



FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

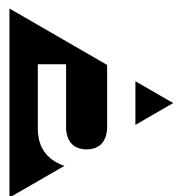
KATEDRA INFORMATIKY  
A VÝPOČETNÍ TECHNIKY

## Diplomová práce

# Detekce anomálií v datech z knih limitních objednávek

Dominik Zappe





FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

KATEDRA INFORMATIKY  
A VÝPOČETNÍ TECHNIKY

## Diplomová práce

# Detekce anomálií v datech z knih limitních objednávek

Bc. Dominik Zappe

**Vedoucí práce**

doc. Ing. Jan Pospíšil, Ph.D.

PLZEŇ

2025

© Dominik Zappe, 2025.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

**Citace v seznamu literatury:**

ZAPPE, Dominik. *Detekce anomálií v datech z knih limitních objednávek*. Plzeň, 2025. Diplomová práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce doc. Ing. Jan Pospíšil, Ph.D.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2024/2025

# ZADÁNÍ DIPLOMOVÉ PRÁCE

## (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Dominik ZAPPE**

Osobní číslo: **A23N0011P**

Studijní program: **N0613A140037 Informatika a její specializace**

Specializace: **Zpracování přirozeného jazyka**

Téma práce: **Detekce anomalií v datech z knih limitních objednávek**

Zadávající katedra: **Katedra informatiky a výpočetní techniky**

### Zásady pro vypracování

1. Seznamte se s technikami detekce anomalií v časových řadách a následně zpracujte rešerši vhodných metod pro detekci anomalií ve velkých souborech dat charakterem podobných datům z knih nákupních a prodejných příkazů (tzv. limit orders books).
2. Navrhněte a pomocí vhodných vývojových nástrojů implementujte algoritmus pro detekci objednávek podezřelých z tzv. spoofingu (tedy zadávání velkých falešných příkazů k nákupu nebo prodeji) a generátor reportů o výskytu těchto případů v analyzovaných datech. Speciální pozornost věnujte metodám detekce fungujícím na principech strojového učení; implementujte alespoň tři odlišné mechanizmy detekce.
3. Realizujte detekci nad velkými daty z velkého množství objednávkových knih ve vhodném prostředí (e-INFRA CZ, MetaCentrum) a vhodným způsobem detailně vizualizujte dynamiku příslušné objednávkové knihy po celou dobu přítomnosti nalezených spoofingových objednávek.
4. Navržené postupy a dosažené výsledky přehledně zdokumentujte v průvodním textu práce.

Rozsah diplomové práce: **doporuč. 50 s. původního textu**  
Rozsah grafických prací: **dle potřeby**  
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

dodá vedoucí diplomové práce

Vedoucí diplomové práce: **Doc. Ing. Jan Pospíšil, Ph.D.**  
Katedra matematiky

Datum zadání diplomové práce: **9. září 2024**  
Termín odevzdání diplomové práce: **15. května 2025**

L.S.

---

**Doc. Ing. Miloš Železný, Ph.D.**  
děkan

---

**Doc. Ing. Přemysl Brada, MSc., Ph.D.**  
vedoucí katedry

# Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., zákon o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění, a zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Plzni dne 15. května 2025

.....  
Dominik Zappe

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

# **Abstrakt**

Moderní finanční trhy jsou rychlé, komplexní a stále častěji se stávají cílem sofistikovaných forem manipulace. Tato práce se zaměřuje na detekci anomálií v časových řadách odvozených z knih limitních objednávek s cílem rozpoznat manipulativní chování zvané spoofing. Vzhledem k absenci anotovaných dat jsou použity metody strojového učení bez učitele aplikované na reálná historická data. V práci je implementováno šest metod – izolační les, lokální faktor odlehlosti, jednotřídní SVM, plně propojený autoenkovodér, konvoluční autoenkovodér a transformer autoenkovodér. Modely jsou evaluovány pomocí méně známých metrik Excess Mass a Mass Volume, přičemž nejlépe si vedou modely izolační les a transformer. Kombinací nejvýkonnějších modelů vznikl robustní nástroj schopný odhalit podezřelé chování bez ruční anotace. Navržené řešení efektivně identifikuje rizikové oblasti pro následnou expertní analýzu a představuje tak praktický přínos pro detekci nelegálních praktik na finančních trzích.

# **Abstract**

Modern financial markets are fast-paced, complex, and increasingly targeted by sophisticated forms of manipulation. This thesis focuses on anomaly detection in time series derived from limit order books, aiming to identify manipulative behavior known as spoofing. Due to the absence of annotated data, unsupervised machine learning methods are applied to real historical data. Six methods are implemented – Isolation Forest, Local Outlier Factor, One-Class SVM, Fully Connected Autoencoder, Convolutional Autoencoder, and Transformer-based Autoencoder. The models are evaluated using the less commonly known metrics Excess Mass and Mass Volume, with the Isolation Forest and Transformer models achieving the best results. By combining the most effective models, a robust tool is created, capable of detecting suspicious behavior without manual annotation. The proposed solution efficiently identifies high-risk areas for subsequent expert analysis and thus offers a practical contribution to detecting illicit practices in financial markets.

# **Klíčová slova**

kniha limitních objednávek • spoofing • detekce anomálií • strojové učení • učení bez učitele • autoenkovodér • vizualizace dat

# Poděkování

Na tomto místě bych rád vyjádřil své upřímné poděkování doc. Ing. Janu Pospíšilovi, Ph.D., vedoucímu mé diplomové práce, za jeho cenné odborné rady a podnětné nápady. Jeho neustálá ochota věnovat mi svůj čas a energii byla klíčová pro úspěšné dokončení této práce. Bez jeho podpory by tato práce nevznikla v této podobě.

Mé upřímné poděkování patří také Ing. Kamilu Ekšteinovi, Ph.D., za to, že mi umožnil zapojit se do projektu s velkými finančními daty a uvedl mě do problematiky této práce.

Velké díky bych chtěl vyjádřit také své rodině a přátelům za jejich trpělivost, pochopení a nepřetržitou podporu během celého mého studia. Zvláště si cením jejich ochoty sdílet se mnou mé radosti i starosti, které studium přinášelo. Bez jejich povzbuzení by má studijní cesta nebyla zdaleka tak snadná a úspěšná.

Zvláštní poděkování patří mému otci, Jozefovi Zappemu, který mi byl během celého studia neocenitelnou oporou. Stejně tak bych chtěl vyjádřit hlubokou vděčnost své přítelkyni, Vladimíře Kimlové. Oba projevili nekonečnou trpělivost a pochopení, a svým povzbuzením mi pomáhali překonávat náročné okamžiky, dodávajíce mi sílu pokračovat dál.

*Dominik Zappe*

Výpočetní a úložné zdroje byly poskytnuty v rámci projektu **e-INFRA CZ (ID:90254)**, podpořeného Ministerstvem školství, mládeže a tělovýchovy České republiky.

Tato práce byla také částečně podpořena z Fondu rozvoje **CESNET**, projekt č. 734/2023 Ukládání, přenos a zpracování velkých vědecko-výzkumných finančních dat v prostředí **e-INFRA CZ**.

Reálná tržní data použitá v této práci byla poskytnuta v rámci Smlouvy o poskytnutí dat pro akademické účely od **Deutsche Börse AG** ze dne 14. 11. 2022.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Přehled současného stavu problematiky . . . . .	2
1.2	Cíl práce . . . . .	3
1.3	Struktura práce . . . . .	3
<b>2</b>	<b>Terminologie a metodologie</b>	<b>5</b>
2.1	Limitní kniha objednávek . . . . .	5
2.1.1	Struktura limitní knihy objednávek . . . . .	6
2.1.2	Typy objednávek v limitní knize . . . . .	7
2.1.3	Algoritmus párování objednávek . . . . .	8
2.1.4	Vliv velkých objednávek na trh . . . . .	8
2.1.5	Uložení limitní knihy objednávek . . . . .	9
2.1.6	Vizualizace knihy objednávek . . . . .	10
2.1.7	Existující řešení . . . . .	12
2.2	Časové řady . . . . .	13
2.2.1	Limitní kniha objednávek jako časová řada . . . . .	15
2.2.2	Detekce anomalií . . . . .	16
2.3	Spoofing . . . . .	17
2.3.1	Konkrétní odhalené případy spoofingu . . . . .	18
2.4	Strojové učení . . . . .	19
2.4.1	Strojové učení a detekce anomalií . . . . .	20
<b>3</b>	<b>Implementovaná řešení</b>	<b>25</b>
3.1	Data . . . . .	26
3.1.1	Předzpracování dat . . . . .	27
3.1.2	Extrakce příznaků . . . . .	28
3.1.3	Redukce dimenze dat . . . . .	32
3.2	Strojové učení . . . . .	33
3.2.1	Vybrané modely . . . . .	33
3.2.2	Trénování . . . . .	35

3.2.3	Evaluace . . . . .	37
3.3	Vizuální analýza . . . . .	40
3.3.1	Interaktivní vizualizační nástroj . . . . .	40
<b>4</b>	<b>Výsledky a diskuze</b>	<b>45</b>
4.1	Vizuální analýza . . . . .	47
4.2	Redukce dimenze dat . . . . .	49
4.2.1	Korelační analýza . . . . .	49
4.2.2	Důležitost příznaků . . . . .	50
4.2.3	Analýza hlavních komponent . . . . .	51
4.2.4	Výsledná redukce dimenze . . . . .	52
4.3	Prohledávání prostoru hyperparametrů . . . . .	55
4.3.1	Prohledávání prostoru hyperparametrů po redukci dimenze vstupních dat . . . . .	61
4.4	Ověření nejlepších hyperparametrů . . . . .	63
4.4.1	Srovnání referenčních modelů . . . . .	64
4.4.2	Srovnání neuronových modelů . . . . .	64
4.4.3	Srovnání všech modelů . . . . .	66
4.5	Detekované anomálie . . . . .	67
4.5.1	Kombinace modelů . . . . .	71
<b>5</b>	<b>Závěr</b>	<b>75</b>
<b>A</b>	<b>Uživatelská příručka</b>	<b>77</b>
<b>B</b>	<b>Pohled přes všechny dimenze</b>	<b>83</b>
<b>Bibliografie</b>		<b>93</b>
<b>Seznam obrázků</b>		<b>99</b>
<b>Seznam tabulek</b>		<b>103</b>
<b>Seznam výpisů</b>		<b>105</b>

# Úvod

1

Spoofing in the market? More like a game of  
'hide-and-seek' with your orders. Except, you're hiding  
them, and no one is seeking.

---

*Anonymous Trader*

V moderním obchodování na finančních trzích hraje klíčovou roli analýza velkého objemu dat generovaných v reálném čase. Jednou z nejvýznamnějších komponent obchodní infrastruktury je tzv. **kniha limitních objednávek** (*Limit Order Book, LOB*), která zaznamenává všechny příkazy k nákupu a prodeji na daném trhu. Tato data poskytují cenné informace o dynamice trhu, ale vzhledem ke své velikosti a složitosti představují zároveň významnou výzvu pro analýzu.

Jedním z klíčových problémů při zpracování těchto dat je identifikace anomalií, které mohou signalizovat neobvyklé chování na trhu nebo dokonce manipulativní praktiky, jako je tzv. **spoofing**. **Spoofing** je nelegální praktika, kdy obchodníci zadávají velké falešné objednávky s cílem ovlivnit cenové pohyby a následně z toho těžit. Detekce těchto aktivit vyžaduje kombinaci pokročilých algoritmů z oblasti zpracování dat a strojového učení.

Jak trefně vystihuje úvodní citát této práce – v prostředí, kde se většina snaží své záměry „schovat“, je zapotřebí těch, kteří se nebojí aktivně „hledat“. Právě tato práce se snaží naplnit roli aktivního pozorovatele – využívá metody strojového učení bez učitele k identifikaci podezřelých vzorců v datech, aniž by vyžadovala ručně anotované případy manipulace.

Využitím reálných historických dat poskytnutých společností **Deutsche Börse AG** se práce opírá o autentický materiál z prostředí reálných finančních trhů. Díky tomu je možné testovat navržené algoritmy ve věrohodných podmínkách. Výpočty probíhaly na infrastruktuře **e-INFRA CZ**, což umožnilo efektivní práci i s velmi rozsáhlými datovými sadami.

## 1.1 Přehled současného stavu problematiky

Problematika detekce manipulací na finančních trzích, zejména spoofingu, se stává stále relevantnější v oblasti **vysokofrekvenčního obchodování** (*High-Frequency Trading, HFT*). Vzhledem k neustálým změnám tržních podmínek a zvyšujícímu se množství dat je stále obtížnější spolehlivě identifikovat manipulativní praktiky, jako je *spoofing*, který spočívá v zadávání a následném zrušení velkých objednávek s cílem ovlivnit tržní ceny. Současné přístupy zahrnují jak tradiční metody, tak i moderní techniky strojového učení, které se stále vyvíjejí.

Jedním z přístupů, které si získávají pozornost, jsou **hybridní modely** (hybridní ve smyslu učení s / bez učitele) kombinující **autoencodery** s **one-class klasifikátory**. Například studie **Nasdaq** z roku 2024 testovala tyto modely v porovnání s **izolačním lesem** (*Isolation Forest*) a **One-Class SVM**. Výsledky ukázaly, že izolační les dosáhl **AUC ROC** 0.82 při práci se syntetickými daty, což naznačuje jejich potenciál v detekci spoofingu i v případě, kdy nejsou data *anotovaná* („bez **labelů**“) [1].

Na druhé straně metody **učení s učitelem** – např. **KNN klasifikátor** trénovaný na datech ze **simulaci s agenty**. Studie ukázaly, že tento přístup, který kombinuje umělé trhy s reálnými daty z **Euronext**, je robustní, ale silně závislý na kvalitě trénovacích dat [2]. Dalším příkladem je použití modelu **GRU** s prahovými podmínkami, který dokáže identifikovat *spoofing* na základě **objemu zrušených objednávek, vzdálenosti bid-ask spreadu a zvýšené volatilitě**, a to pomocí dat z reálných trhů [3].

Současně se objevují novější přístupy založené na analýze anomálních vzorů prostřednictvím **grafových sítí**. Tyto metody, které kombinují lokální a globální vlastnosti trhu, dokázaly dosáhnout **AUC ROC** až 0.95 [4]. Adaptivní prahy reagující na měnící se tržní podmínky jsou příkladem techniky, která se ukazuje jako efektivní v dynamických prostředích.

Mezi zajímavé interdisciplinární přístupy patří například použití nástrojů z oblasti **čisticové fyziky**, jak ukazuje projekt **HighLO**. Tento přístup využívá **ROOT framework** z **CERNu** pro analýzu mikrostruktury trhu a odhalování *spoofingu* prostřednictvím analýzy časoprostorových vzorců v knihách limitních objednávek. Příkladem úspěšné aplikace tohoto přístupu je odhalení *spoofingového* případu **JPMorgan** z roku 2020 [5, 6].

Současný výzkum ukazuje, že existuje široké spektrum metod a přístupů, které se kombinují, aby efektivněji detekovaly a analyzovaly manipulativní praktiky na finančních trzích.

