo zabezpečuje transparentnosť a kontrolovateľnosť procesu výpočtov.

## 3.4 Priebeh experimentov

Táto kapitola opisuje proces vykonávania experimentov, ktoré zahŕňali hľadanie a optimalizáciu hyperparametrov pre GRU model pomocou metód Random Search, Bayesian Search a následne Grid Search. Cieľom bolo identifikovať najlepšiu kombináciu hyperparametrov pre čo najvyššiu výkonnosť modelu.

### 3.4.1 Počiatočná sieť hyperparametrov

Pre experimenty bola vytvorená nasledujúca počiatočná siet hyperparametrov:

**Tabuľka 3 – 4:** Počiatočná siet hyperparametrov

Hyperparameter	Hodnoty
Hidden size	32, 64, 128, 256, 512
Počet vrstiev	1, 2, 3, 4, 5
Dropout	0.0, 0.1, 0.2, 0.3, 0.4, 0.5
Learning rate	0.01, 0.001, 0.0001, 0.00001
Weight decay	0.0, 0.0001, 0.001, 0.00001
Bidirectional	False, True
Early stopping patience	5, 10, 15
Learning rate factor	0.5, 0.25, 0.1
Learning rate patience	5, 10

Táto široká siet parametrov umožnila dôkladné preskúmanie možností a určenie, ktoré hyperparametre majú najvýznamnejší vplyv na výkonnosť modelu. Rozmanitosť nastavení bola zvolená zámerne, aby boli pokryté extrémne aj stredné hodnoty hyperparametrov a ich interakcie.

Podrobny log výsledkov je verejne dostupný v príslušnom notebooku `experiments.ipynb` na GitHub [2] repozitári projektu.

### 3.4.2 Výsledky Random Search

Random Search dosiahol nasledovné najlepšie výsledky:

**Tabuľka 3 – 5:** Výsledky Random Search

Hyperparameter	Najlepšia hodnota
Validation RMSE	0.054382
Hidden size	512
Počet vrstiev	1
Dropout	0.0
Learning rate	0.0001
Weight decay	0.0001
Bidirectional	True
Early stopping patience	15
Learning rate factor	0.5
Learning rate patience	10

Random Search umožnil efektívne preskúmať náhodne zvolené kombinácie, čím sa zvýšila pravdepodobnosť nájdenia dobre fungujúcich kombinácií, ktoré by grid search [4] mohol prehliadnuť. Táto metóda poskytla dobré východiskové body pre ďalšie optimalizačné kroky.

### 3.4.3 Výsledky Bayesian Search

Bayesian Search dosiahol mierne lepší výsledok:

**Tabuľka 3 – 6:** Výsledky Bayesian Search

Hyperparameter	Najlepšia hodnota
Validation RMSE	0.054269
Hidden size	128
Počet vrstiev	2
Dropout	0.1
Learning rate	0.01
Weight decay	0.0
Bidirectional	True
Early stopping patience	15
Learning rate factor	0.1
Learning rate patience	10

Bayesian Search bol efektívnejší v hľadaní optimálnych hyperparametrov vďaka svojmu adaptívному prístupu, ktorý využíva predchádzajúce výsledky na inteligenčnejšie nasmerovanie ďalších experimentov. Výsledkom bola rýchlejšia konvergencia k optimálnym riešeniam.

### 3.4.4 Analýza najlepších sád hyperparametrov

Z kombinovaných výsledkov Random a Bayesian Search boli vybrané tri najlepšie sady hyperparametrov:

**Tabuľka 3 – 7:** Najlepšie sady hyperparametrov z Random a Bayesian Search

Hyperparameter	Sada 1	Sada 2	Sada 3
Hidden size	128	512	64
Počet vrstiev	2	1	1
Dropout	0.1	0.0	0.0
Learning rate	0.01	0.0001	0.01
Weight decay	0.0	0.0001	1e-05
Bidirectional	True	True	False
Early stopping patience	15	15	10
Learning rate factor	0.1	0.5	0.5
Learning rate patience	10	10	10
RMSE	0.054269	0.054382	0.054632

Tieto tri najlepšie kombinácie hyperparametrov boli vybrané na základe dosiahnutých hodnôt RMSE 3.3 a slúžili ako východiskový bod pre detailnejšiu optimalizáciu pomocou Grid Search 3.4.5. Ich výber reflektuje rôznorodosť úspešných konfigurácií, čo umožňuje hlbšie pochopenie ich vplyvu na model.

### 3.4.5 Grid Search optimalizácia

Z dôvodu obmedzených výpočtových kapacít pre spracovanie veľkého množstva kombinácií sme na základe predchádzajúcej analýzy 3 – 7 vytvorili finálnu optimalizačnú sieť pre Grid Search:

**Tabuľka 3 – 8:** Grid Search optimalizačná sieť

Hyperparameter	Hodnoty
Hidden size	64, 128, 256
Počet vrstiev	2, 3
Dropout	0.0, 0.3
Learning rate	0.001, 0.0001, 0.00001
Weight decay	0.0001, 0.00001
Bidirectional	True
Early stopping patience	10
Learning rate factor	0.25
Learning rate patience	5, 10

Táto zúžená optimalizačná sieť bola navrhnutá na základe predchádzajúcich výsledkov, aby sa minimalizoval počet testovaných kombinácií a zároveň sa maximalizovala efektivita výpočtových zdrojov. Grid Search týmto umožnil dôkladné preskúmanie najperspektívnejších oblastí parametrového priestoru.

Po vykonaní viac ako 500 rôznych kombinácií Grid Search bola nájdená najlepšia sada parametrov:

**Tabuľka 3 – 9:** Najlepšie hyperparametre z Grid Search

Hyperparameter	Najlepšia hodnota
Hidden size	64
Počet vrstiev	2
Dropout	0.0
Learning rate	0.01
Weight decay	1e-05
Bidirectional	False
Early stopping patience	15
Learning rate factor	0.1
Learning rate patience	10
Najlepšie RMSE	0.052874

Táto optimálna sada parametrov bola následne použitá na tréning finálnej verzie modelu, čím bol úspešne ukončený optimalizačný proces.

## 4 Analýza a vizualizácia výsledkov

Táto kapitola sa venuje detailnej analýze a interpretácii výsledkov získaných počas optimalizácie hyperparametrov a tréningu modelu pomocou metód Random Search 3.4.2, Bayesian Search 3.4.3, Grid Search 3.4.5 a následnej finálnej optimalizácie modelu.

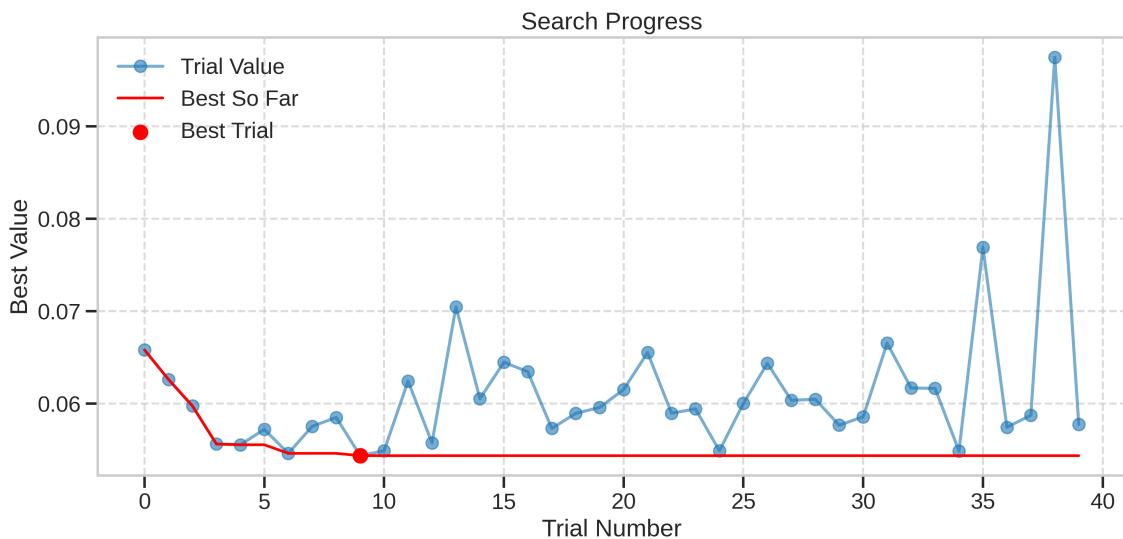
### 4.1 Vizualizácia výsledkov Random Search

Random Search umožnil rýchle preskúmanie veľkého priestoru hyperparametrov. Výsledky z Random Search boli analyzované pomocou viacerých grafických vizu-

alizácií, ktoré poskytujú prehľad o výkonnosti modelu v závislosti od použitých hyperparametrov.

#### 4.1.1 Progres vyhľadávania

Graf progresu vyhľadávania ukazuje, ako sa menila najlepšia hodnota RMSE počas jednotlivých pokusov. Tento graf poukazuje na stabilizáciu výsledkov okolo najlepšej hodnoty po niekoľkých počiatočných pokusoch.

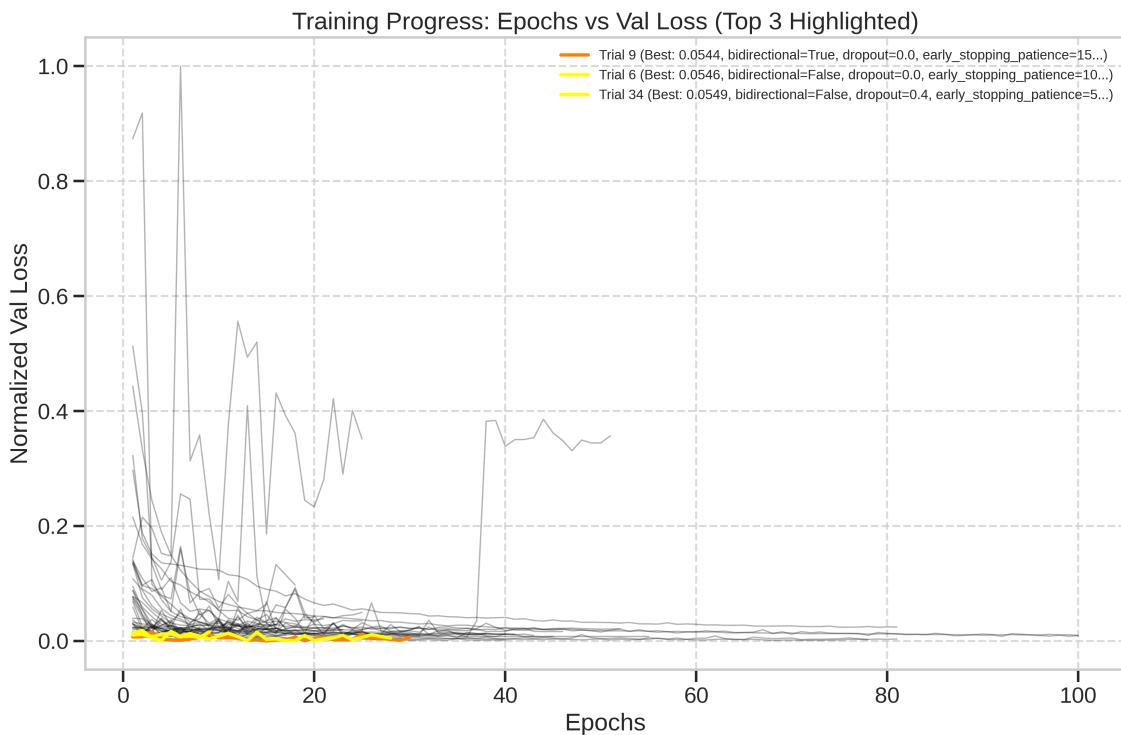


**Obrázok 4–1:** Progres vyhľadávania

Analýza grafu odhaluje, že v počiatočnej fáze experimentov bolo pozorované lineárne klesanie hodnoty RMSE, čo indikuje systematické zlepšovanie výkonnosti modelu. Avšak po approximativne 10 experimentoch dochádza k výraznej volatilite v hodnotách RMSE. Pozoruhodné je, že optimálna hodnota RMSE bola dosiahnutá už pri deviatom experimente. Následná vysoká volatilita hodnôt naznačuje, že ďalšie testované kombinácie hyperparametrov boli menej efektívne, čo potvrdzuje, že algoritmus random search úspešne identifikoval oblasť optimálneho nastavenia v relatívne skorej fáze experimentovania.

#### 4.1.2 Vzťah medzi epochami a stratou (Loss)

Graf Epochs vs Loss zobrazuje priebeh trénovacej a validačnej chyby počas tréningových epoch. Tento graf nám umožňuje sledovať, ako rýchlo a efektívne model konverguje.

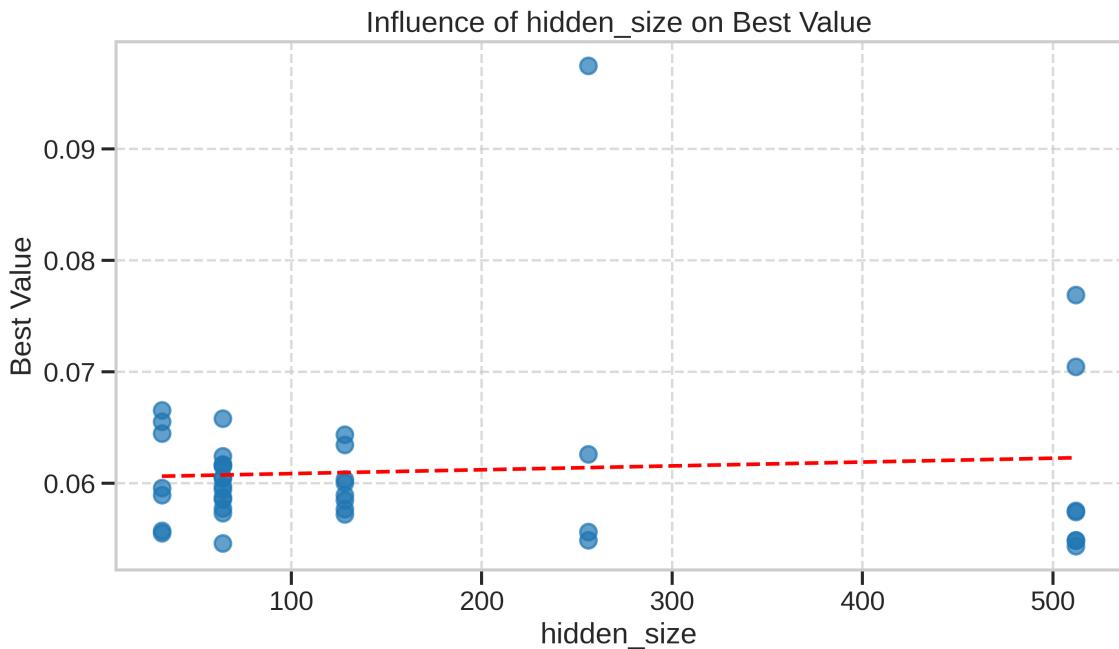


**Obrázok 4 – 2:** Epochs vs Loss

Pozorujeme významnú variabilitu v procesoch učenia, ako aj vysokú volatilitu medzi rôznymi kombináciami hyperparametrov. Najmenej efektívne kombinácie vyzkazujú extrémnu volatilitu a vo väčšine prípadov vysokú hodnotu funkcie straty. Väčšina kombinácií demonštruje pomalú schopnosť učenia, pri ktorej funkcia straty klesá lineárne, čo nie je optimálne z hľadiska efektivity. Najlepšie kombinácie vyzkazujú nízku hodnotu funkcie straty, rýchlo ukončujú proces učenia na zabranenie pretrénovaniu, pričom rozdiel v hodnotách funkcie straty sa prejavuje až na tretom desatinnom mieste.