## 1.2 Cíl práce

Tato diplomová práce se zaměřuje na detekci anomalií v časových řadách odvozených z knihy limitních objednávek, se zvláštním zaměřením na identifikaci potenciálních případů *spoofingu*. Cílem je nejen analyzovat a porovnat dostupné metody pro detekci anomalií v neanotovaných datech, ale především navrhnout prakticky použitelný rámec, který dokáže identifikovat podezřelé chování bez nutnosti dohledu člověka nebo předem známých příkladů manipulace.

Jak naznačuje úvodní myšlenka této práce, úkolem je postavit se výzvě „hledání“ tam, kde většina pouze „schovává“. Práce zohledňuje aktuální stav poznání v oblasti strojového učení a analýzy datových souborů, přičemž se soustředí na praktickou aplikaci nad velkými datovými sety.

Hlavní přínos práce spočívá v kombinaci teoretického přístupu a praktické implementace, která umožňuje efektivní detekci a analýzu manipulativních aktivit na finančních trzích. Získané poznatky mohou sloužit jako základ pro další vývoj metod pro zajištění transparentnosti a spravedlnosti na trzích.

## 1.3 Struktura práce

Tato práce je rozdělena do pěti hlavních kapitol, které logicky pokrývají celý proces detekce anomalií ve finančních datech, od teoretických základů až po výsledky a závěry. **Kapitola 1 – Úvod** – nynější kapitola – uvádí čtenáře do problematiky, shrnuje aktuální stav v dané oblasti a formuluje cíle práce. **Kapitola 2 – Terminologie a metodologie** poskytuje přehled klíčových pojmu, které jsou důležité pro pochopení tématu. Detailně je popsána *limitní kniha objednávek*, principy *časových řad*, samotný jev *spoofingu* a základy *strojového učení* se zaměřením na *detekci anomalií*. **Kapitola 3 – Implementovaná řešení** se zaměřuje na popis zpracování dat a implementaci jednotlivých modelů strojového učení. Je zde uveden proces *předzpracování*, *extrakce příznaků*, *redukce dimenziality*, výběr modelů, jejich *trénování* a následná vizuální analýza. **Kapitola 4 – Výsledky a diskuze** prezentuje dosažené výsledky, a to jak z pohledu vizualizace, tak z hlediska jednotlivých kroků analýzy. Kapitola obsahuje podrobnou diskuzi o výběru příznaků, vlivu redukce dimenziality, optimalizaci *hyperparametrů* a o srovnání výkonnosti jednotlivých modelů. Závěrečná část se věnuje detekovaným anomaliím a možnostem kombinace modelů. **Kapitola 5 – Závěr** shrnuje klíčové poznatky, zhodnocuje přínos navrženého řešení a navrhuje možnosti dalšího rozvoje v oblasti detekce tržních manipulací.



# Terminologie a metodologie

— 2

Big Data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it.

---

Dan Ariely (1967-), Professor of Psychology and Behavioral Economics at Duke University, 2013

S růstem objemu dat a zvyšující se složitostí jejich zpracování se problematika práce s **Big Data** stala jedním z klíčových témat moderního výzkumu a praxe. Jak trefně pojmenovává *Dan Ariely*, práce s velkými daty často provází více teorie a očekávání než skutečné efektivity. Cílem této kapitoly je zavést teoretické podklady a také praktické metodologie, které jsou skutečně využitelné pro analýzu dat z finančních trhů.

## 2.1 Limitní kniha objednávek

V dnešní době je **limitní kniha objednávek** (*Limit Order Book, LOB*) jedním z nejdůležitějších nástrojů na finančních trzích řízených objednávkami. Jedná se o dynamickou databázi, která zaznamenává příkazy k nákupu a prodeji pro určitý instrument na daném trhu v reálném čase. Každý příkaz obsahuje informace o ceně, objemu a čase zadání [7, 8].

### 2.1.1 Struktura limitní knihy objednávek

Limitní kniha objednávek se skládá z **nabídkové strany** (*Bid Side*), která obsahuje **objednávky** na **nákup** (*Buy*), a **poptávkové strany** (*Ask Side*), která zahrnuje **objednávky** na **prodej** (*Sell*) [7, 8]. Překvapivě na některých internetových zdrojích dochází k častému zaměňování těchto termínů, což může vést k terminologickým nepřesnostem a potenciálním nejasnostem v interpretaci dat.

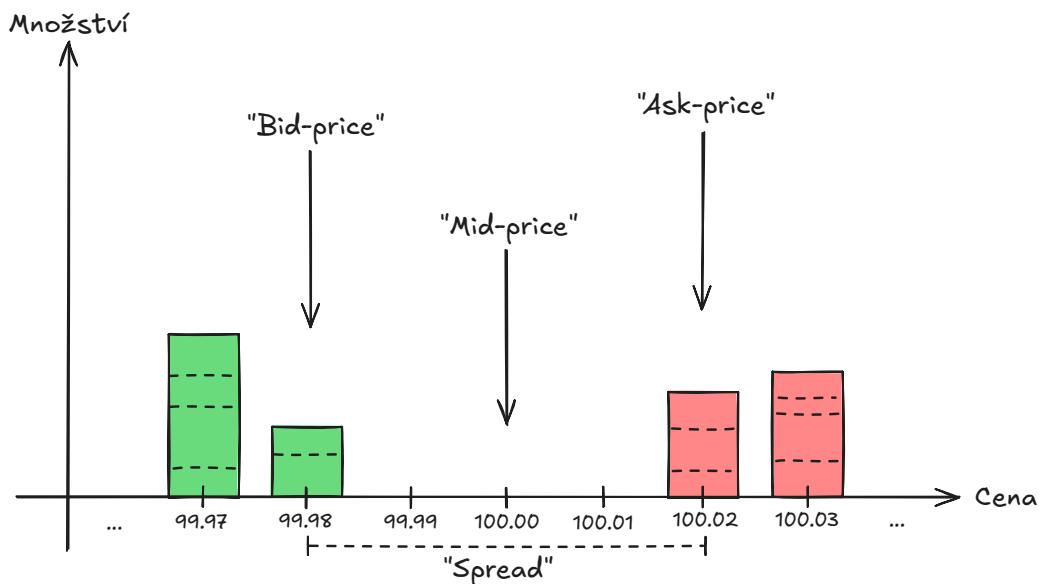
Objednávky jsou v obou případech řazeny podle ceny a času zadání [7, 8]:

- Nabídkové objednávky (*Bid*) jsou seřazeny **sestupně** podle ceny, přičemž nejvyšší nákupní nabídka je na vrcholu a označuje se jako **nabídková cena** (*Bid Price*).
- Poptávkové objednávky (*Ask*) jsou seřazeny **vzestupně** podle ceny, přičemž nejnižší prodejní poptávka je na vrcholu a označuje se jako **poptávková cena** (*Ask Price*).

Aritmetický průměr nabídkové a poptávkové ceny vytváří tzv. **střední cenu** (*Mid Price*) [7].

**Grafická vizualizace.** Výše popsané termíny jsou graficky znázorněny na Obrázku 2.1. Na levé straně je vidět zelenou barvou nabídková strana, spolu s nejlepší nabídka (nabídková cena); obdobně pak na pravé je červenou znázorněna poptávková strana. Mezi nejlepšími cenami obou stran navíc vzniká tzv. **Spread** – rozdíl mezi cenami (může mimo jiné indikovat **likviditu**<sup>1</sup> daného trhu) [7]. Každý sloupec má navíc pro názornost označené přerušovanou čarou různě velké (objemné) objednávky např. i různých obchodníků.

<sup>1</sup> Likvidita trhu označuje schopnost trhu realizovat transakce (nákupy a prodeje) rychle a za stabilní ceny, bez výrazných cenových pohybů. Vysoká likvidita znamená, že na trhu je dostatek kupujících a prodávajících, což umožňuje snadno uskutečnit obchody za aktuální tržní cenu. Naopak nízká likvidita může vést k větší volatilitě a širším „**spreadům**“ mezi nabídkovou a poptávkovou cenou [9].



Obrázek 2.1: Příklad možné vizualizace knihy limitních objednávek pro konkrétní časový okamžik

**Atributy objednávky.** Každá objednávka je dále charakterizována těmito klíčovými atributy [7, 8]:

- **Cena (Price)**, za kterou je obchodník ochoten nakoupit (resp. prodat).
- **Objem (Volume)** značí počet jednotek<sup>2</sup> daného instrumentu, který je předmětem obchodu.
- **Časová značka (Timestamp)** – čas, kdy se objednávka dostala do systému<sup>3</sup>.

## 2.1.2 Typy objednávek v limitní knize

Existuje několik typů objednávek, které se liší svým účelem a pravidly realizace. Mezi nejčastější patří [8]:

- **Limitní objednávky (Limit Orders)** stanovují maximální (resp. minimální) cenu, za kterou je obchodník ochoten nakoupit (resp. prodat).  
Příklad: „Chtěl bych koupit 100 akcií za cenu maximálně 100 Kč.“

<sup>2</sup>**Lot** představuje standardní množství jednotek daného instrumentu, po kterých se obchoduje na finančních trzích. Tento pojem vyjadřuje granularitu konkrétní komodity a každá objednávka musí být zadávána v celočíselných násobcích tohoto množství.

<sup>3</sup>Objednávka může mít se sebou spojených více časových značek, viz [10].

- **Tržní objednávky** (*Market Orders*) jsou realizovány okamžitě za nejlepší dostupnou cenu v daném čase.  
Příklad: „Chtěl bych koupit 100 akcií hned teď (za jakoukoliv cenu).“
- **Stop objednávky** (*Stop Orders*) se aktivují pouze při dosažení specifikované ceny (*Stop Price*).  
Příklad: „Chtěl bych koupit akcie, pokud cena klesne pod 100 Kč.“
- **IOC objednávky** (*Immediate or Cancel*) jsou realizovány okamžitě; pokud to není možné, tak nevyřízená část objednávky se automaticky zruší.
- **FOK objednávky** (*Fill or Kill*) jsou realizovány okamžitě a **kompletně**, jinak jsou **zcela** zrušeny.
- **Iceberg objednávky** mají viditelnou pouze malou část objednávky. Zbytek se postupně zveřejňuje tím, jak se realizují jednotlivé části obchodů.

### 2.1.3 Algoritmus párování objednávek

Každý trh je doplněn nezbytným **algoritmem párování objednávek**, jehož úkolem je efektivně zpracovávat příchozí objednávky. Pokud dorazí nová objednávka, algoritmus se pokusí tuto objednávku spárovat s již existující protistranou. V případě, že není možné najít odpovídající protistranu, objednávka se stává aktivní a je uložena do limitní knihy. Objednávka zůstává v knize, dokud není realizován odpovídající obchod, nebo dokud ji obchodník nezruší – to obvykle znamená, že se rozhodl již neprodat nebo nekoupit za původně stanovenou cenu [7, 8].

**Prioritní fronta a řazení.** Algoritmus párování objednávek je úzce spojen s frontou aktivních objednávek, pro řazení v rámci této prioritní fronty existuje mnoho různých přístupů – nejčastěji se však priorita v rámci stejné ceny řadí podle příchozího času (*Timestamp*) objednávky; nejprve je celkově řazeno dle ceny (*Price*) [7, 8].

### 2.1.4 Vliv velkých objednávek na trh

**Velké objednávky** (*block orders*) mohou mít významný dopad na **dynamiku trhu** a strukturu limitní knihy objednávek. Vzhledem k jejich velikosti mohou tyto objednávky způsobit výrazné pohyby cen a ovlivnit chování ostatních účastníků trhu [8].

Velké objednávky ovlivňují dostupné ceny na opačné straně knihy objednávek – dochází tak k posunu ceny. Velké objemy mohou také „vyčerpat“ dostupné objednávky v knize a může tak dojít ke **snižení likvidity** trhu [8].

**Zneužití velkých objednávek.** V některých případech mohou být velké objednávky využity k manipulaci s trhem. Příklady takových manipulativních burzou zakázaných technik zahrnují [11, 12]:

- **Spoofing** – zadávání velkých objednávek s úmyslem je stáhnout dříve, než budou provedeny. Tato taktika slouží k manipulaci s cenami vytvořením iluze *vysoké poptávky* nebo *nabídky* na trhu [13]. Podrobněji se této nelegální praktice věnuje Kapitola 2.3.
- **Layering** – strategické zadávání více velkých objednávek napříč různými cenovými úrovněmi, které tak mají vytvářet iluzi **vysoké likvidity** (nebo poptávky) [13].

## 2.1.5 Uložení limitní knihy objednávek

Ukládání dat z limitní knihy objednávek je klíčovým krokem pro jejich následnou analýzu, detekci anomalií a návrh modelů. Struktura a formát těchto dat závisí na konkrétním použití a požadavcích na efektivitu ukládání, rychlosť přístupu a kompatibilitu s analytickými nástroji.

**Formáty ukládání.** Pro uložení limitní knihy objednávek je možné použít následující formáty po vzoru **Deutsche Börse Group** [14, 15]:

- **JSON (JavaScript Object Notation)** – vhodný formát pro ukládání koplexních struktur dat – jako je časová značka s kompletními detaily objednávek. Hlavní výhodou tohoto formátu je snadná rozšířitelnost. Naopak hlavní nevýhodou jsou vysoké paměťové nároky ve srovnání s jinými formáty [16].
- **CSV (Comma-Separated Values)** – vhodný formát pro jednoduché ukládání a základní analýzu dat. Typicky sloupce mohou zahrnovat časovou značku, jednotlivé ceny a objemy, případně další detaily objednávek. Hlavní výhodou je lidská čitelnost a široká podpora analytickými nástroji. Mezi hlavní nevýhody patří absence metadat a omezená efektivita při práci s velkými objemy dat (*Big Data*) [17].

Dalším vhodným formátem se nabízí formát postavený na CSV – **LOBSTER** (*Limit Order Book System – The Efficient Reconstructor*). Z názvu je již jasné, že se jedná o formát vhodný pro efektivní uložení zrekonstruované limitní knihy objednávek [18]. Příklad *LOBSTER* formátu, který odpovídá předchozímu Obrázku 2.1 je vidět na Tabulce 2.1.

Tabulka 2.1: Příklad *LOBSTER* formátu (do druhé úrovně cen a objemů)

Ask Price 1	Ask Size 1	Bid Price 1	Bid Size 1	Ask Price 2	Ask Size 2	Bid Price 2	Bid Size 2	...
:	:	:	:	:	:	:	:	..
100.02	50	99.98	30	100.03	60	99.97	70	...
:	:	:	:	:	:	:	:	..

**Databázová řešení a ukládání ve velkém měřítku.** Této problematice se věnuje bakalářská práce [19], na kterou tato práce navazuje a využívá její výsledky. Konkrétně byl rozšířen nástroj pro vizualizaci zrekonstruovaných objednávkových knih pro manuální detekci a verifikaci potencionálních anomalií. Rekonstrukce samotné využitelné bohužel nebyly.

Pro zpracování velkých objemů knihy objednávek dat lze použít relační i nerelační databáze:

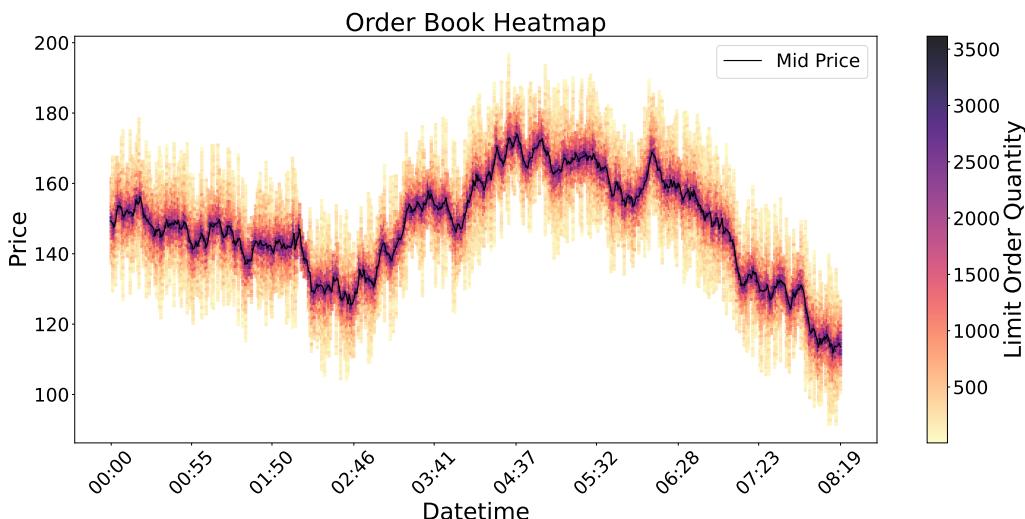
- **SQL** databáze (např. *PostgreSQL*) umožňují ukládání strukturovaných dat a jejich dotazování.
- **NoSQL** databáze (např. *MongoDB*) jsou vhodné pro hierarchická a semi-strukturovaná data – např. JSON.
- **Time-series** databáze (např. *InfluxDB*) jsou optimalizované pro ukládání časových řad, což se zdá být ideálním řešením pro data z knihy objednávek.

## 2.1.6 Vizualizace knihy objednávek

Vizualizace knihy objednávek (*LOB*) je důležitá pro pochopení tržní dynamiky a chování jednotlivých účastníků trhu. Prostřednictvím grafických zobrazení lze snadno analyzovat vztahy mezi nabídkou a poptávkou, identifikovat trendy a detektovat anomálie, které mohou být spojeny s tržní manipulací.

**Typy vizualizací.** Existuje několik přístupů k vizualizaci knihy objednávek, z nichž každý nabízí specifický pohled na data. Zajímavý souhrn všech možných vizualizací poskytuje práce Dumas et al. [20]. Mezi klasické způsoby patří (zmíněné vizualizace jsou často užívány v kombinaci pro lepsí interpretovatelnost viditelných fenoménů) [21]:

- **Hloubkový graf (Depth Chart)** zobrazuje kumulovanou hloubku objednávek na nabídkové i poptávkové straně. Křivky se kříží v oblasti nejlepší nabídkové a poptávkové ceny, přičemž rozdíl mezi nimi představuje **spread**. Tato vizualizace již byla ukázana v Kapitole 2.1.1 na Obrázku 2.1.
- **Teplovní mapa (Heatmap)** zobrazuje intenzitu objednávek na různých cenových úrovních pomocí barevné škály. Často se používá pro zobrazení historických změn v knize objednávek, čímž umožňuje identifikaci vzorců chování trhu. Tato vizualizace je vidět na Obrázku 2.2.
- **Časová osa (Time Series)** slouží ke sledování vývoje střední ceny (*Mid Price*), *spreadu* a dalších klíčových ukazatelů v čase. Příklad vizualizace je k nahlédnutí na Obrázku 2.2



Obrázek 2.2: Příklad možné vizualizace knihy limitních objednávek jako časové řady v kombinaci s teplovní mapou

## 2.1.7 Existující řešení

Existuje několik komerčně dostupných i *open-source* nástrojů a knihoven, které umožňují implementaci a vizualizaci knihy objednávek (*LOB*). Tato řešení mohou být využita pro analýzu trhů a testování obchodních algoritmů. Mezi nejznámější a nejběžněji používané nástroje patří následující:

**A7 (Advanced Order Book).** A7 je sofistikovaný proprietární nástroj od **Deutsche Börse** určený pro analýzu a vizualizaci limitních knih objednávek na finančních trzích. Tento nástroj nabízí širokou škálu funkcí pro zobrazení *hloubky trhu*, analýzu *order flow* a identifikaci klíčových cenových úrovní, kde dochází k vyšší *likviditě* (nebo potenciálním manipulacím). A7 poskytuje interaktivní vizualizace, které obchodníkům umožňují sledovat dynamiku trhu v reálném čase [15].

**Bookmap.** Bookmap je proprietární vizualizační nástroj, který poskytuje interaktivní zobrazení knihy objednávek v reálném čase. Umožňuje obchodníkům a analytikům vizualizovat cenové úrovně, objemy a pohyby na trhu na základě *order flow*, čímž poskytuje vhled do *likvidity* a dynamiky trhu. Bookmap nabízí pokročilé funkce pro sledování změn v hloubce trhu [22].

**QuantConnect.** QuantConnect je open-source platforma pro vývoj a testování algoritmických obchodních strategií. Ačkoli se zaměřuje především na algoritmické obchodování, nabízí také nástroje pro práci s historickými daty, včetně knihy objednávek. S pomocí QuantConnect mohou obchodníci testovat a implementovat obchodní strategie založené na analýze knihy objednávek a dalších tržních ukazatelů. Podporuje několik trhů a burz a umožňuje integraci s různými externími datovými zdroji [23].

**NinjaTrader.** NinjaTrader je známý proprietární nástroj pro obchodování a analýzu trhů, který podporuje vizualizaci knihy objednávek. NinjaTrader poskytuje obchodníkům přehled o dostupné *likviditě* na různých cenových úrovních a umožňuje vizualizace v reálném čase. Tato platforma je oblíbená mezi profesionálními obchodníky díky svému robustnímu rozhraní a široké podpoře různých tržních instrumentů [24].

**Lightweight Charts.** **Lightweight Charts** je open-source knihovna pro vizualizaci finančních trhů, která se zaměřuje na rychlou a efektivní tvorbu interaktivních grafů s nízkými nároky na výkon. Tento nástroj je ideální pro použití na webových stránkách nebo v aplikacích, které potřebují zobrazovat živá tržní data, včetně knihy objednávek. Knihovna nabízí širokou škálu grafických možností, které mohou být kombinovány pro zobrazení různých aspektů trhu [25].

**Vizualizace s využitím knihoven v Pythonu.** Pro analýzu a vizualizaci knihy objednávek existuje také řada knihoven v jazyce *Python*, které mohou sloužit k vytvoření vlastních analytických nástrojů. Mezi nejpoužívanější patří:

- **Matplotlib**<sup>4</sup> a **Seaborn**<sup>5</sup> – pro vytváření jednoduchých vizualizací a grafů.
- **Plotly**<sup>6</sup> – pro interaktivní vizualizace, které umožňují obchodníkům podívat se na změny v knihách objednávek v reálném čase.
- **Bokeh**<sup>7</sup> – pro interaktivní vizualizace s velkým množstvím dat.
- **Pandas**<sup>8</sup> – pro efektivní zpracování historických dat a jejich následné analýzy.

## 2.2 Časové řady

**Časové řady** představují uspořádaný soubor datových bodů, kde každé pozorování odpovídá konkrétnímu okamžiku  $t$ . Takové řady mohou být [26]:

- **Diskrétní** – pozorování probíhá v konkrétních časových okamžicích (např. denní uzavírací ceny akcií).
- **Spojité** – pozorování je dostupné v každém časovém okamžiku v určitém intervalu (např. teplota měřená v reálném čase).

Příklady časových řad zahrnují finanční data (např. indexy akcií nebo vývoj měnových kurzů), meteorologická měření (např. denní srážky a teploty) a ekonomické ukazatele (např. roční hrubý domácí produkt) [26].

---

<sup>4</sup><https://pypi.org/project/matplotlib/>

<sup>5</sup><https://pypi.org/project/seaborn/>

<sup>6</sup><https://pypi.org/project/plotly/>

<sup>7</sup><https://pypi.org/project/bokeh/>

<sup>8</sup><https://pypi.org/project/pandas/>

**Matematická definice.** Matematicky lze časovou řadu reprezentovat jako posloupnost hodnot  $y_t$ , kde  $t \in \mathbb{T}$  je index časového okamžiku. Zápis této posloupnosti je např.:

$$(y_t)_{t \in \mathbb{T}}, \quad t = 1, 2, 3, \dots \quad (2.1)$$

kde  $y_t$  je hodnota v čase  $t$ . Pro diskrétní časovou řadu jsou hodnoty  $t$  zpravidla celá čísla, zatímco pro spojitu časovou řadu může být  $t$  libovolné reálné číslo v nějakém časovém intervalu [26].

**Analýza časových řad.** Klíčovým cílem analýzy časových řad je modelování jejich dynamiky v čase. To zahrnuje určení statistických údajů jako např. **střední hodnoty** (pro odhad očekávané hodnoty v čase) nebo např. **kovariance** (pro analýzu vztahů mezi jednotlivými hodnotami v čase) [26, 27].

Z hlediska vlastností lze časové řady rozdělit na dvě základní skupiny [26, 27]:

- **Stacionární** – statistické vlastnosti časové řady (např. **střední hodnota** a **rozptyl**) se nemění v čase.
- **Nestacionární** – statistické vlastnosti časové řady se v čase mění.

Každou časovou řadu lze rozložit na několik základních složek, které společně určují její chování [27]:

- **Trend** – dlouhodobý směr vývoje řady (např. postupný růst HDP).
- **Sezónnost** – periodické vzorce spojené s pravidelnými událostmi (např. sezonní poptávka v maloobchodě).
- **Náhodný šum** – nepravidelné, nepředvídatelné změny způsobené náhodnými faktory.

Detailní analýza časových řad je zásadní pro odhalení jejich vnitřní struktury a pro predikci budoucího vývoje. Typické cíle analýzy jsou [27]:

- **Predikce budoucích hodnot** – například využití autoregresních modelů nebo metod strojového učení.

- **Detekce anomalií** – identifikace neobvyklých událostí, které mohou indikovat významné změny nebo rizika, více v Kapitole 2.2.2.
- **Odhad volatility** – kvantifikace rizika spojeného s časovými řadami, například u finančních trhů.

## 2.2.1 Limitní kniha objednávek jako časová řada

Jedním z pohledů na limitní knihu objednávek (*LOB*) je její interpretace jako **časová řada**. Ačkoliv je kniha objednávek primárně vnímána jako dynamická databáze, která zachycuje aktuální stav trhu v daném okamžiku, její postupný vývoj v čase lze vnímat jako posloupnost datových bodů. Tyto datové body reprezentují změny cen, objemů nebo jiných atributů objednávek, což otevírá možnosti pro analýzu pomocí metod určených pro práci s časovými řadami [8].

**Granularita časových řad v LOB.** Limitní kniha objednávek může být sledována na různých úrovních granularit, např. v rámci sekundových nebo milisekundových intervalů. Na každé časové úrovni lze analyzovat různé aspekty trhu, jako jsou změny cen, objemů obchodů nebo změny v hloubce knihy. Takto získané časové řady lze analyzovat za účelem *pochopení tržních dynamik*, *odhadování rizik* nebo *modelování budoucích pohybů cen* [8, 26].

**Analytické postupy.** V reálném čase se analýza knihy objednávek často používá k monitorování tržní *likvidity*, identifikaci potenciálních příležitostí nebo k odhadu změn cen. Časové řady z limitní knihy umožňují *obchodníkům* a *analytikům* modelovat tržní pohyby a predikovat reakce na určité obchodní akce. Příkladem může být predikce změn střední ceny (*Mid Price*) na základě historických trendů v knize objednávek [8, 27].

Vysoká dynamika limitní knihy objednávek a citlivost na krátkodobé změny činí její analýzu složitou. Vzhledem k rychlým změnám v nabídce a poptávce na finančních trzích je obtížné predikovat okamžité změny. Příchozí objednávky mají často velmi vysokou frekvenci (v rádech milisekund), což opět značně stěžuje analýzu v reálném čase. Mezi užívané techniky pro zpracování knihy objednávek patří např. [27]:

- **Autoregresní modely (AR)** – tyto modely lze aplikovat na predikci změn cen nebo objemů v závislosti na historických hodnotách v limitní knize objednávek. Autoregresní modely jsou vhodné pro analýzu závislostí v časových řadách a mohou být rozšířeny o sezónnost a trend.

- **Skryté Markovské modely (HMM)** – modelování pomocí Markovských řetězců se často používá pro analýzu změn v tržních podmínkách. Například změny v likviditě mohou být modelovány jako přechody mezi různými tržními režimy.
- **Statistické metody pro detekci anomalií** – analýza limitní knihy objednávek pomocí statistických testů může odhalit neobvyklé změny v cenách nebo objemech, což může signalizovat manipulaci s trhem nebo rizika v rámci určitého časového intervalu. Obecné analýzy pro detekci anomalií se věnuje následující Kapitola 2.2.2.

## 2.2.2 Detekce anomalií

**Anomalie** lze obecně definovat jako data, která se významně odlišují od ostatních pozorování v datovém souboru [28]. Konkrétně u limitní knihy objednávek mohou tyto odchylky být způsobeny specifickými událostmi, jako jsou výrazné změny v poptávce a nabídce, technické chyby, nebo dokonce manipulativní chování některých obchodníků, například *spoofing* [13].

**Metody detekce anomalií.** Existuje několik přístupů pro detekci anomalií, které jsou běžně používané při analýze knihy objednávek a časových řad [27, 28]:

1. **Statistické metody** předpokládají, že data následují určitý pravděpodobnostní model. Hodnoty, které spadají mimo předem definovaný interval (například tři standardní odchylky od střední hodnoty), jsou považovány za anomálie.
  - **Z-skóre** – měří počet standardních odchylek hodnoty od průměru.
  - **Percentilové analyzy** – identifikují extrémní hodnoty na základě distribuce dat.
2. **Modely založené na časových řadách** detekují anomálie s využitím historických dat a trendů. Konkrétním příkladem jsou **autoregresivní modely (ARIMA)** – umožňují identifikovat hodnoty, které nesedí s predikovaným vývojem.
3. **Strojové učení** hraje zásadní roli při detekci složitějších a nelineárních vzorců v datech. Nejčastěji používaným metodám je věnována Kapitola 2.4

**Výzvy při detekci anomálií.** Detekce anomálií na finančních trzích čelí několika výzvám. Data z limitní knihy objednávek mohou (a často i jsou) aktualizována v milisekundových intervalech – práce s tak jemnou granularitou vyžaduje výkonné metody zpracování dat. Statistické vlastnosti trhu se mohou měnit v čase, což ztěžuje aplikaci stacionárních modelů a statických metod. Některé anomálie navíc mohou být výsledkem přirozených tržních fluktuací, nikoliv skutečných problémů (či manipulace) – mnoho falešně pozitivních výsledků. Největší výzvou může být, že v mnoha případech není dopředu známo, jak anomálie vypadají, což komplikuje jejich detekci pomocí klasických modelů – příkladem může být *spoofing* [13, 27, 28].