#### 4.1.3 Vplyv veľkosti skrytých vrstiev (Hidden size)

Analýza vplyvu veľkosti skrytej vrstvy ukazuje, ako tento parameter ovplyvňuje celkovú chybu modelu. Z grafu je možné vidieť, že optimálne výsledky sa pohybovali okolo stredných až vyšších hodnôt veľkosti skrytých vrstiev.

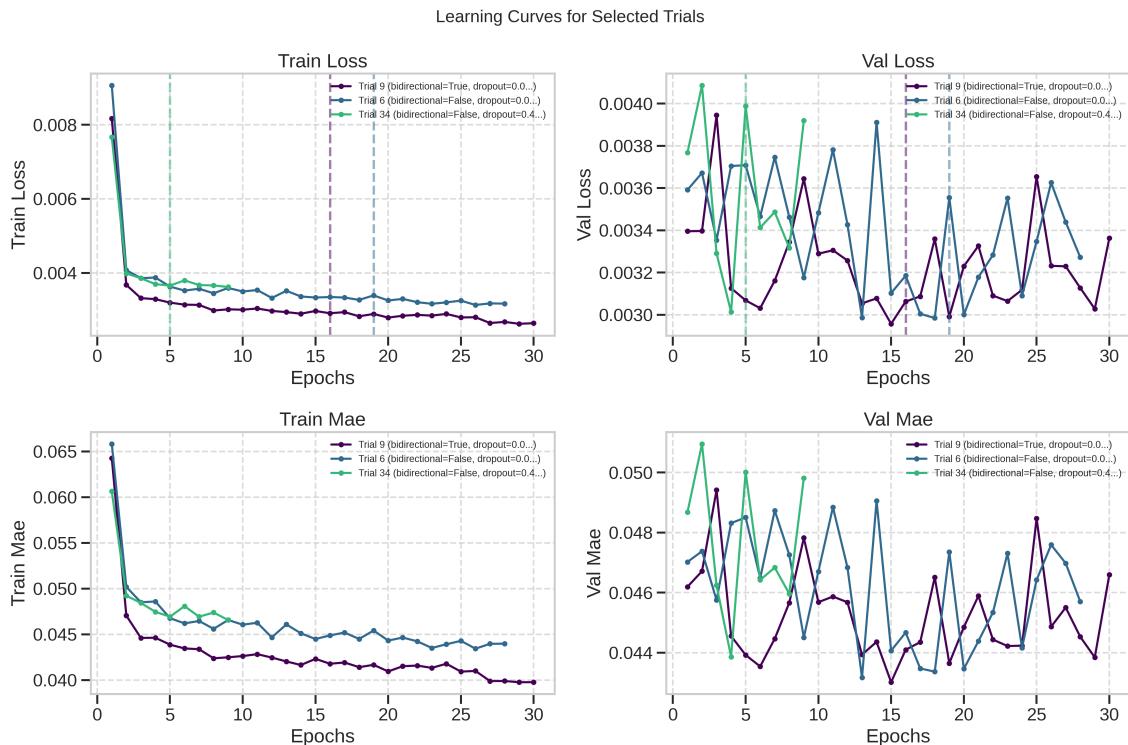


Obrázok 4–3: Hidden size influence

Vplyv veľkosti skrytých blokov sa vzájomne výrazne nelísi, s výnimkou rýchlosťi učenia. Čím väčšia je veľkosť skrytých blokov, tým dlhšie trvá tréning modelu. Napriek hlbšej architektúre výsledky ukazujú, že menšia veľkosť skrytých blokov vyzkazuje menší rozptyl a stabilnejšie výsledky. Zároveň takmer každá veľkosť skrytých blokov má kombinácie, ktoré demonštrujú takmer nerozlíšiteľné hodnoty funkcie straty.

#### 4.1.4 Krivky učenia (Learning curves)

Krivky učenia demonštrujú vývoj chyby (loss) a metriky MAE počas tréningu a validácie. Táto analýza poskytuje pohľad na schopnosť modelu generalizovať na nové dátá.

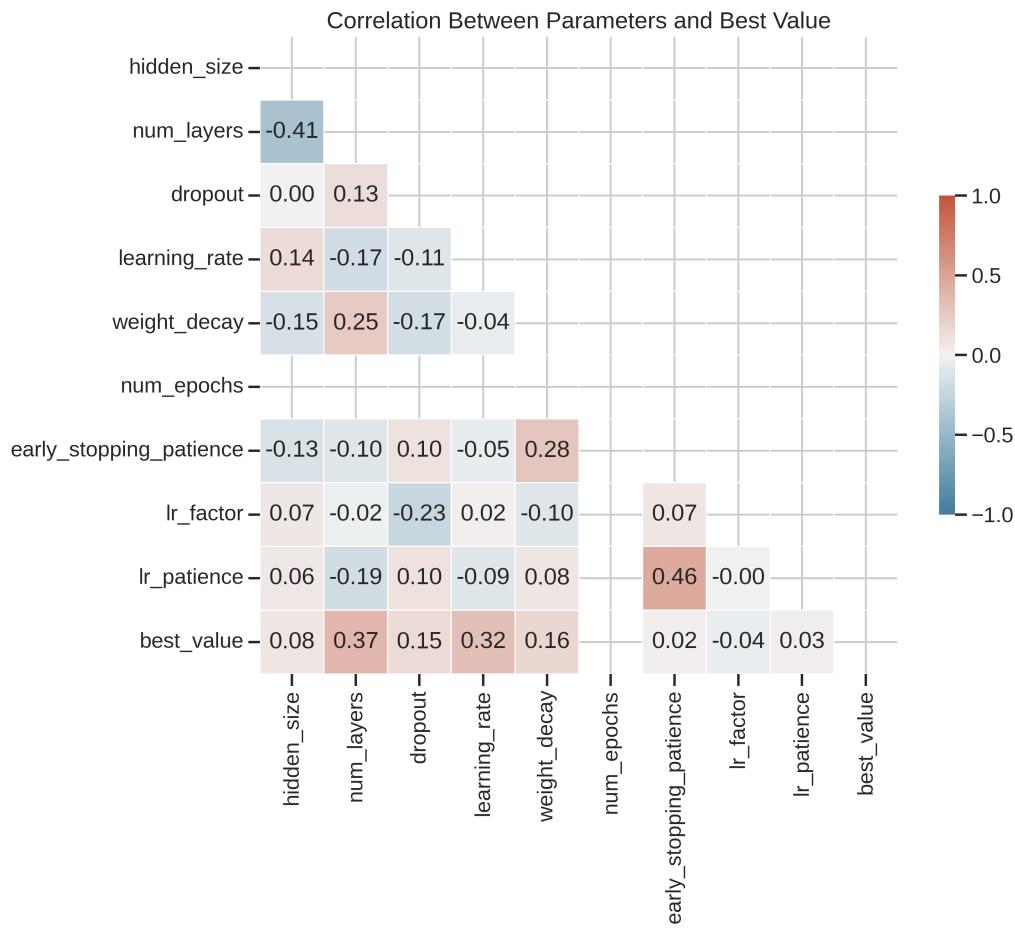


Obrázok 4 – 4: Learning curves

Pozorujeme očakávané lineárne zníženie funkcie straty počas trénovania pre 3 najlepšie modely. Počas validačnej fázy metriky vykazujú porovnatelný pokles, avšak s vyššou volatilitou. Napriek tomu každý z modelov dosahuje presvedčivé pozitívne výsledky. Vysoká volatilita bez výrazného zníženia funkcie straty môže signalizovať pretrénovanie modelov.

#### 4.1.5 Korelácia medzi parametrami a výkonom

Graf korelácie medzi parametrami a výkonom znázorňuje, ako jednotlivé parametre ovplyvňujú výslednú hodnotu RMSE. Z tohto grafu je možné identifikovať parametre, ktoré majú najväčší vplyv na výkon modelu.