## 2.3 Spoofing

**Spoofing** je jednou z praktik manipulace s finančními trhy, která spočívá v zadávání velkých objednávek s úmyslem je stáhnout předtím, než budou realizovány. Tato strategie je využívána k ovlivnění tržních cen a chování ostatních obchodníků, přičemž samotné objednávky nejsou zamýšleny k obchodní realizaci [11, 12]. Tento postup je považován za nelegální v mnoha jurisdikcích [13].

*Spoofing* zahrnuje několik základních kroků [11, 29]:

1. **Vytvoření iluze tržní poptávky (resp. nabídky).** Obchodník zadá velké objednávky na jedné straně knihy, tím vytvoří dojem vysokého zájmu o nákup (resp. prodej) finančního instrumentu.
2. **Ovlivnění chování ostatních účastníků trhu.** Ostatní obchodníci, včetně algoritmických agentů, interpretují falešné objednávky jako indikátor možného budoucího pohybu ceny a mohou tak přizpůsobovat své obchodní strategie.
3. **Stornování falešných objednávek.** Těsně před realizací obchodu jsou tyto objednávky zrušeny, aby se zabránilo skutečnému vypořádání obchodu.
4. **Využití cenového pohybu.** Obchodník, jež manipuloval trhem, těží z manipulovaného pohybu ceny buď realizací zisků na protistraně trhu, nebo vytvořením příznivějších podmínek pro své budoucí obchody.

**Negativní dopady na trh.** Falešné objednávky ovlivňují přirozenou tvorbu cen a narušují mechanismus nabídky a poptávky. Přítomnost manipulativních praktik, jako je *spoofing*, může vést k nedůvěře mezi účastníky trhu a snížení celkové likvidity trhu. Spoofing mimo jiné také může způsobit přímé ztráty ostatním obchodníkům, kteří reagují na falešné tržní signály [11, 12].

**Detekce spoofingu.** Detekce spoofingu představuje významnou výzvu kvůli jeho krátkodobé a dynamické povaze. Jak již bylo zmíněno dříve, dokázaných případů *spoofingu* není mnoho, proto neexistuje žádná přesná datová sada pro snadné natrénování klasických modelů (např. strojové učení, *učení s učitelem*). V praxi mohou být užity statistické metody pro analýzu objednávkové knihy za účelem detekce neobvyklých vzorců, jako jsou rychle zadávané velké objednávky. Další praktikou může být **monitoring chování účastníků na trhu**. Moderním přístupem je užití algoritmů strojového učení – problém však může být s kvalitou a anotací (*Labeling*) dat [6, 11, 12].

**Regulace a právní důsledky.** Regulační orgány, jako je americká **Commodity Futures Trading Commission (CFTC)** nebo evropské orgány dohledu, přistoupily k zavedení přísnějších pravidel a pokut za manipulativní praktiky, včetně *spoofingu*. Postihy zahrnují vysoké finanční sankce, dočasné nebo trvalé zákazy obchodování a v některých případech i trestní stíhání [12, 29].

### 2.3.1 Konkrétní odhalené případy spoofingu

**JPMorgan Chase (2020).** V roce 2020 souhlasila společnost **JPMorgan Chase & Co.** s tím, že zaplatí rekordní pokutu ve výši 920 milionů dolarů za manipulativní praktiky, včetně *spoofingu* na trzích s drahými kovy a americkými státními dluhopisy. Obvinění zahrnovalo koordinované akce mezi obchodníky, kteří zadávali velké objednávky s úmyslem je stáhnout, aby ovlivnili tržní ceny a zvýšili zisky. Tento případ je považován za jeden z největších, jaký kdy byl v oblasti *spoofingu* řešen [6, 30].

**Deutsche Bank (2021).** **Deutsche Bank** byla obviněna z toho, že její obchodníci se zapojili do *spoofingu* na trzích s drahými kovy. V rámci vyrovnání s americkými regulačními orgány souhlasila banka se zaplacením 130 milionů dolarů. Vyšetřování ukázalo, že obchodníci používali manipulativní praktiky k ovlivnění cen zlata a stříbra, což poškodilo ostatní účastníky trhu [31].

**Bank of Nova Scotia (2020).** Kanadská banka **Bank of Nova Scotia** souhlasila s tím, že zaplatí 127,5 milionu dolarů, aby urovnala obvinění z manipulace trhů s drahými kovy. Vyšetřování odhalilo, že obchodníci banky se podíleli na *spoofingu* a dalších manipulativních praktikách v období několika let [32].

**Tower Research Capital (2019).** Tower Research Capital, společnost zabývající se vysokofrekvenčním obchodováním, souhlasila se zaplacením pokuty ve výši 67 milionů dolarů za *spoofing* na trzích s *futures* kontrakty. Podle americké CFTC se obchodníci společnosti podíleli na *spoofingu* mezi lety 2012 a 2013, kdy zadávali a rušili tisíce falešných objednávek, aby ovlivnili tržní ceny [33].

**Michael Coscia (2015).** Michael Coscia byl prvním obchodníkem v USA, který byl trestně obviněn podle zákonů zakazujících *spoofing*. Byl shledán vinným z manipulace trhů s futures a odsouzen na tři roky do vězení. Coscia používal algoritmy, které automaticky zadávaly velké objednávky, jež byly následně zrušeny, aby vytvořily klamné tržní signály [34].

**Navinder Singh Sarao (2015).** Navinder Singh Sarao byl obviněn z manipulace s *futures* kontrakty na S&P 500 prostřednictvím *spoofingu*, což údajně přispělo k tzv. *Flash Crash* v roce 2010. Sarao používal algoritmy, které zadávaly velké objednávky na prodej, které následně rušil, aby vytvořil falešné tržní signály a manipuloval cenami. V roce 2015 byl zadržen v Británii na základě amerických obvinění a čelil porušení zákonů proti manipulaci trhů [35].

## 2.4 Strojové učení

**Strojové učení** (*Machine Learning, ML*) představuje oblast umělé inteligence, která se zabývá vývojem algoritmů schopných automaticky se zlepšovat na základě nabytých zkušeností – tedy dat. Hlavním cílem strojového učení je extrakce užitečných vzorců z dat bez explicitního naprogramování pravidel. Tímto způsobem je možné efektivně řešit úlohy, kde je tradiční programování nepraktické nebo i nemožné – například klasifikace komplexních skrytých struktur v datech, predikce tržního chování nebo rozpoznávání anomalií [36, 37].

Typicky se ve strojovém učení rozlišují tři základní přístupy [36]:

- **Učení s učitelem** (*Supervised Learning*) – algoritmus se učí z dat, která obsahují vstupy a k nim přiřazené výstupy (tzv. „*labels*“) – jedná se předem oanotovaná data. Cílem je vytvořit model, který bude schopný predikovat výstup i pro nové, dosud neviděné vstupy. Tento přístup je často využíván při klasifikaci do předem známých kategorií.

- **Učení bez učitele** (*Unsupervised Learning*) – v tomto případě jsou k dispozici pouze vstupní data bez předem definovaných výstupů – data nejsou anotovaná. Algoritmus se snaží objevit skryté struktury nebo vzorce v datech, typicky pomocí **shlukování** (*Clustering*). Tento přístup je velmi relevantní pro detekci anomalií, kdy často data s jasně definovanými příklady anomálního chování nejsou k dispozici.
- **Posilované učení** (*Reinforcement Learning*) – využívá se v situacích, kdy agent interahuje s prostředím a učí se na základě zpětné vazby – zpětná vazba je formě odměn a trestů. Tento přístup není příliš v kontextu práce užitečný, ale nachází uplatnění například při optimalizaci strategií obchodování na základě dlouhodobé výnosnosti.

**Strojové učení a finanční data.** Finanční data, zejména ta z vysokofrekvenčního obchodování (*HFT*), představují pro strojové učení specifickou výzvu. Tato data jsou ovlivněna externími událostmi a obsahují značný podíl šumu. Navíc se kniha objednávek může měnit až v řádu stovek za sekundu, práce s takovými daty v reálném čase vyžaduje jak výkonné algoritmy, tak vhodnou datovou reprezentaci.

### 2.4.1 Strojové učení a detekce anomalií

V kontextu detekce anomalií nabízí strojové učení výhodu v možnosti modelovat i velmi složité a nelineární vztahy v datech. Zatímco tradiční metody často spoléhají na předpoklady o distribuci dat (např. *normalita*), moderní techniky strojového učení – jako jsou např. **izolační lesy** (*Isolation Forest*), **hluboké neuronové sítě** (*Deep Neural Networks*) nebo **autoenkovadery** (*Autoencoders*) – jsou schopny adaptivně rozpoznávat neobvyklé chování bez těchto předpokladů a omezení.

#### Vybrané modely

V rámci práce byly vybrány tři modely do každé skupiny modelů – **referenční modely** a modely využívající neuronové sítě – dále jen **neuronové modely**. Referenční modely slouží jako „baseline“ pro srovnání. Všechny neuronové modely jsou založeny na principu *autoenkovadér*, liší se vybranou architekturou.

**Izolační les (Isolation Forest, IF).** Izolační les je algoritmus, který je navržen tak, aby efektivně detekoval anomálie v datových souborech. Tento algoritmus je speciálně vhodný pro datové sady, které mají velký objem, což je právě případ dat z *limitní knihy objednávek*. Algoritmus funguje na těchto principech [38]:

- **Izolace dat.** Algoritmus využívá „princip izolace“ – cílem algoritmu je izolovat každý datový vzorek. Anomálie jsou bodové hodnoty, které jsou snadno izolovatelné, protože se nacházejí daleko od většiny ostatních datových bodů. Naopak, běžné (normální) body jsou silněji propojené s ostatními body a izolovat je vyžaduje více kroků.
- **Stromová struktura.** Algoritmus vytváří tzv. **izolační stromy**, což jsou binární stromy, které rekurzivně rozdělují data do menších částí. Iterativně se rozdělují data na základě náhodně vybraných atributů a hodnot, dokud nedojde k izolaci jednotlivých datových bodů.
- **Izolační index.** Čím rychleji je bod izolován, tím vyšší je pravděpodobnost, že se jedná o anomálii. Bodům, které jsou izolovány po relativně malém počtu rozdělení, je přiřazeno vysoké **skóre anomálnosti**.

Výhodou *izolačních lesů* navíc je, že algoritmus je velmi efektivní a robustní – algoritmus umožňuje práci s velkými daty a není silně ovlivněn počtem atributů (dimenzí) v datech [38].

**Jednotřídní podpůrný vektorový stroj (One-Class SVM, OCSVM).** Jednotřídní SVM je model, který je speciálně navržen pro detekci anomálií v nelineárních a neanotovaných datech. Jedná se o speciální variantu klasického SVM (*Support Vector Machine*), který se obvykle využívá ke klasifikaci, ale zde je použit pro detekci odlehlých bodů – anomálií. Algoritmus funguje na těchto principech [39]:

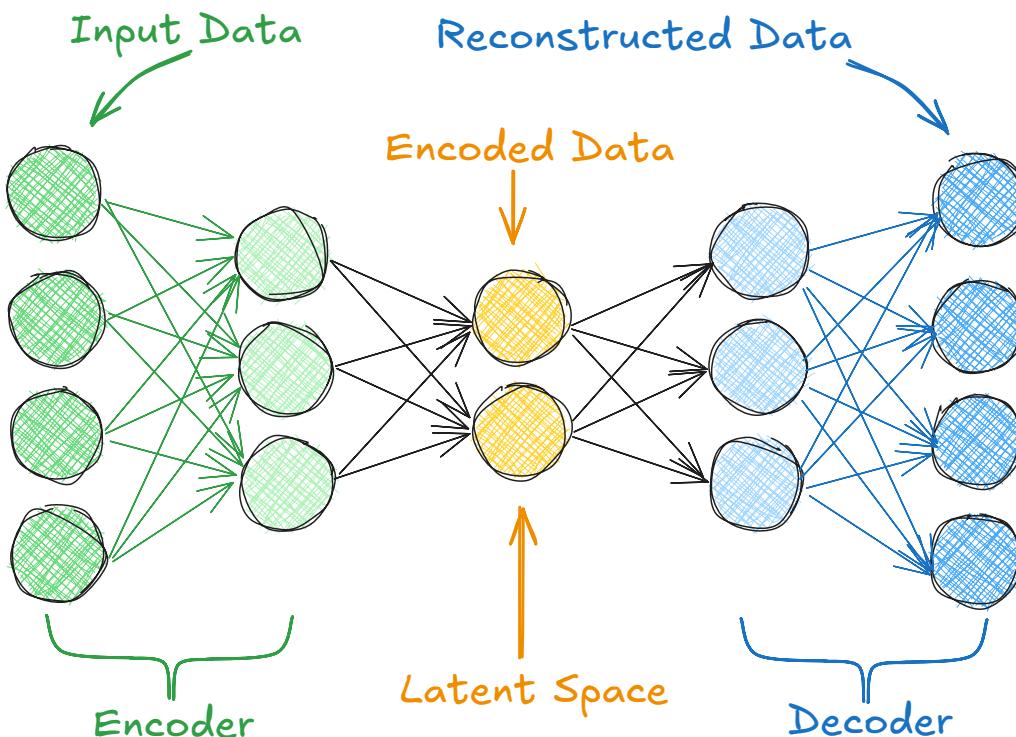
- **Hledání optimální rozhodovací nadroviny.** Jednotřídní SVM se nejprve snaží najít rozhodovací hranici, která odděluje většinu dat od ostatních bodů. Snahou algoritmu je, aby trénovací sada ležela celá v této hranici. Pokud jsou data oddělitelná pouze nelineárně, je možno využít tzv. **jader** (*Kernel*). Tato jádra slouží jako funkce, které mapují data do vyšších dimenzí, kde je možné data snáze oddělit na normální a anomální body výše dimenzionální nadrovinou.
- **Prostor normálních bodů.** Algoritmus trénuje model na datech, která jsou považována za normální, tj. nejsou to anomálie. Na základě těchto dat model určuje hranici, která obklopuje většinu bodů. Body ležící mimo hranici jsou považovány za anomálie.

Jednotřídní SVM umožňuje pracovat s nelineárně závislými daty a algoritmus je navíc velmi přesný, je-li předem známá definice anomálií – přítomnost anotací (*labelů*) [39].

**Lokální faktor odlehlosti (Local Outlier Factor, LOF).** *Lokální faktor odlehlosti* je metoda pro detekci anomálií, která hodnotí lokalitu každého bodu zvlášť ve vztahu k jeho sousedům. Algoritmus se zaměřuje na detekci bodů, které jsou vzhledem k sousedům neobvyklé a jiné, což znamená, že v porovnání se svými sousedy mají výrazně jinou hustotu bodů ve svém okolí. Algoritmus funguje na těchto principech [40]:

- **Lokální hustota** *Lokální faktor odlehlosti* nejprve spočítá hustotu okolí každého bodu na základě vzdálenosti mezi ním a jeho sousedy – obvykle se neberou v potaz data celá, ale jen předem zvolené k nejbližším sousedů.
- **Porovnání hustot** Algoritmus dále ohodnotí bod tím, jak se jeho lokální hustota liší od hustoty jeho sousedů. Pokud má bod výrazně nižší hustotu než jeho sousedé, je považován za anomálii.

Hlavní výhodou této metody je, že se zaměřuje na hustotu v lokálním okolí, což znamená, že lépe detekuje anomálie, které jsou izolované ve své lokalitě, ale mohou se tvářit normálně v širším globálnějším kontextu [40].