**Obrázok 4 – 5:** Parameter correlation

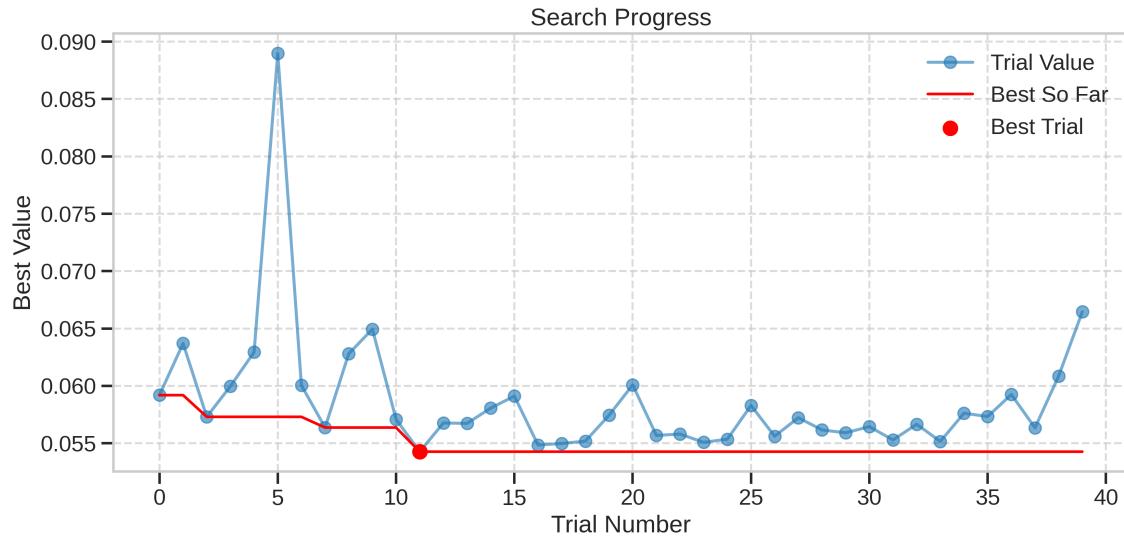
Korelačná matica nám názorne demonštruje, že na výsledky najväčším spôsobom vplývali nasledujúce parametre: počet skrytých vrstiev a počiatočná hodnota learning rate. Zároveň pozorujeme priamy vzájomný vplyv early stopping patience na lr patience, čo môže signalizovať, že learning rate sa nestihol znížiť pre ďalšie trénovanie modelu, kvôli čomu boli tieto kombinácie označené ako neefektívne z dôvodu nedostatočného tréningu.

## 4.2 Vizualizácia výsledkov Bayesian Search

Bayesian Search je efektívna optimalizačná metóda, ktorá umožňuje inteligentné hľadanie hyperparametrov s využitím informácií z predchádzajúcich pokusov. Táto metóda rýchlo konvergovala k optimálnym výsledkom, čo potvrdzuje aj graf progresu vyhľadávania.

### 4.2.1 Progres vyhľadávania (Search Progress)

Graf progresu vyhľadávania ukazuje, ako sa menila najlepšia hodnota RMSE počas jednotlivých pokusov. Tento graf poukazuje na stabilizáciu výsledkov okolo najlepšej hodnoty po niekoľkých počiatocných pokusoch.



Obrázok 4 – 6: Search progress

Na grafe je možné sledovať, ako rýchlo Bayesian Search identifikoval najlepšiu hodnotu RMSE, ktorá sa stabilizovala už po niekoľkých pokusoch. V porovnaní s progresom vyhľadávania algoritmu Random Search pozorujeme prakticky opačnú situáciu, hoci najefektívnejšia kombinácia bola nájdená v bezprostrednej blízkosti výsledkov Random Search. Na začiatku možno vidieť vysokú volatilitu neefektívnych kombinácií, ale vďaka tomu, že Bayesian Search adaptívne hľadá kombinácie

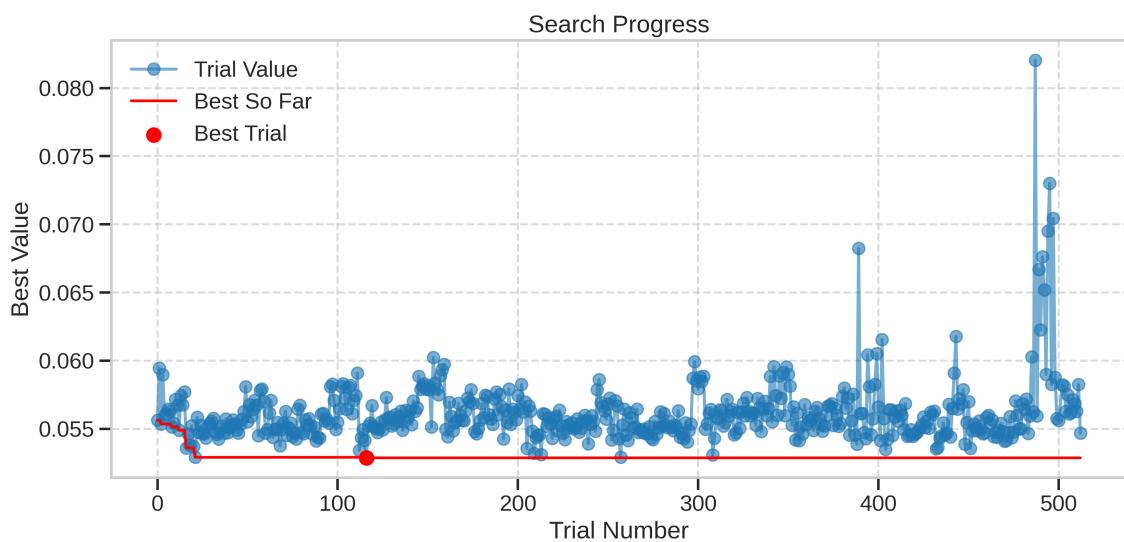
na základe predchádzajúcich výsledkov, a ako vidno na grafe, pri ďalšom hľadaní kombinácií sa výsledky nevyznačujú volatilitou, možno s istotou tvrdiť, že Bayesian Search má vyššiu efektívnosť v porovnaní s predchádzajúcim algoritmom. V prípade Random Search by sa dalo povedať, že mu "prialo šťastie" pri nájdení efektívnej kombinácie.

### 4.3 Vizualizácia výsledkov Grid Search

Grid Search bol záverečnou optimalizačnou fázou, v ktorej sme sa zamerali na dôkladné preskúmanie najperspektívnejších kombinácií hyperparametrov identifikovaných v predošlých experimentoch. Vďaka predchádzajúcej selekcii sme mohli obmedziť počet kombinácií a tým výrazne zvýšiť výpočtovú efektivitu bez kompromisu na kvalite výsledkov.