Obrázek 2.3: Základní model autoenkovéru

**Autoenkovdér (Autoencoder).** Autoenkovdér je typ neuronové sítě, který se skládá ze dvou hlavních částí: **enkodér** (*Encoder*) a **dekodér** (*Decoder*). Enkodér komprimuje vstupní data do nižší dimenze, do **latentního prostoru**, zatímco dekodér se pokouší obnovit původní vstupní data z této komprimované reprezentace. Na Obrázku 2.3 je jednoduchý náčrt autoenkovdéra, kde *enkodér* je znázorněn zelenou barvou, *dekodér* modrou a *latentní prostor* žlutou. Cílem autoenkovdérů je minimalizace rozdílu mezi původními vstupními daty a zrekonstruovanými výstupními daty. Využívá se v mnoha oblastech, včetně detekce anomálií, protože autoenkovdér se naučí rekonstruovat normální datové vzory a anomálie se obvykle nepodaří zrekonstruovat bezchybně – což je snadno zachytitelné [41].

V rámci práce byly zvoleny tři konkrétní architektury autoenkovdérů:

- **Dopředná neuronová síť** (*Feed Forward Neural Network, FFNN*) je nejjednodušší typ neuronové sítě, kde data procházejí jedním směrem od vstupu až po výstup, a to bez zpětných vazeb. Tato síť je tvořena vrstvami neuronů, kde je každá vrstva plně propojena s předchozí a následující vrstvou. Každý neuron v síti je spojen s váhami a používá aktivační funkci, která rozhoduje o výstupu neuronu [42].
- **Konvoluční neuronová síť** (*Convolutional Neural Network, CNN*) je typ neuronové sítě, který je navržen speciálně pro práci s mřížkovými daty, jako jsou obrázky nebo prostorově strukturovaná data. *Konvoluční síť* využívá *konvoluční vrstvy*, které aplikují filtry (*jádra*) na vstupní data, čímž extrahují lokální vzory, jako jsou hrany, textury, nebo jiné důležité rysy. Tyto vrstvy jsou následovány *poolingovými vrstvami*, které slouží ke zmenšení dimenze dat. V neposlední řadě *konvoluční sítě* využívají plně propojené vrstvy, které generují výstupy [42].
- **Transformer** je model, který byl původně navržen pro zpracování sekvencí, přičemž je založen na mechanismu nazývaném **Attention** (respektive **Self-Attention**). Tento mechanismus umožňuje modelu zaměřit se na různé části vstupní sekvence při zpracování dat, a to bez potřeby tradičních rekurentních struktur, jako je tomu tak u **LSTM** nebo **GRU** [43]. *Transformery* se staly populárními díky své efektivitě při zpracování sekvenčních dat, zejména ve **zpracování přirozeného jazyka** (*Natural Language Processing, NLP*). Autoenkovdery založené na transformerech jsou často používány pro analýzu sekvencí, jako jsou časové řady. Díky svému fungování jsou transformery schopny zachytit dlouhodobé závislosti v datech, což může být velmi užitečné při detekci anomálií.



# Implementovaná řešení

3

The computing scientist's main challenge is not to get confused by the complexities of his own making.

---

*Edsger W. Dijkstra (1930-2002), 1972 Turing Award recipient, Known for his work in algorithms and programming*

S rostoucí složitostí tržních dat a výzev spojených s jejich analýzou se zvyšuje i náročnost implementovaných řešení. Tato kapitola se zaměřuje na detailní popis implementace, která byla vyvinuta v rámci této práce. Cílem je ukázat, jak teoretické přístupy a metodologie, uvedené v předchozích kapitolách, byly aplikovány v praxi s cílem řešit konkrétní problém analýzy finančních trhů – detekci objednávek podezřelých ze *spoofingu*.

Tato kapitola se nejprve zaměřuje na přípravu dat a provedenou analýzu, která byla nezbytná pro následné modelování. Následuje popis implementovaných modelů, včetně detailů jejich architektur, a nakonec jsou uvedeny použité metody evaluace.

**Použité technologie.** Pro implementaci byl zvolen jazyk **Python** ve vybrané verzi **Python 3.11**. Prostředí **Python** je široce používané pro analýzu dat a strojové učení.

V **Pythonu** je běžné využívat různé knihovny pomocí standardního **správce balíčků pip**. Knihovny využité v této práci usnadňují analýzu dat, strojové učení a tvorbu vizualizací. Mezi hlavní použité knihovny, které pomohly urychlit implementaci a optimalizovat procesy analýzy a modelování, patří:

- **NumPy**<sup>1</sup> – knihovna pro efektivní práci s pokročilými matematickými operacemi a více dimenzionálními poli.
- **Pandas**<sup>2</sup> – knihovna pro manipulaci a analýzu dat poskytující strukturované tabulky pro snadnou práci s časovými řadami.
- **Matplotlib**<sup>3</sup> a **Plotly**<sup>4</sup> – knihovny pro vizualizace dat a vytváření interaktivních grafů.
- **Scikit-learn**<sup>5</sup> – knihovna pro základní strojové učení obsahující implementace jednoduchých algoritmů pro klasifikace, regrese, shlukování, atp.
- **PyTorch**<sup>6</sup> – knihovna pro pokročilé strojové učení a hluboké učení – využívá se pro implementaci neuronových sítí a jejich následný trénink
- **WandB**<sup>7</sup> – nástroj pro sledování a monitorování experimentů a modelů strojového učení.

V neposlední řadě byl využit verzovací systém **git** pro efektivní správu a verzování kódu. Celý projekt je veřejně dostupný na platformě **GitHub** na adrese [https://github.com/SpeekeR99/DP\\_2024\\_2025\\_Zappe](https://github.com/SpeekeR99/DP_2024_2025_Zappe)

## 3.1 Data

V rámci práce byla poskytnuta reálná historická data německou společností **Deutsche Börse AG** v rámci Smlouvy o poskytnutí dat pro akademické účely ze dne 14. 11. 2022 ve formátu **PCAP**<sup>8</sup> souborů, pokrývajících celý rok 2021. Tato data představují surový síťový provoz z burzovního serveru, který je nutné následně dekódovat a transformovat do strukturovanější podoby pro další analýzu – tento proces se dále označuje jako **rekonstrukce** dat. Smlouva nedovoluje zveřejnění těchto dat (*Non-Disclosure Agreement*), nicméně umožňuje hodnotitelům a oponentům této práce zpřístupnit na vyžádání tato data na dobu nezbytnou pro vypracování příslušného hodnocení.

---

<sup>1</sup><https://pypi.org/project/numpy/>

<sup>2</sup><https://pypi.org/project/pandas/>

<sup>3</sup><https://pypi.org/project/matplotlib/>

<sup>4</sup><https://pypi.org/project/plotly/>

<sup>5</sup><https://pypi.org/project/scikit-learn/>

<sup>6</sup><https://pypi.org/project/torch/>

<sup>7</sup><https://pypi.org/project/wandb/>

<sup>8</sup>PCAP (*Packet Capture*) je formát souboru používaný pro zachytávání a ukládání síťového provozu. Umožňuje detailní analýzu přenášených dat na úrovni jednotlivých paketů.

Dále byl v rámci práce získán přístup k nástroji **A7** (Advanced Order Book), který je blíže popsán v kapitole 2.1.7. Tento proprietární nástroj umožňuje stahování historických dat prostřednictvím rozhraní *API*, přičemž výsledná data jsou dostupná ve formátu *JSON*. Nástroj nabízí dvě základní *API* rozhraní – surové zprávy (obdoba *PCAP* souborů) a zrekonstruovanou knihu limitních objednávek. Vzhledem k tomu, že v rámci této práce dochází k vlastní **rekonstrukci** dat, bylo primárně využíváno rozhraní poskytující surové zprávy. Díky nim je možné získat podrobnější informace, které v samotné knize limitních objednávek již nemusí být dostupné, což je klíčové pro pozdější analýzu a detekci anomálií.

Pro potřeby analýzy byly vybrány především tzv. **Triple Witching Days** – třetí pátky v *březnu*, *červnu*, *září* a *prosinci*, kdy dochází k současné expiraci různých typů *derivátových kontraktů*, zejména **opcí na akcie** a **opcí a futures na akciové indexy**. **Opce** (z anglického *Options*) jsou finanční nástroje, které dávají jejich držiteli právo, nikoli však povinnost, kupit nebo prodat určité aktivum za předem stanovenou cenu k určitému datu nebo před ním. **Futures** jsou naopak závazné kontrakty, které účastníky zavazují k nákupu nebo prodeji aktiva za pevně danou cenu k budoucímu datu. *Witching Days* jsou známé zvýšenou volatilitou a obchodní aktivitou, což z nich činí zajímavý subjekt pro sledování možných manipulativních praktik, jako je *spoofing*.

### 3.1.1 Předzpracování dat

Hlavním cílem fáze předzpracování bylo převést surové zprávy získané prostřednictvím **A7 API** do formátu *LOBSTER*, který je podrobněji popsán v Kapitole 2.1.5. Tento formát strukturuje knihu limitních objednávek do diskrétních časových kroků, přičemž pro každou časovou značku je zaznamenáno několik úrovní hloubky trhu. V rámci této práce, na základě konzultace s odborníkem, bylo zvoleno zpracovávat **30 cenových úrovní (Levels)** na každé straně knihy – nabídky i poptávky.

Díky tomu, že vstupem předzpracování byly surové zprávy, a nikoliv již zrekonstruovaná kniha objednávek, bylo možné při transformaci spočítat i doplňující metriky. Mezi tyto dodatečně vypočítané informace patří:

- Počet zrušených objednávek v daném časovém okamžiku – zvlášť pro *nákupní* a *prodejní* stranu.
- Počet realizovaných obchodů – opět zvlášť pro *nákupní* a *prodejní* stranu.

Tento obohacený datový formát poskytuje důležitý kontext pro detekci anomálního chování a výrazně zvyšuje informační hustotu záznamů pro následnou extrakci dalších metrik a příznaků.

Vlastní skript provádějící tuto popsanou transformaci se nachází ve složce /src/data\_preprocess pod názvem `json-detailed2lobster.py`.

### 3.1.2 Extrakce příznaků

Po fázi předzpracování je výsledná *zrekonstruovaná* kniha načtena zpět do paměti, a následně je druhým průchodem obohacena o další metriky a příznaky (*Features*). Tento krok je realizován skriptem `augment_lobster.py`, který se nachází ve stejné složce jako předchozí skript.

Tento proces extrakce příznaků je klíčový pro následnou analýzu a modelování, protože poskytuje cenné ukazatele, které mohou pomoci identifikovat anomální chování na finančních trzích – *spoofing*.

Mezi extrahované metriky patří následující:

**Imbalance Index.** Tato metrika vyjadřuje nerovnováhu mezi nabídkou a poptávkou na trhu. V kontextu *spoofingu* je důležité, že manipulátor může vytvářet dojem silného zájmu o nákup nebo prodej pomocí většího množství objednávek na jedné straně knihy. Tento index tedy sleduje, zda existuje výrazná nerovnováha mezi nákupními a prodejnými objednávkami, což je typické pro manipulativní strategie [44]. *Imbalance Index*  $II(t)$  je definován vztahem:

$$II(t) = \frac{V_b(t) - V_a(t)}{V_b(t) + V_a(t)}, \quad (3.1)$$

kde  $V_b(t)$  je vážený objem nabídek (*Bids*) a  $V_a(t)$  je vážený objem poptávek (*Asks*).

Pro výpočet  $V_b(t)$  a  $V_a(t)$  lze použít:

$$V_b(t) = \sum_{i=1}^L V_{b,i}(t) \cdot e^{-\alpha \cdot i}, \quad (3.2)$$

$$V_a(t) = \sum_{i=1}^L V_{a,i}(t) \cdot e^{-\alpha \cdot i}, \quad (3.3)$$

kde  $L$  je počet úrovní (v implementaci zvoleno  $L=30$ ),  $V_{b,i}(t)$  a  $V_{a,i}(t)$  jsou objemy na  $i$ -té cenové úrovni nabídky (*Bid*) a poptávky (*Ask*) a  $\alpha$  je parametr pro vážení úrovní (v implementaci zvoleno  $\alpha = 0.5$ ).

**Frekvence příchozích zpráv.** Tento příznak sleduje frekvenci přicházejících zpráv na trh v určitém časovém intervalu. Měří aktivitu na trhu a vyjadřuje, jak často dochází k novým příkazům – přidání, vypořádání, modifikace a rušení objednávek. Vyšší frekvence zpráv může naznačovat agresivní manipulaci s trhem. Tento příznak je vypočítán pomocí *klouzavého průměru* s oknem  $o$  velikosti 5 minut, čímž se zohledňuje krátkodobá volatilita. Formální definice počtu zpráv  $n(t_i)$  v časovém okně  $\langle t_i - o, t_i \rangle$  je:

$$n(t_i) = \sum \mathbb{I}_{\{t_j \in \langle t_i - o, t_i \rangle\}}, \quad (3.4)$$

kde  $\mathbb{I}$  je indikátor, který je roven 1, pokud  $t_j$  spadá do intervalu  $\langle t_i - o, t_i \rangle$ , jinak 0 – je zde využito toho, že v knize jsou pouze časové značky, při kterých nějaká zpráva přišla. Parametr  $o$  je pak možné volit, v implementaci je zvoleno 5 minut.

Převod na frekvenci pro daný časový okamžik je pak:

$$f(t_i) = \frac{n(t_i)}{o}. \quad (3.5)$$

**Míra zrušených objednávek.** Jedná se o podobný ukazatel jako je výše popsaná obecná frekvence *všech* příchozích zpráv. V případě *spoofingu* manipulátor často zruší objednávky, které nikdy neměly být vykonány, aby vytvořil falešný dojem o hloubce trhu. Tato míra zrušených objednávek se vypočítává na základě informací z fáze předzpracování, která zahrnovala detekci a sledování zrušení objednávek. Pro výpočet této metriky je opět užit *klouzavý průměr* s oknem  $o$  velikosti 5 minut – výpočet se řídí výše uvedenými vzorcemi (3.4) a (3.5), pouze je pozměněna indikátorová funkce, kdy 1 se rovná pouze tehdy, je-li v daném časovém okamžiku záZNAM o zrušení objednávky.

**„High Quoting Activity“.** Tato metrika odráží vysokou aktivitu v nabídce a poptávce na trhu. Indikuje množství limitních objednávek, které jsou na trhu přítomné, ale nevyplněné. V případě *spoofingu* manipulátor používá velké objemy limitních objednávek, aby vytvořil dojem silného zájmu o danou cenu, což může ovlivnit rozhodování ostatních účastníků trhu [45]. Metrika je definována následujícím vztahem:

$$HQ_s = \max_{t \in s} \frac{|EntryAskSize_t - EntryBidSize_t|}{AskSize_t + BidSize_t}, \quad (3.6)$$

kde  $\text{EntryAskSize}_t$  je nárůst celkového objemu objednávek na pěti nejvyšších cenových úrovních poptávky (*Ask*) v čase  $t$  (pokud nedošlo k nárůstu, hodnota je rovna 0); obdobně  $\text{EntryBidSize}_t$  je nárůst celkového objemu objednávek na pěti nejvyšších cenových úrovních nabídky (*Bid*) v čase  $t$  (opět – pokud nedošlo k nárůstu, hodnota je rovna 0).  $\text{AskSize}_t$  je celková hloubka (agregovaný objem objednávek) na pěti nejvyšších cenových úrovních poptávky (*Ask*) v čase  $t$ ; obdobně  $\text{BidSize}_t$  je celková hloubka (agregovaný objem objednávek) na pěti nejvyšších cenových úrovních nabídky (*Bid*) v čase  $t$ .  $t$  označuje konkrétní časovou značku,  $s$  je 1-sekundový interval. Obecně lze uvažovat i časové intervaly různé délky, např. i jeden den [45].