#### 4.3.1 Progres vyhľadávania (Search Progress)

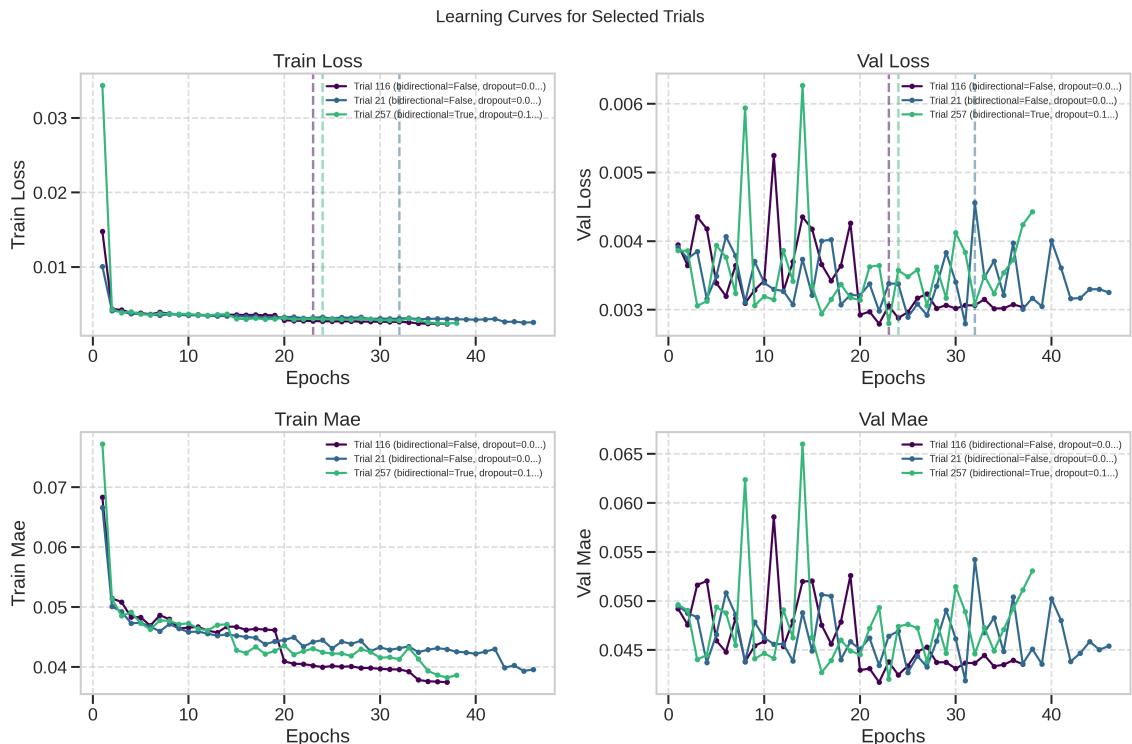
Graf vyhľadávania ukazuje priebeh znižovania hodnoty validačnej chyby RMSE. Aj napriek veľkému počtu testovaných konfigurácií bola najlepšia konfigurácia nájdená v relatívne skorých iteráciách.



Obrázok 4 – 7: Search progress

### 4.3.2 Krivky učenia (Learning Curves)

Porovnanie vývoja loss a MAE pre top 3 pokusy počas tréningu a validácie odhaluje stabilné učenie modelu s dobrou schopnosťou generalizácie. Z grafov vyplýva, že overfitting bol minimálny vďaka použitým technikám ako dropout a early stopping.

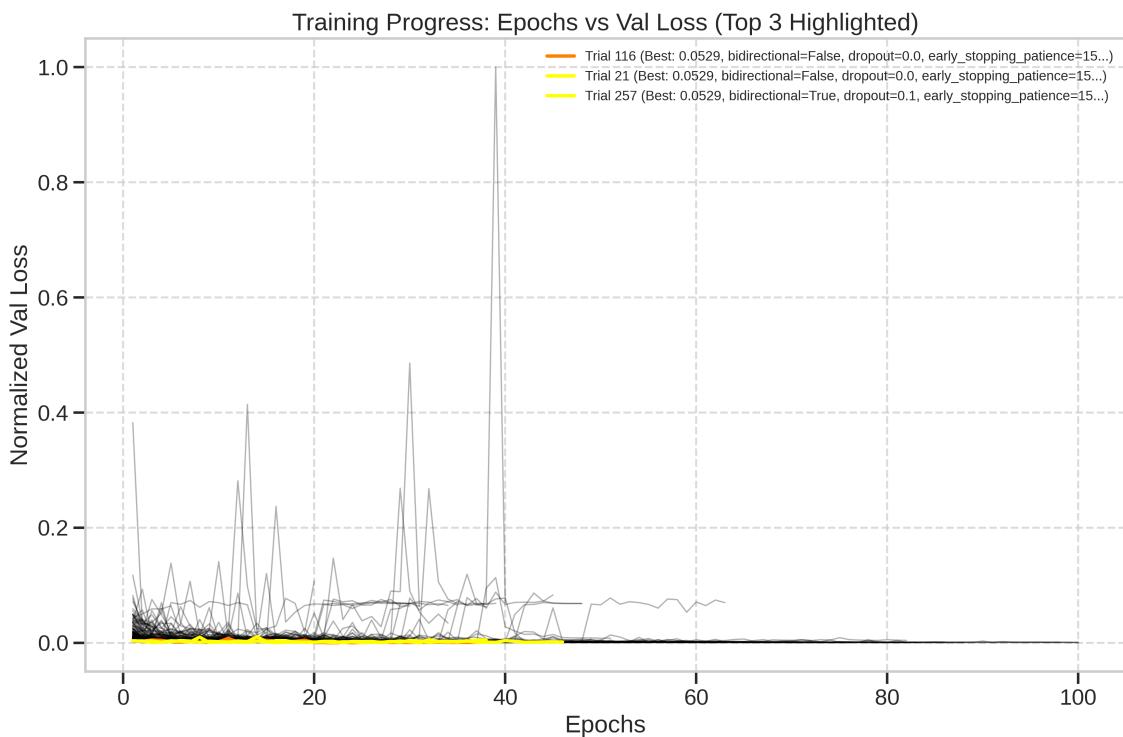


Obrázok 4 – 8: Learning curves

Výsledky sú veľmi podobné výsledkom predchádzajúcich vyhľadávaní. V tomto jedinečnom prípade najlepšie 3 kombinácie preukázali dobrú schopnosť generalizácie a prakticky lineárnu tendenciu znižovania funkcie straty pri validácii zodpovedajúcu trénovaniu, a to napriek štandardnej volatilite na začiatku.

#### 4.3.3 Vzťah medzi epochami a stratou (Epochs vs Loss)

Tento graf poskytuje informácie o dynamike učenia počas jednotlivých epoch. Modely dosahovali veľmi nízke chyby už v prvých 20 epochách, pričom ďalšie zlepšovanie bolo pozvolné.

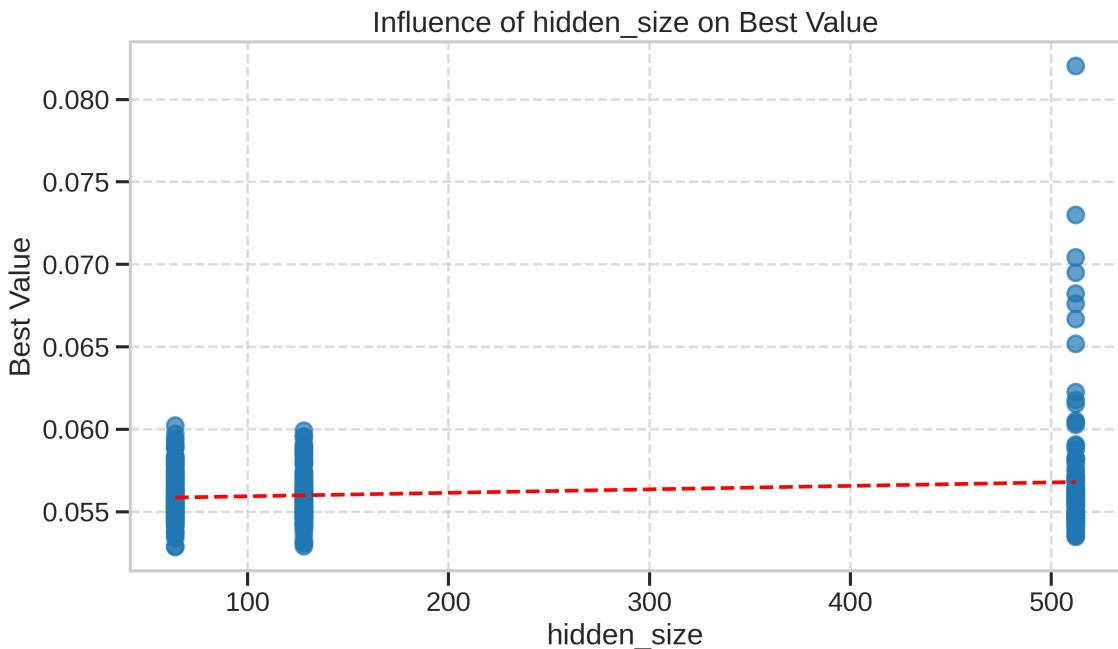


**Obrázok 4 – 9:** Epochs vs Loss

Zúženie siete hyperparametrov poslúžilo nielen ako dobrá praktika pre zrýchlenie vyhľadávania, ale sa tiež odrazilo na výsledkoch. Absolútna väčšina kombinácií preukázala vynikajúcnu konvergenciu, lineárne klesala a v najlepších prípadoch vykazovala vysokú efektivitu počas prvej polovice epoch. Samozrejme, existujú aj ojedinelé prípady vykazujúce enormnú volatilitu a neefektívnosť. Najlepšie 3 kombinácie sa navzájom líšia hodnotami funkcie straty menej než o 4 desatinné miesta, čo demonštruje vynikajúci prieskum oblasti efektívnych hyperparametrov.

#### 4.3.4 Vplyv skrytej vrstvy (Hidden Size Influence)

Z grafu je zrejmé, že vyššie hodnoty skrytých vrstiev (napr. 512) neznamenali automaticky lepsiú výkonnosť. Optimálne výsledky boli dosiahnuté aj pri nižších hodnotách ako 64 alebo 128, čo poukazuje na potrebu citlivej optimalizácie tohto parametra.

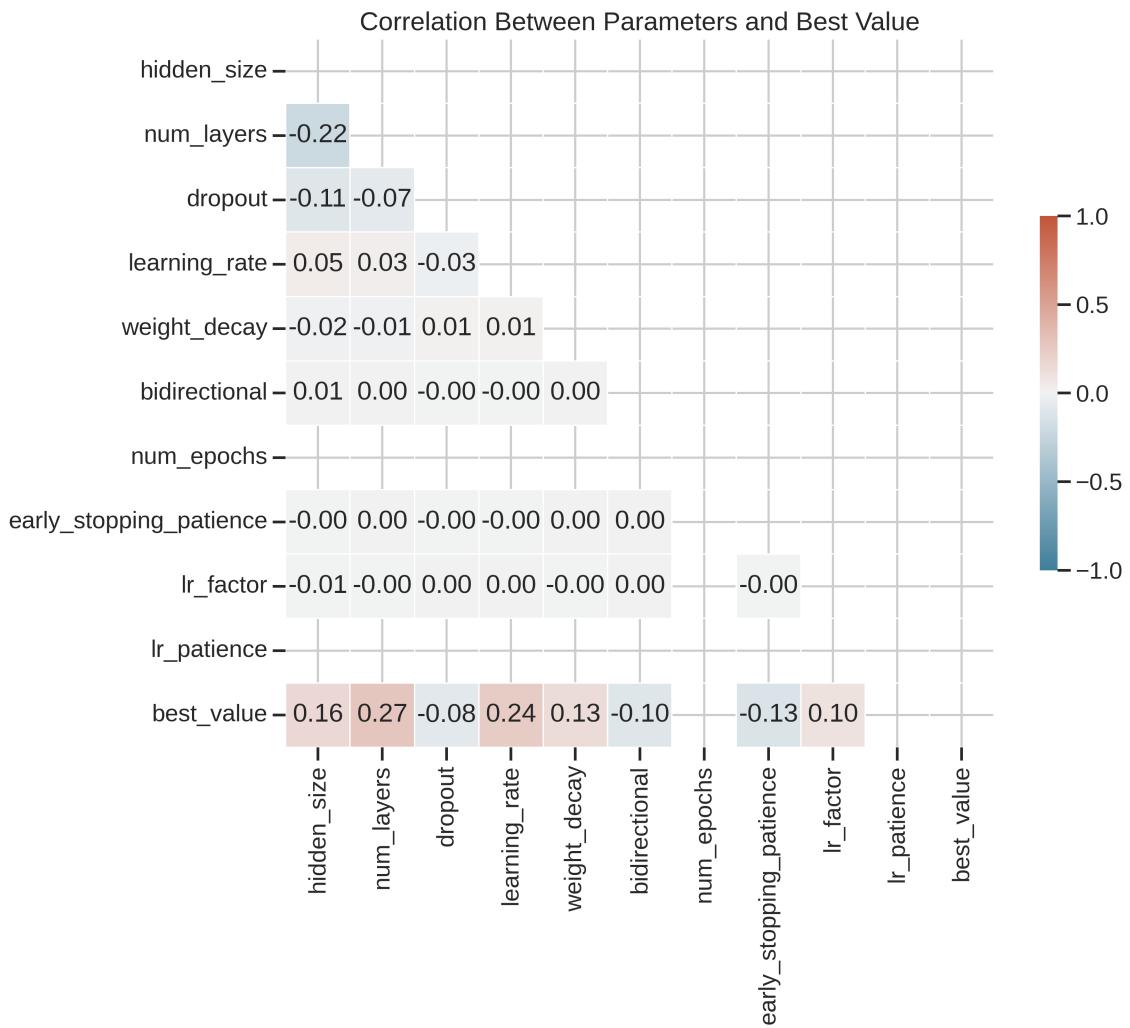


Obrázok 4–10: Hidden size influence

Výsledok je veľmi podobný predchádzajúcim algoritmom vyhľadávania, pričom sa zachovala tendencia od väčšej veľkosti blokov k menšej. Zároveň názorne vidíme, že hodnoty pre väčšie veľkosti blokov vykazujú veľký rozptyl a volatilitu v porovnaní s menšími veľkosťami.

#### 4.3.5 Korelácia parametrov s výkonom (Parameter Correlation)

Korelácia medzi jednotlivými hyperparametrami a výkonom modelu ukazuje, že najväčší negatívny vplyv mal počet vrstiev, zatiaľ čo hidden size a learning rate boli mierne pozitívne korelované s lepsou výkonnosťou. Tieto zistenia podporujú výber jednoduchšej, ale dobre nastavenej architektúry.



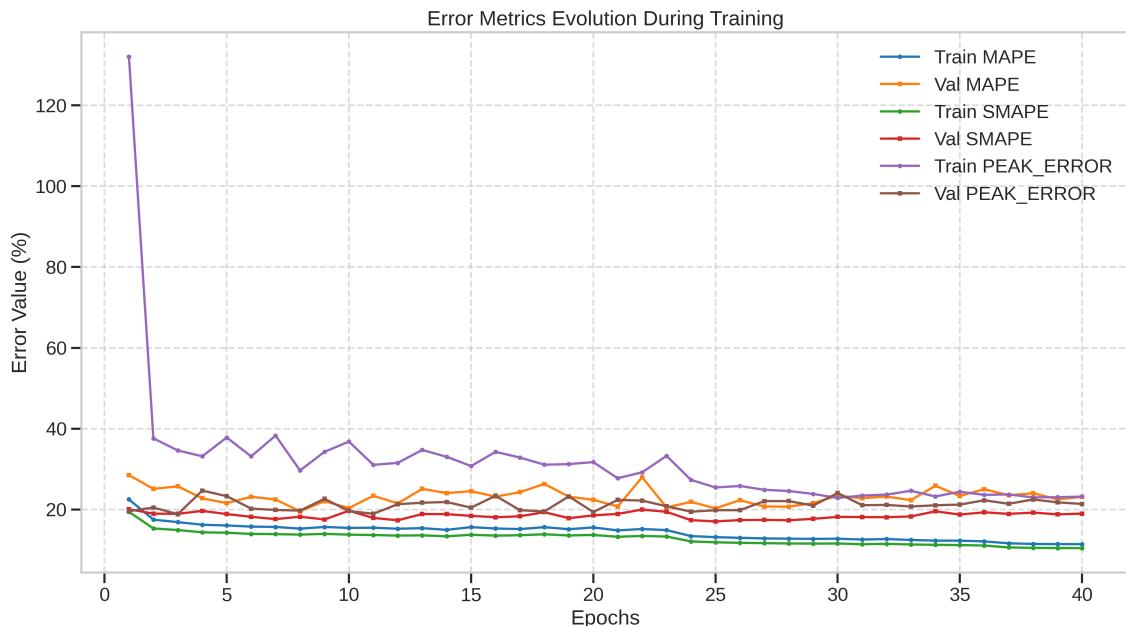
Obrázok 4–11: Parameter correlation

## 4.4 Vizualizácia výsledkov modelu natrénovaného na najlepších parametroch

Po ukončení optimalizácie pomocou Grid Search a identifikácii najlepšej kombinácie hyperparametrov bola natrénovaná finálna GRU modelová architektúra. Tento model bol detailne analyzovaný z pohľadu vývoja učiacich kriviek, dynamiky učiacej rýchlosťi, ako aj hodnotení širokého spektra metrických indikátorov.