**„Unbalanced Quoting“.** Tento indikátor se zaměřuje na zjištění, kdy jsou na trhu přítomny objednávky v nerovnováze mezi nabídkou a poptávkou. Pokud je více objednávek na jedné straně než na druhé (například více nákupních objednávek než prodejních), může to být známka manipulace [45]. Tato metrika je podobná *Imbalance Indexu*, ale sleduje nerovnováhu na méně úrovních a rozsah jejích hodnot není od -1 do 1, ale pouze od 0 do 1. Vztah (3.7) definuje tento indikátor:

$$\text{UQ}_s = \max_{t \in s} \frac{|\text{AskSize}_t - \text{BidSize}_t|}{\text{AskSize}_t + \text{BidSize}_t}, \quad (3.7)$$

kde  $\text{AskSize}_t$  je kumulativní hloubka (agregovaný objem objednávek) na pěti nejvyšších cenových úrovních poptávky (*Ask*) v čase  $t$ ; obdobně  $\text{BidSize}_t$  je kumulativní hloubka na pěti nejvyšších cenových úrovních nabídky (*Bid*) v čase  $t$ ,  $t$  označuje konkrétní časovou značku,  $s$  je opět 1-sekundový interval.

**„Low Execution Probability“.** Tento příznak se zaměřuje na situace, kdy jsou objednávky umístěny na trhu, ale je velmi nízká pravděpodobnost jejich realizace. Objednávky mohou být umístěny daleko od nejlepší ceny nebo na zadních pozicích v dlouhých frontách objednávek. Manipulátor provozující *spoofing* nevkládá objednávky s cílem je realizovat, ale spíše je používá pro vytvoření falešného dojmu o trhu [45] Příznak je definován jako:

$$\text{LE}_s = \max_{t \in s} \frac{|\text{AskSizeLevel2to5}_t - \text{BidSizeLevel2to5}_t|}{\text{AskSize}_t + \text{BidSize}_t}, \quad (3.8)$$

kde  $\text{AskSizeLevel2to5}_t$  je kumulativní hloubka na cenových úrovních 2 až 5 poptávky (*Ask*) v čase  $t$ ; obdobně  $\text{BidSizeLevel2to5}_t$  je kumulativní hloubka na cenových úrovních 2 až 5 nabídky (*Bid*) v čase  $t$ .  $\text{AskSize}_t$  je kumulativní hloubka na pěti nejvyšších cenových úrovních poptávky (*Ask*) v čase  $t$ ; obdobně  $\text{BidSize}_t$  je

kumulativní hloubka na pěti nejvyšších cenových úrovních nabídky (*Bid*) v čase  $t$ .  $t$  označuje konkrétní časovou značku,  $s$  je 1-sekundový interval.

**Trades Oppose Quotes.** Manipulátoři často obchodují v opačném směru, než naznačuje jejich objednávková nerovnováha. Například pokud mají více nákupních objednávek než prodejných, mohou provádět prodeje, aby vytvořili dojem opačného chování na trhu. Tento příznak může být klíčový pro identifikaci, kdy obchodování probíhá v rozporu s indikovaným směrem, který by měl být vyvolán objednávkovou nerovnováhou [45]. Metrika je definována vztahem:

$$TOQ_t = \begin{cases} 1 & \text{if } II(t-s) < -10\% \text{ and } Trade_t^{\text{Bid}} = 1 \\ 1 & \text{if } II(t-s) > 10\% \text{ and } Trade_t^{\text{Ask}} = 1 \\ 0 & \text{jinak,} \end{cases} \quad (3.9)$$

kde  $Trade_t^{\text{Bid}}$  je roven 1 právě tehdy, když dojde k obchodu na nabídkové (*Bid*) straně během  $\langle t-s, t \rangle$ ; obdobně  $Trade_t^{\text{Ask}}$  je roven 1 právě tehdy, když dojde k obchodu na poptávkové (*Ask*) straně během  $\langle t-s, t \rangle$ .  $II(t-s)$  označuje výše popsaný *Imbalance Index* v časovém okamžiku před  $t$ , tedy v čase  $t-s$ .  $t$  označuje konkrétní časovou značku,  $s$  je 1-sekundový interval.

**Cancels Oppose Trades.** Tato metrika ukazuje na situace, kdy manipulátor zruší objednávky na jedné straně trhu poté, co dojde k realizaci obchodu na straně opačné. Například po nákupu může manipulátor zrušit prodejní objednávky, což ukazuje, že objednávky nebyly skutečně určeny k realizaci obchodu [45]. Příznak je definován jako:

$$COQ_t = \begin{cases} 1 & \text{if } CR_{t-s}^{\text{Ask}} > threshold \text{ and } Trade_t^{\text{Bid}} = 1 \\ 1 & \text{if } CR_{t-s}^{\text{Bid}} > threshold \text{ and } Trade_t^{\text{Ask}} = 1 \\ 0 & \text{jinak,} \end{cases} \quad (3.10)$$

kde  $Trade_t^{\text{Bid}}$  je roven 1 právě tehdy, když dojde k obchodu na nabídkové (*Bid*) straně během  $\langle t-s, t \rangle$ ; obdobně  $Trade_t^{\text{Ask}}$  je roven 1 právě tehdy, když dojde k obchodu na poptávkové (*Ask*) straně během  $\langle t-s, t \rangle$ .  $CR_{t-s}^{\text{Bid}}$  označuje výše popsanou *Míru zrušených objednávek*, ale pouze na nabídkové (*Bid*) straně, v časovém okamžiku před  $t$ , tedy v čase  $t-s$ ; obdobně  $CR_{t-s}^{\text{Ask}}$  označuje *Míru zrušených objednávek*, ale pouze na poptávkové (*Ask*) straně, v časovém okamžiku před  $t$ , tedy v čase  $t-s$ .  $threshold$  je práh definovaný jako 10. percentil celé *Míry zrušených objednávek*.  $t$  označuje konkrétní časovou značku,  $s$  je 1-sekundový interval.

### 3.1.3 Redukce dimenze dat

V rámci této práce byla provedena **redukce dimenze vstupních dat**, která je klíčovým krokem pro zjednodušení modelu a zlepšení jeho výkonnosti. Hlavním cílem redukce dimenze je odstranit redundantní nebo málo relevantní příznaky, což vede ke snížení složitosti modelu a zároveň může přispět k urychlení procesu trénování a zlepšení přesnosti modelu.

Pro účely redukce dimenze byla vybrána technika **PCA (Principal Component Analysis)**. Nicméně, v rámci této implementace nebyl algoritmus aplikován na celý dataset, ale pouze na *strategicky vybrané* kombinace příznaků, které vykazují silnou koreaci. Korelující příznaky byly sloučeny do jedné **hlavní komponenty** užitím PCA, což pomáhá zachovat interpretovatelnost modelu. Tento přístup minimalizuje ztrátu interpretovatelnosti, která by jinak mohla nastat při aplikaci PCA na celý dataset. Korelující komponenty obvykle sdílejí podobný význam a výpočty, což znamená, že výsledná komponenta stále zachovává značnou míru interpretovatelnosti.

Tato a níže popsané analýzy jsou součástí skriptu `feature_selection.py` ve složce `/src/anomaly_detection/analysis`. Implementace vizualizačních částí jednotlivých analýz jsou k nalezení ve skriptu `visuals.py` ve stejném adresáři.

### Korelační analýza

V rámci další části analýzy dat byla provedena **korelační analýza** na všech vstupních příznacích (*Features*). Korelační analýza je technika používaná k odhalení vztahů a vzorců mezi jednotlivými proměnnými v datasetu. Pomocí korelační matice je možné zjistit, jak silně jsou jednotlivé příznaky navzájem propojeny. Silná korelace mezi dvěma příznaky může signalizovat, že některé příznaky jsou redundantní, což může vést k nežádoucí složitosti modelu.

Korelační analýza slouží k několika účelům:

1. Zjištění lineárních vztahů mezi příznaky. Pomocí korelačních matic lze zjistit, jak silně jsou jednotlivé příznaky navzájem propojeny. Pokud je mezi dvěma příznaky vysoká korelace, může to znamenat, že některé příznaky jsou redundantní a mohou být odstraněny, aby se zjednodušil budoucí model.
2. Detekce multikolinearity. Vzhledem k tomu, že některé příznaky (např. *Imbalance Index*, *Unbalanced Quoting*, *Low Execution Probability*) mají podobné výpočty, mohou vzájemně silně korelovat.

## Důležitost příznaků

Pro určení **důležitosti příznaků** (*Feature Importance*) byl v této práci použit algoritmus **DIFFI** (*Depth-based Isolation Forest Feature Importance*), který je speciálně navržen pro *izolační lesy* (*Isolation Forest*), ale i tak vykazuje obecně velmi dobré výsledky v analýze důležitosti příznaků. *Izolační les* je robustní algoritmus pro detekci anomálií, který efektivně identifikuje odlehle hodnoty, což činí tento přístup užitečný i pro zjištění, jak moc jednotlivé příznaky přispívají k modelu.

Pro zajištění stabilních výsledků byla analýza provedena opakováně, konkrétně stokrát, aby se eliminovala náhodná složka a získaly stabilní odhady důležitosti příznaků. Výsledky analýzy jsou následně použity k identifikaci klíčových příznaků, které mají největší vliv na predikci anomálií.

Implementace je založena na veřejně přístupné knihovně <https://github.com/britojr/diffi.git>, která je specificky zaměřena na výpočet důležitosti příznaků v kontextu implementace *izolačních lesů* od **Scikit-learn**.

## 3.2 Strojové učení

Na základě poznatků z předchozí diplomové práce [46] bylo v této práci rozhodnuto upřednostnit využití metod strojového učení před klasickými statistickými přístupy. Důvodem je skutečnost, že analyzovaná data nevykazují zjevnou nebo konzistentní pravděpodobnostní distribuci, címž dochází k omezení použitelnosti tradičních modelů, jako je například **ARIMA**, které na podobných předpokladech závisejí. Z tohoto důvodu se jako vhodnější jeví nasazení moderních přístupů využívajících metod umělé inteligence.

### 3.2.1 Vybrané modely

Moderní přístupy založené na strojovém učení, především ty využívající *neuronové sítě*, umožňují modelovat složité a nelineární vztahy mezi vstupními příznaky. Tyto schopnosti jsou zvláště důležité v úlohách detekce anomálií, kde často dochází k porušení standardních předpokladů normality nebo lineárnosti. Z tohoto důvodu byla jako hlavní metoda pro detekci anomálií zvolena architektura *autoenkovému*.

Pro účely porovnání výkonnosti byl kromě autoenkovému implementován také *referenční model* ve formě **izolačního lesu** (*Isolation Forest, IF*), jenž představuje efektivní stromovou metodu pro detekci odlehlych hodnot. Dále byly zařazeny i dva další tradiční modely pro detekci anomálií, a to **lokální faktor odlehlosti**

(**Local Outlier Factor, LOF**) a **jednotřídní SVM (One-Class SVM, OCSVM)**. Tyto modely slouží jako srovnávací základ (*Baseline*) při vyhodnocování přínosu hlubokých neuronových sítí.

Teoretický základ všech uvedených metod je detailně popsán v Kapitole 2.4.1. Všechny implementace jsou dostupné ve složce `/src/anomaly_detection/models`.

## Referenční modely

Implementace klasických modelů *IF*, *LOF* a *OCSVM* je realizována ve skriptu `if_ocsvm_lof.py`. Tyto implementace přímo vycházejí z knihovny **Scikit-learn**. Po načtení dat, v souladu s popisem v Kapitole 3.1, je vytvořen zvolený model, následně natrénován a aplikován na testovací množinu. Výsledky klasifikace a skóre jednotlivých vzorků jsou následně uloženy do souboru pro další analýzu a vizualizaci. Rovněž jsou ukládány i samotné natrénované modely pro možné opakované použití.

## Modely založené na hlubokém učení

Hlavní část výzkumu je zaměřena na modely založené na hlubokém učení, konkrétně na *autoenkodéry*. Ty byly implementovány ve skriptu `autoencoder.py` pomocí frameworku **PyTorch**. Skript obsahuje implementaci společné základní třídy `BaseAutoencoder`, která je navržena v duchu rozhraní, které je inspirováno klasickými modely ze **Scikit-learn**, tedy definuje metody `fit`, `score_samples` a `decision_function`. Mimo jiné samozřejmě standardní metodu `forward`, která aplikuje kódování a dekódování na data pomocí příslušných podsítí – *enkodér* a *dekodér*.

Z této základní třídy dědí konkrétní varianty autoenkodérů:

- `FFNNAutoencoder` – plně propojená (Feedforward Neural Network) architektura.
- `CNNAutoencoder` – konvoluční autoenkodér.
- `TransformerAutoencoder` – architektura založená na transformerech.

Pro funkčnost *transformer* architektury byla rovněž implementována vrstva `PositionalEncoding`, která dodává informaci o pořadí prvků v sekvenci (*Sequence*). Vzhledem k povaze konvolučních sítí a transformerů bylo nezbytné vstupní data převést do formy sekvencí, s čímž souvisí i fáze zpětného mapování

detekovaných anomálií z výstupu modelu do konkrétních časových bodů. Tento proces je detailně zpracován ve skriptu `/src/anomaly_detection/data/sequences.py`.

Výsledky trénování autoenkodérů, stejně jako jejich váhy a struktury, jsou opět ukládány pro pozdější analýzu a reprodukovatelnost výpočtu.

### 3.2.2 Trénování

Proces trénování modelů představuje klíčovou fázi návrhu systému pro detekci anomálií. V rámci této práce byla navržena a implementována vlastní trénovací smyčka, která zohledňuje specifika dat i zvolených architektur. Veškerá implementace týkající se trénování modelů se nachází ve skriptu `training.py` ve složce `/src/anomaly_detection/models`.

Trénovací proces je založen na principu, že každý den a každý produkt je zpracováván zvlášť. Tento přístup je odůvodněn skutečností, že chování dat se napříč jednotlivými dny a produkty výrazně liší – jak z pohledu volatility, tak z pohledu distribuce příznaků. V důsledku této variability by použití globálního testovacího datasetu nebylo relevantní a mohlo by vést k nesprávným závěrům o schopnostech modelů. Z toho důvodu je trénování provedeno opakováně – zvlášť pro každý den a produkt.

Trénování referenčních modelů *IF*, *LOF* a *OCSVM* je poměrně přímočaré. Vzhledem k jejich implementaci v knihovně **Scikit-learn** postačuje standardní postup.