### 4.4.1 Evolúcia chýb a dynamika tréningu

Vizualizácia **Error Metrics Evolution** ilustruje, ako sa vyvíjali chyby typu MAPE 3.3, SMAPE 3.3 a PEAK\_ERROR 3.3 v priebehu tréningu a validácie. Po počiatokom prudkom poklese sa metriky ustálili a dosiahli stabilné hodnoty, čo naznačuje dobrú generalizáciu modelu.

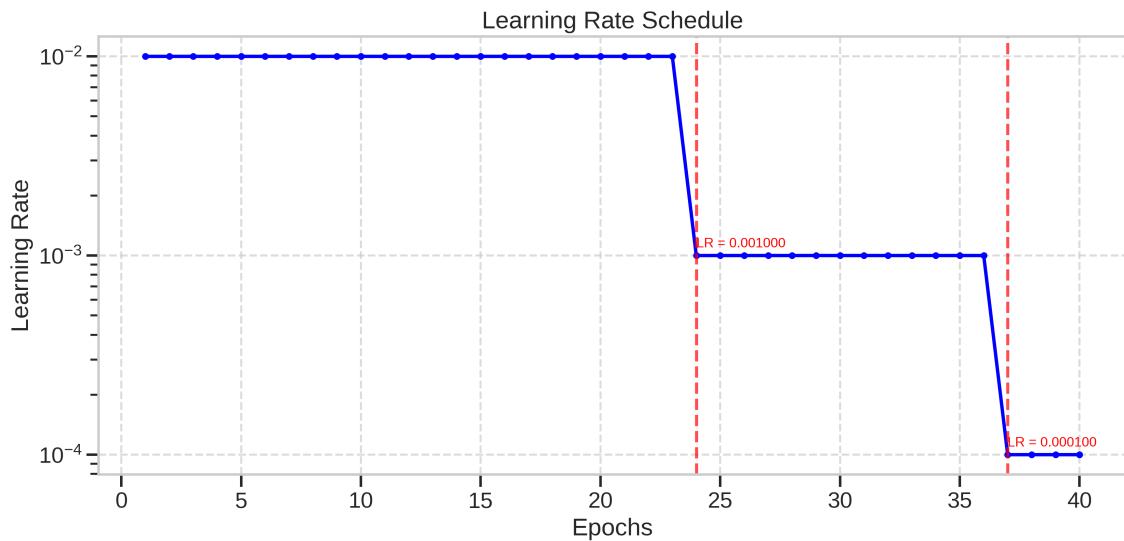


Obrázok 4–12: Best model error metrics evolution

Pri analýze výsledkov trénovania modelu s optimálnymi hyperparametrami bolo zistené, že dosiahnutá hodnota funkcie chyby bola nižšia v porovnaní s rovnakými hyperparametrami použitými počas fázy vyhľadávania.

#### 4.4.2 Dynamické prispôsobovanie učiacej rýchlosťi

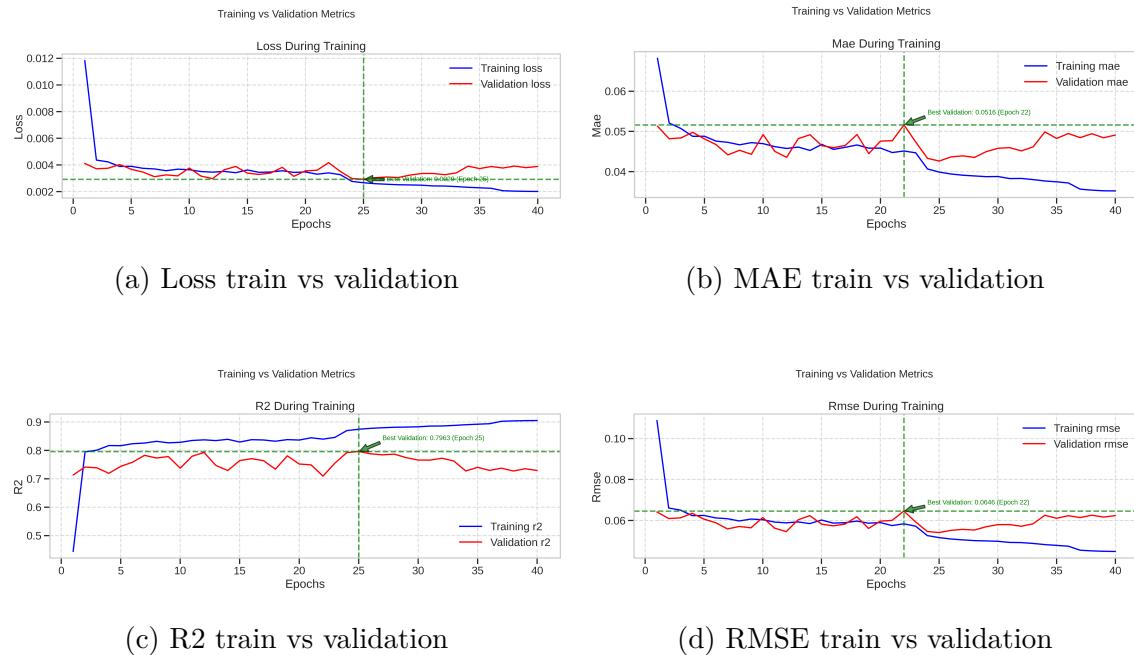
Graf **Learning Rate Schedule** zobrazuje zmenu učiacej rýchlosťi v priebehu tréningu. Použitie schémy adaptívneho zmenšovania učiacej rýchlosťi (learning rate scheduler) po určitom počte epoch pomohlo modelu vyhnúť sa preškoleniu a zároveň optimálizovať konvergenciu.



Obrázok 4–13: Best model learning rate schedule

#### 4.4.3 Porovnanie tréningových a validačných metrík

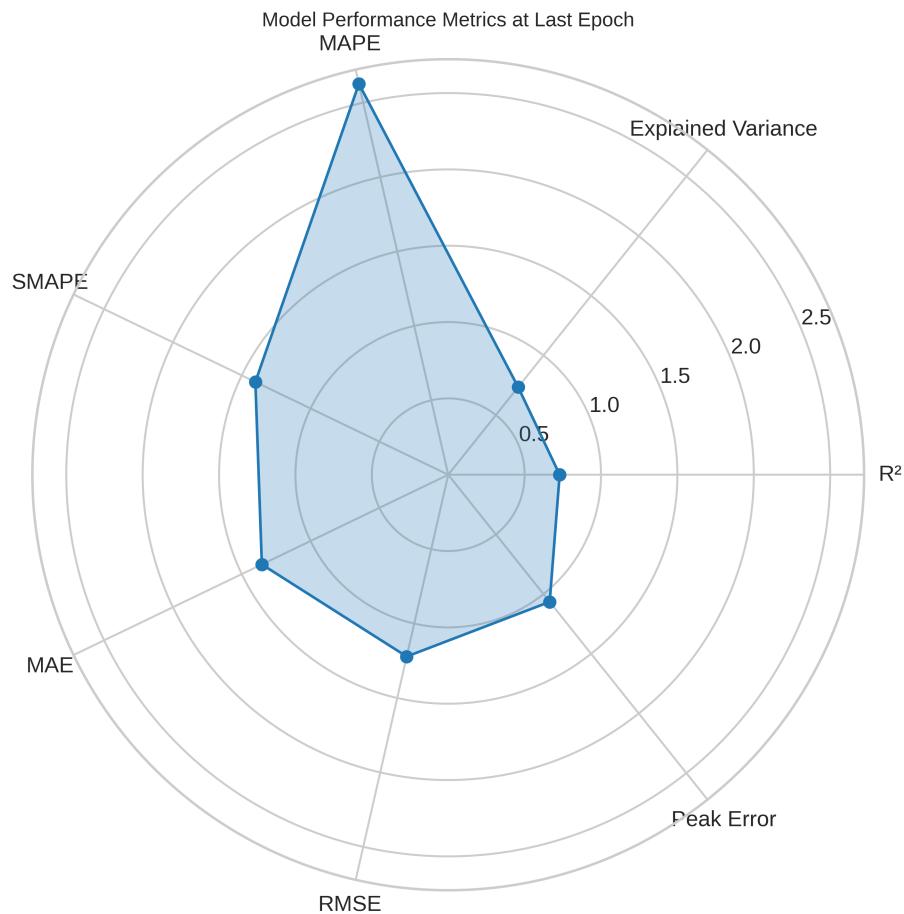
Na grafoch **Loss Train vs Validation**, **MAE Train vs Validation**, **R<sup>2</sup> Train vs Validation** a **RMSE Train vs Validation** je jasne vidieť, že medzi trénovacou a validačnou sadou nie je zjavný overfitting. Hodnoty R<sup>2</sup> 3.3 a RMSE 3.3 ukazujú na vysokú presnosť modelu pri predikcii.