Trénování hlubokých neuronových modelů je podstatně komplexnější. V rámci této práce byla implementována vlastní trénovací smyčka v rámci metody `fit()` ve skriptu `autoencoder.py`. Jako ztrátová funkce byla zvolena **střední kvadratická odchylka** (*Mean Squared Error*, *MSE*), která slouží nejen jako trénovací metrika, ale rovněž jako míra anomálnosti – čím vyšší je *rekonstrukční chyba* pro daný vzorek, tím pravděpodobněji se jedná o anomálii.

### Modifikovaná K-Fold křížová validace

Přestože rozsah dat nevyžaduje nasazení technik jako je **K-Fold křížová validace** (*K-Fold cross-validation*) (typicky používané v případech s nedostatkem dat), byl tento přístup využit a modifikován pro specifické potřeby práce. Byl použit klasický *K-Fold* s  $k = 5$ , tedy rozdělení dat na 80 % trénovacích a 20 % validačních vzorků. Každý *fold* tedy obsahuje jinou část dat jako validační množinu, přičemž výsledky z každého z nich jsou po trénování sloučeny.

Tento mechanismus byl využit pro vytvoření robustní predikce pro celý den – každá část dat figuruje právě jednou ve validačním setu a modely jsou tak schopné pokrýt celý den v pěti iteracích. Po skončení všech pěti *foldů* jsou výstupy agregovány, čímž je získána predikce pro celý den. Hlavní nevýhodou však zůstává, že modely je nutné trénovat vždy znova – pro každý den i produkt zvlášť.

## Optimalizace trénovacího procesu

Pro dosažení stabilní a efektivní konvergence modelů byly při trénování neuronových sítí využity pokročilé optimalizační techniky:

- **Learning Rate Scheduler** – adaptivní řízení *učící konstanty* (*Learning Rate*) umožňuje dynamické přizpůsobení rychlosti učení v průběhu trénování. Tím se zvyšuje šance na dosažení globálního optima a snižuje riziko přeskakování optimálních řešení.
- **Early Stopping** – aby se předešlo *přetrénování* (*Overfitting*), je implementován mechanismus včasného ukončení trénování. Pokud validační chyba nevykazuje zlepšení po dobu 10 epoch, je trénování ukončeno a načtou se parametry modelu z epochy s nejlepší dosaženou validací.
- **Monitorování metrik** – trénovací smyčka průběžně sleduje hodnoty ztrátové funkce na trénovacím i validačním datasetu. Nejlepší model je vždy uchován na základě minimální validační chyby.

Tato kombinace metod přispívá k efektivnímu trénování a zároveň zajišťuje, že výsledné modely mají dobrou schopnost generalizace a odolnost vůči přetrénování.

## MetaCentrum a Weights & Biases

Pro efektivní a systematické sledování průběhu trénování modelů byl využit nástroj **Weights & Biases (WandB)**. Tento nástroj umožnil podrobné logování průběhu učení jednotlivých modelů, včetně záznamu metrik, vizualizací a hyperparametrů. Pomocí **WandB** bylo rovněž realizováno rozsáhlé *prohledávání prostoru hyperparametrů* (*Grid Search*), jehož cílem bylo nalézt optimální konfigurace modelů – jak v případě referenčních modelů, tak hlubokých neuronových architektur.

### Prohledávání prostoru hyperparametrů.

- **Scikit-learn modely** – celkem bylo testováno 31 různých kombinací hyperparametrů.
- **PyTorch autoencodery** – hluboké modely byly laděny v rámci 135 různých kombinací hyperparametrů.
- Všechny varianty byly ještě jednou spuštěny i nad daty s redukovanou dimenzionalitou, čímž se počet experimentů dále násobil.

Celkem bylo vygenerováno a zpracováno více než **2 800** unikátních úloh (včetně testovacích a pokusných běhů), přičemž výpočetní náročnost dosáhla celkově přibližně **250 CPU dny**.

**Výpočetní infrastruktura.** Z důvodu výpočetní náročnosti nebylo možné tyto experimenty realizovat na osobním zařízení. Jako hlavní výpočetní prostředí bylo proto zvoleno **MetaCentrum**, které poskytlo potřebné kapacity pro paralelní výpočty ve velkém měřítku. Díky modularité implementace bylo možné efektivně distribuovat jednotlivé běhy napříč distribuovanými uzly. Skripty pro spuštění úloh v rámci **MetaCentra** jsou dostupné v adresáři `/src/anomaly_detection/meta_centrum`.

Všechny běhy byly propojeny sinstancemi **WandB**, kam byly průběžně ukládány všechny důležité informace – volby hyperparametrů a konfigurace modelů, vývoj parametrů modelu, průběhy ztrátových funkcí v čase, trénovací a validační metriky (tém se bude podrobně věnovat následující Kapitola 3.2.3). Tato detailní dokumentace následně sloužila jako základ pro výběr nejlepších modelů.

## 3.2.3 Evaluace

Jelikož se v rámci této práce pracuje s *neannotovanými* daty, jedná se o problém *učení bez učitele* (*Unsupervised Learning*). To znamená, že není k dispozici žádná *pevná pravda* ohledně toho, které vzorky jsou skutečně anomální. Z tohoto důvodu nelze přímo použít klasické metriky jako *přesnost* (ať už ve smyslu *Accuracy*, tak i *Precision*), *úplnost* (*Recall*) či *F1 skóre*, které předpokládají existenci správného označení tříd.

### Metriky použité pro evaluaci.

- **Rekonstrukční chyba (MSE)** – pro modely založené na autoenkodérech byla jako hlavní evaluační metrika použita jejich ztrátová funkce – tedy *střední kvadratická chyba (Mean Squared Error, MSE)*. Tato chyba udává, jak přesně je model schopen rekonstruovat původní vstupní data. Předpokládá se, že anomální vzorky budou mít vyšší chybu rekonstrukce, neboť se liší od běžného rozložení trénovacích dat.
- **Excess-Mass, Mass-Volume křivky** – pro účely porovnání výkonnosti napříč všemi modely (včetně *IF*, *LOF*, *OCSVM* i neuronových architektur) byly použity tzv. ***Excess-Mass*** a ***Mass-Volume*** metriky. Jedná se o méně rozšířený, alternativní způsob evaluace pro neanotovaná data, který byl popsán v práci [47] a jehož teoretické základy pochází z prací [48, 49].

### Chyba rekonstrukce autoenkodéru

U autoenkodérů je klíčovou metrikou ztrátová funkce – v tomto případě rekonstrukční chyba měřená pomocí *MSE*, definovaná jako:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \quad (3.11)$$

kde  $x_i$  je původní vstupní hodnota a  $\hat{x}_i$  je výstup dekodéru pro daný vstup. Vyšší hodnota chyby značí, že model nebyl schopen daný vzorek přesně rekonstruovat, což může indikovat jeho odlišnost od běžných vzorků – tedy potenciální anomálii.

Pro každý den a produkt je výsledkem vektor skóre anomálnosti, kde vyšší hodnota odpovídá vyšší pravděpodobnosti, že daný vzorek je anomální.

### Excess-Mass, Mass-Volume křivky

Vzhledem k absenci anotací bylo nutné zvolit metriky nezávislé na konkrétním prahu rozhodnutí nebo *pevné pravdě (Ground Truth)*. K tomuto účelu byly použity ***Excess-Mass*** a ***Mass-Volume*** křivky, které poskytují informaci o tom, jak dobře je model schopen separovat normální a anomální data na základě rozložení skóre.

- ***Excess Mass (EM)*** měří, jak efektivně dokáže model koncentrovat velkou část dat v oblastech s vysokou hustotou (malý objem prostoru). Vyšší *EM* pak znamená, že většina normálních dat se nachází v kompaktní oblasti, což

je žádoucí vlastnost dobrého detektoru anomalií. Formálně je tato metrika definována následovně:

$$\text{EM}_s(\lambda) = \sup_{u \geq 0} \mathbb{P}(s(\mathbf{X}) \geq u) - \lambda \text{Leb}(s \geq u), \quad (3.12)$$

kde  $s \in S$  je ohodnocovací funkce a  $\mathbb{P}(s(\mathbf{X}) \geq u)$  udává pravděpodobnost, že skóre náhodné proměnné  $\mathbf{X}$  přesáhne práh  $u$ , zatímco  $\text{Leb}(s \geq u)$  značí *Lebesgueovu míru* oblasti, kde skóre přesahuje  $u$ . Pravděpodobnost  $\lambda$  představuje penalizaci za prostorový objem.

- **Mass Volume (MV)** sleduje, jaký objem prostoru je potřeba k pokrytí určitého procenta dat. Nižší hodnota MV pak znamená, že normální data jsou lépe shlukována a detektor je schopný identifikovat anomálie jako odlehlé body. Metriku definuje vztah:

$$\text{MV}_s(\alpha) = \inf_{u \geq 0} \text{Leb}(s \geq u) \quad \text{takové, že } \mathbb{P}(s(\mathbf{X}) \geq u) \geq \alpha, \quad (3.13)$$

kde  $s \in S$  je opět ohodnocovací funkce a  $\mathbb{P}(s(\mathbf{X}) \geq u)$  vyjadřuje pravděpodobnostní podíl dat ležících nad prahem  $u$ , a  $\text{Leb}(s \geq u)$  odpovídá objemu oblasti, kde skóre tuto hranici překračuje. Cílem je nalézt oblast minimálního objemu, která pokrývá alespoň  $\alpha$  (procentuální) část.

Ačkoliv se tyto metriky v praxi nepoužívají běžně, práce [47] ukázala, že jejich výpovědní hodnota je srovnatelná s klasickými metrikami jako ROC-AUC či F1 skóre v případech, kde jsou dostupné anotace. Proto byly tyto metriky zvoleny jako vhodný nástroj pro evaluaci v kontextu této práce.

**Implementace.** Pro výpočet EM/MV křivek byla jako základ zvolena oficiální implementace z repozitáře autora článku [47], dostupná na adrese [https://github.com/ngoix/EMMV\\_benchmarks.git](https://github.com/ngoix/EMMV_benchmarks.git).

Tato implementace však byla nekompatibilní s novějšími verzemi **Pythonu 3**, a proto byla přepracována a upravena. Upravená verze se nachází v adresáři `src/anomaly_detection/eval` v souboru `em.py`. Aktualizovány byly i oba originální demonstrační příklady od autora, které jsou ve stejné složce. Pro komplexní evaluaci modelů, jaká byla potřebná v této práci, byl vytvořen skript `eval.py`, rovněž dostupný ve zmíněném adresáři.

## 3.3 Vizuální analýza

Ačkoliv je hlavní téžiště této práce zaměřeno na automatizované metody detekce anomálií pomocí modelů strojového učení, nezastupitelnou roli zde hraje také vizuální analýza. Cílem této kapitoly je představit vyvinutý interaktivní vizualizační nástroj.

V této kapitole je popsán princip a implementace nástroje, jehož přínos je dále ilustrován v Kapitole 4.1.

### 3.3.1 Interaktivní vizualizační nástroj

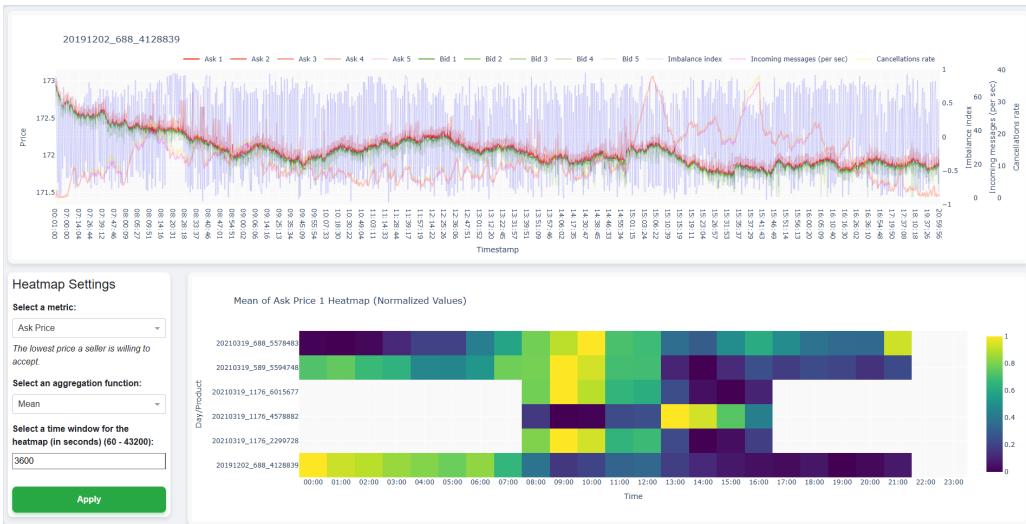
Vyvinutý vizualizační nástroj slouží k interaktivní analýze burzovních tržních dat v čase. Primární inspirací byla potřeba rychlé a přehledné analýzy velkého množství dat s možností detailního zanoření (*Details-on-Demand*). Jako vzorový případ pro testování a ladění nástroje byl zvolen náhodně vybraný den 2. 12. 2019, konkrétně trh s Market Segment ID = 688 a produktem se Security ID = 4128839.

Hlavními součástmi vizualizace jsou:

- **Cenový graf** zobrazující vývoj ceny v čase, doplněný o:
  - *Imbalance index* na druhé ose  $y$  – indikátor nerovnováhy mezi nabídkou a poptávkou.
  - *Frekvenci příchozích zpráv a frekvenci rušených objednávek* na třetí a čtvrté ose  $y$ , které mohou indikovat zvýšenou aktivitu a nestandardní tržní chování.
- **Teplotní mapa** umístěná ve spodní části obrazovky:
  - Na ose  $x$  je čas rozdelený podle zvolené granularity (výchozí hodnota: 1 hodina; nastavitelná v rozmezí od 1 minuty do 12 hodin).
  - Na ose  $y$  jsou jednotlivé kombinace dnů a produktů.
  - Hodnoty ( $z$ ) představují aggregaci vybrané dimenze pomocí zvolené aggregační funkce (*průměr, medián, minimum, maximum, standardní odchylka*).

Tyto části jsou znázorněny na Obrázku 3.1. V horní části obrazovky je zobrazen cenový graf s pěti cenovými úrovněmi, imbalance indexem a frekvencemi zpráv (všech a pouze těch o rušení objednávek). Ve spodní části je viditelné nastavení teplotní mapy (vlevo) a samotná teplotní mapa (vpravo) pro vybrané dny a produkty.

### 3.3.1 Interaktivní vizualizační nástroj



Obrázek 3.1: Základní snímek obrazovky z vizualizačního nástroje

Interaktivita nástroje je klíčová a zahrnuje následující funkce:

- Kliknutím na řádek v teplotní mapě dojde k načtení odpovídajícího cenového grafu pro daný den a produkt.
- Přejetím kurzoru nad teplotní mapou (*Hover*) se v cenovém grafu zobrazí tmavě modré zvýraznění příslušného časového okna.
- Přibližování a oddalování v cenovém grafu je synchronizováno s teplotní mapou – při přiblížení určité části cenového grafu se odpovídajícím způsobem aktualizuje i teplotní mapa.

Ukázka těchto interakcí je zachycena na Obrázku 3.2. Na snímku je vidět přefiltrování cenového grafu (pomocí kliknutí na legendu), který nyní zobrazuje pouze jednu (*nejlepší*) cenovou úroveň a frekvenci všech příchozích zpráv. Změnilo se také nastavení teplotní mapy – je zvolena jiná metrika, agregační funkce a časové okno, což vedlo k odlišnému vzhledu teplotní mapy oproti Obrázku 3.1. Dále je patrné přejetí kurzorem nad teplotní mapou (*Hover*) s interaktivní odezvou v podobě zvýraznění odpovídající části v cenovém grafu.