**Obrázok 4 – 14:** Best model training and validation metrics comparison

#### 4.4.4 Komplexná metrická vizualizácia

Radarový graf **Model Performance Radar Chart** sumarizuje výkonnostné metriky modelu na poslednej epoche, čo poskytuje intuitívny prehľad o rovnováhe medzi jednotlivými chybami (MAE 3.3, RMSE 3.3, MAPE 3.3, SMAPE 3.3,  $R^2$  3.3, Explained Variance 3.3, Peak Error 3.3).



Obrázok 4–15: Best model radar chart

#### 4.4.5 Porovnanie výkonu modelu v priebehu epoch

Na viacozmerných krivkách **Metrics Comparison Over Epochs** je možné sledovať paralelne vývoj stratovej funkcie a výkonu ( $R^2$  3.3) v priebehu epoch. Model preukázal stabilný rast presnosti a pokles chyby bez znakov divergencie.



**Obrázok 4 – 16:** Metrics comparison over epochs

Detailnou analýzou grafu možno pozorovať, že iba v záverečných fázach tréningu sa objavuje mierne rozbiehanie validačných metrík. Tento jav môže indikovať začínajúci sa proces pretrénovania modelu, hoci divergencia nie je výrazná. Takáto tendencia naznačuje, že model postupne stráca svoju generalizačnú schopnosť a začína sa príliš špecializovať na trénovacie dátá.

Sumarizujúc výsledky experimentov, optimalizovaný GRU model vykazuje vynikajúce prediktívne schopnosti s vysokou presnosťou na validačnej množine. Použitie early stopping mechanizmu zabezpečilo, že model bol zastavený v optimálnom bode pred výrazným prejavom pretrénovania, čo potvrdzuje efektívnosť zvoleného prístupu k optimalizácii hyperparametrov a tréningu modelu.

## 5 Záver

Cieľom tejto práce bolo navrhnúť, implementovať a vyhodnotiť optimalizačný rámec pre model GRU, ktorý umožňuje efektívnu predikciu na základe časových radov. V priebehu riešenia sme aplikovali tri rôzne prístupy k optimalizácii hyperparametrov: Random Search, Bayesian Search a Grid Search. Finálna verzia modelu bola trénovaná s využitím najlepšej kombinácie hyperparametrov získaných z Grid Search. Následná analýza preukázala vysokú generalizačnú schopnosť a stabilitu modelu.

### 5.1 Zhrnutie dosiahnutých výsledkov

- Bol navrhnutý a implementovaný univerzálny rámec pre hľadanie optimálnych hyperparametrov GRU modelov.
- Vyvinutý model dosiahol výborné výsledky na reálnej dátovej množine (požičky bicyklov v Londýne).
- Získané výsledky boli dôkladne analyzované pomocou metrických ukazovateľov (RMSE, MAE,  $R^2$ , MAPE, SMAPE, atď.).
- Bola overená účinnosť kombinovaného prístupu k optimalizácii – Random a Bayesian Search na zúženie priestoru a Grid Search pre detailné ladenie.
- Najlepší model dosiahol hodnotu  $R^2 = 0.83122$  na validačnej množine, čo predstavuje vysoko presný výsledok vzhľadom na komplexnosť úlohy.

### 5.2 Hodnotenie kvality riešenia

Model vykazuje:

- Rýchlu konvergenciu počas trénovania bez známok výrazného overfittingu.
- Stabilitu počas celého tréningového procesu.
- Výborný súlad medzi trénovacou a validačnou množinou.

- Robustnosť voči výkyvom v dátach, ako ukazujú metriky PEAK ERROR a SMAPE.

Zvolený prístup sa ukázal ako efektívny a prakticky realizovateľný aj pri obmedzených výpočtových zdrojoch, najmä vďaka kombinácií adaptívnych a systematických metód optimalizácie.

### 5.3 Odporúčania a návrhy na zlepšenie

- Rozšíriť množinu testovaných metód o ďalšie optimalizačné techniky ako napr. Tree-structured Parzen Estimator (TPE) alebo Hyperband.
- Preskúmať vplyv iných architektúr (napr. LSTM) pre porovnanie s GRU.
- Implementovať možnosti automatickej predikcie na produkčné dátá v reálnom čase.

### 5.4 Záverečné zhodnotenie

Zvolený prístup bol v praxi úspešný a viedol k vytvoreniu predikčného systému, ktorý je nielen presný, ale aj stabilný a efektívne navrhnutý. Model GRU trénovaný na optimalizovanej sade hyperparametrov dosiahol veľmi dobré výsledky a preukázal potenciál využitia pri predikcii časových radov v rôznych doménach. Celý vývojový cyklus — od analýzy problému cez výber metód, implementáciu, optimalizáciu až po vizualizáciu a vyhodnotenie — bol realizovaný systematicky a metodicky správne.

## Literatúra

- [1] H. Mavrodiev, “London Bike Sharing Dataset” Kaggle, 2025. [Online]. Available: <https://www.kaggle.com/datasets/hmavrodiev/london-bike-sharing-dataset>
- [2] Nikita Chernysh, “bike-gru-experiments” GitHub, 2025. [Online]. Available: <https://github.com/kitnew/bike-gru-experiments>
- [3] A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [4] <https://www.sciencedirect.com/topics/mathematics/grid-search>, ”Grid Search in *Journal of Mathematical Psychology, 2013*”, Jay I. Myung a, Daniel R. Cavagnaro b, Mark A. Pitt a et al.. Available at: <https://doi.org/10.1016/j.jmp.2013.05.005>
- [5] <https://www.sciencedirect.com/topics/engineering/random-search-method>, ”Random Search Method in *Energy Conversion and Management, 2016*”, Houssein Al Moussawi a b, Farouk Fardoun a, Hasna Louahlia-Gualous b et al.. Available at: <https://doi.org/10.1016/j.enconman.2016.04.085>
- [6] H. Alibrahim and S. A. Ludwig, ”Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization,” 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 2021, pp. 1551-1559, doi: 10.1109/CEC45853.2021.9504761. keywords: Training;Machine learning algorithms;Neural networks;Prediction algorithms;Search problems;Time measurement;Bayes methods;Hyperparameter optimization;Grid Search;Bayesian;Genetic Algorithm, Available at: <https://ieeexplore.ieee.org/document/9504761>

## **Zoznam príloh**

**Príloha A** Zdrojový kód

## Príloha A

### Prílohy

[github.com/kitne/bike-gru-experiments](https://github.com/kitne/bike-gru-experiments)