### 3 Implementovaná řešení



Obrázek 3.2: Snímek obrazovky z vizualizačního nástroje podchycující interaktivitu

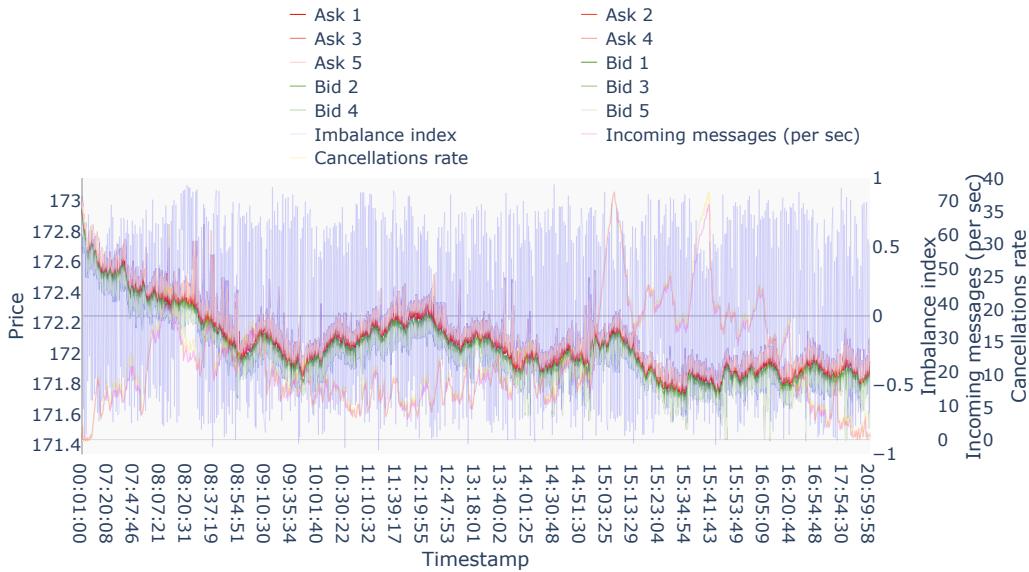
Kompatibilita s rozsáhlými datovými objemy a širokými časovými okny je zajištěna pomocí dynamického vzorkování. Vzhledem k velkému objemu dat je nutné optimalizovat výkon vykreslování – k tomu je využita knihovna **plotly-resampler**, která umožňuje interaktivní zobrazení rozsáhlých časových řad pomocí automatické redukce počtu datových bodů v závislosti na úrovni přiblížení.

Implementace vizualizačního nástroje je dostupná ve složce `src/visualization`, konkrétně v souboru `visuals.py`. Pro samotné vykreslování grafů je využita knihovna **Plotly** a webový framework **Dash**, který umožňuje tvorbu plně interaktivních webových aplikací.

Vyvinutý nástroj slouží jako podpůrný prostředek pro detailní pochopení chování tržních dat a výrazně přispívá k návrhu a validaci automatických metod detekce anomalií založených na strojovém učení.

Protože Obrázky 3.1 a 3.2 zobrazují pouze celé prostředí vizualizačního nástroje (jedná se o snímky obrazovky), byly pro lepší čitelnost exportovány také samotné cenové grafy. Tyto detailní pohledy jsou uvedeny na následujících Obrázcích 3.3 a 3.4.

### 3.3.1 Interaktivní vizualizační nástroj



Obrázek 3.3: Cenový graf s pěti cenovými úrovněmi, imbalance indexem a frekvencemi zpráv (všech a pouze těch o rušení objednávek) (Security ID **4128839**)



Obrázek 3.4: Přefiltrovaný cenový graf s jednou (nejlepší) cenovou úrovní (Top of Book) a frekvencemi příchozích zpráv (Security ID **4128839**)



# Výsledky a diskuze

4

If you think it's simple, then you have misunderstood the problem.

---

*Bjarne Stroustrup (1950-), Creator of C++ and Professor of Computer Science at Texas A&M University and Columbia University*

V této kapitole jsou prezentovány klíčové výsledky získané během testování jednotlivých metod pro detekci anomalií. Výsledky jsou diskutovány především z pohledu praktické interpretace.

**Výběr analyzovaných dat.** Jak již bylo zmíněno v Kapitole 3.1, pro účely experimentální analýzy byly zvoleny především tzv. *Triple Witching Days*, tedy dny s očekávaně zvýšenou *volatilitou* a tržní aktivitou. Doplňkově byl zařazen jeden náhodně vybraný den a produkt sloužící jako referenční případ běžného obchodního dne – 2. 12. 2019 (Market Segment ID 688 a Security ID 4128839). Výběr z *Triple Witching Days* pokrývá dva nejaktivnější tržní instrumenty a tři menší či středně aktivní produkty. Z Tabulky 4.1 je vidět, že v rámci výběru dat v práci jsou Security ID unikátní, není proto do budoucna nutné uvádět celou trojici den, Market Segment ID (v Tabulce 4.1 pouze **MS ID**) a Security ID.

Tabulka 4.1: Konkrétní výběr dnů a produktů

Datum	MS ID	Security ID	Produkt (vysvětlení na další stránce)
2019-12-02	688	4128839	FGBL SI 20200306 PS
2021-03-19	589	5594748	FDAX SI 20210618 CS
2021-03-19	688	5578483	FGBL SI 20210608 PS
2021-03-19	1176	6015677	ODAX SI 20210416 CS EU C 14750 0
2021-03-19	1176	2299728	ODAX SI 20211217 CS EU C 14200 0
2021-03-19	1176	4578882	ODAX SI 20210618 CS EU P 14800 0

Tabulky 4.1 a 4.2 jsou spojitelné přes společný sloupec Security ID. V Tabulce 4.2 je vidět jednoduchý přehled vybraných datových souborů – počet zpráv burzovního serveru, velikost surového JSON souboru a přibližná likvidita.

Tabulka 4.2: Přehled analyzovaných datových souborů

Security ID	Počet zpráv	Velikost [MB]	Likvidita
4128839	747 092	554	střední
5594748	2 966 893	2 280	vysoká
5578483	2 505 886	1 780	vysoká
6015677	262 974	291	střední
2299728	186 075	194	nízká
4578882	180 546	200	nízká

**Vysvětlení názvů produktů.** Názvy produktů na derivátové burze EUREX mají strukturovanou podobu a jednotlivé části zkratek nesou specifické informace o daném kontraktu. První část označuje konkrétní produkt:

- FGBL (*Futures on Euro-Bund*) – *futures* kontrakt na německé státní dluhopisy.
- FDAX (*Futures on DAX*) – *futures* kontrakt na německý akciový index DAX.
- ODAX (*Options on DAX*) – *opční* kontrakt (*call* nebo *put*) na index DAX.

Další část u všech produktů označená jako SI značí tzv. *Standard Instrument*, tedy standardizovaný (běžný) kontrakt (opakem by mohl být tzv. *Flexible Instrument* – FI).

Následuje datum expirace (splatnosti) kontraktu ve formátu RRRRMMDD (YYYYMMDD).

Poslední společná část mezi *futures* a *opcemi* označuje typ vypořádání:

- CS (*Cash Settlement*) – kontrakt je vypořádáván finančně.
- PS (*Physical Settlement*) – při uplatnění kontraktu (nákupu, resp. prodeji) dojde k fyzickému dodání podkladového aktiva.

U *opčních* kontraktů následuje specifikace stylu uplatnění – všude je společné EU, tedy *evropský styl* (opce je možné uplatnit pouze k datu expirace). Druhou možností by bylo AM, tedy *americký styl* (opce je možné uplatnit kdykoliv do expirace).

Dále se uvádí typ opce – C (*Call*) a P (*Put*).

Předposlední částí je **realizační cena** (*Strike Price*) – jedná se o předem stanovenou cenu, za kterou může držitel opce buď koupit (u *Call opce*), nebo prodat (u *Put opce*). Poslední část označuje verzi kontraktu (u všech uvedených produktů je stejná – 0) – slouží k odlišení kontraktů, které mají jinak stejné parametry.

**Výpočetní prostředí.** Experimenty byly prováděny částečně lokálně, hlavně však na distribuovaném výpočetním gridu **MetaCentrum**. Pro transparentnost je uvedena technická specifikace lokálního stroje, na kterém probíhaly všechny experimenty, které nebyly spuštěny na **MetaCentru**:

- **CPU:** AMD Ryzen 5 4600H
- **GPU:** NVIDIA GeForce GTX 1660 Ti
- **RAM:** 16 GB DDR4

V případě gridových výpočtů v rámci **MetaCentrum** nebyl použit konkrétní fixní uzel – výpočetní prostředí bylo závislé na momentální alokaci prostředků plánovačem. Specifikace konkrétního uzlu jsou však pro každou úlohu zaznamenány v hlavičce příslušného skriptu v repozitáři.

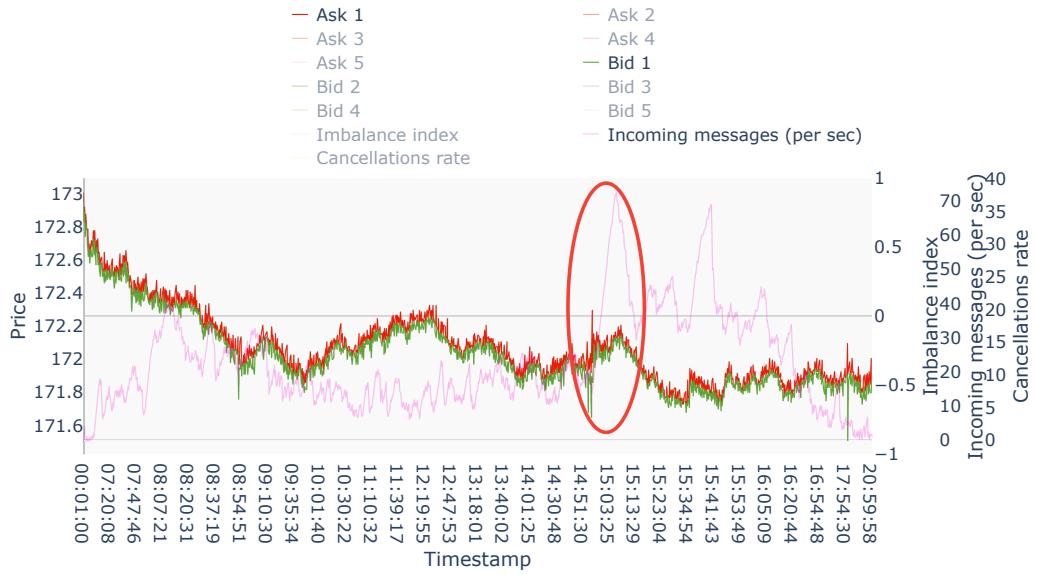
## 4.1 Vizuální analýza

Jako demonstrace schopností vyvinutého vizualizačního nástroje i důvod pro hlubší výzkum anomálních vzorů byla provedena detailní manuální analýza konkrétního produktu ze dne 2. 12. 2019 – Security ID = 4128839. Jednalo se o náhodně vybraný den a produkt, sloužící pro první přibližné pozorování a testování nástroje.

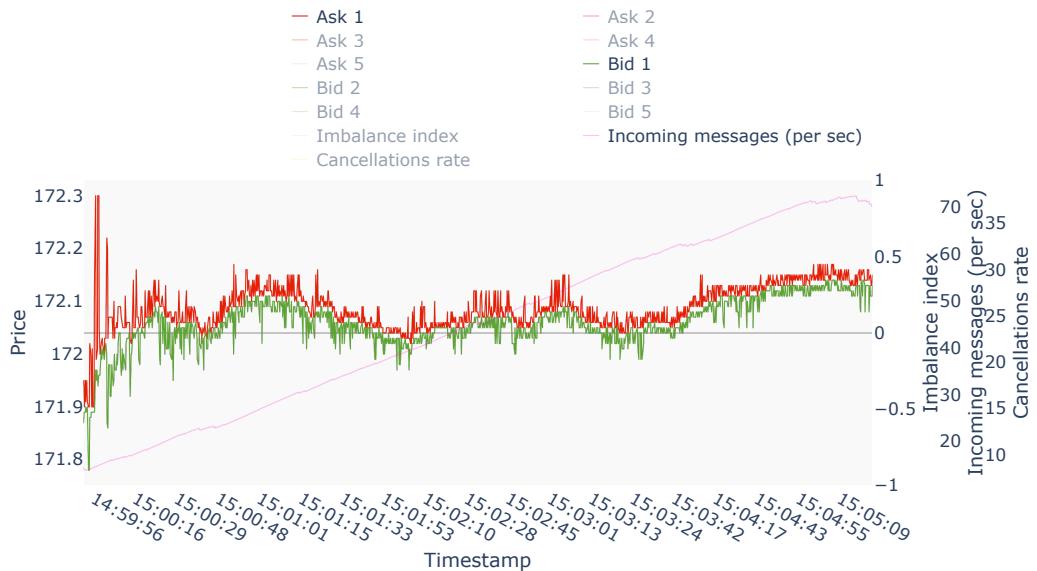
Na Obrázcích 4.1 a 4.2 jsou vidět výsledky zkoumání, tj:

- Mezi **14:59–15:00** došlo k významnému rozšíření *bid-ask spreadu*.
- Rovněž došlo k výraznému nárůstu *frekvence příchozích zpráv*, včetně zpráv o rušení objednávek.
- Zároveň o pár minut později dochází k prudkému nárůstu ceny aktiva.

#### 4 Výsledky a diskuze



Obrázek 4.1: Přefiltrovaný cenový graf s jednou (nejlepší) cenovou úrovni (*Top of Book*) a frekvencemi příchozích zpráv – červenou elipsou je označena *podezřelá oblast (Security ID 4128839)*



Obrázek 4.2: Přiblížení na *podezřelou oblast* z Obrázku 4.1

Tato kombinace znaků byla vyhodnocena jako podezřelá. Následně byla komunikována s experty z **Deutsche Börse AG**, kteří potvrdili, že se sice jedná o „nestandardní chování“, nicméně dle jejich vyjádření se „neprokázala manipulativní aktivita“. Zvýšenou obchodní aktivitu vysvětlují jako „obvyklou“ reakci na předchozí ekonomické oznámení centrální banky.

Tento experiment ilustruje význam vizuální analýzy při exploraci dat a motivoval vývoj automatizovaných metod detekce anomálií. Výsledky modelů v tomto časovém úseku potvrzují zvýšené skóre *anomálnosti* v uvedeném intervalu, což podporuje jejich validitu.

## 4.2 Redukce dimenze dat

Principy použité pro analýzu redukce dimenze dat byly podrobně popsány v Kapitole 3.1.3. Počáteční dataset obsahuje 13 atributů, včetně času, tedy 12 závislých proměnných na časové ose. Pro lepší pochopení struktury dat jsou v Příloze B uvedeny obrázky přes všechny dimenze v závislosti na čase.

Všechny obrázky v této sekci budou demonstrovány na Security ID **4128839**.

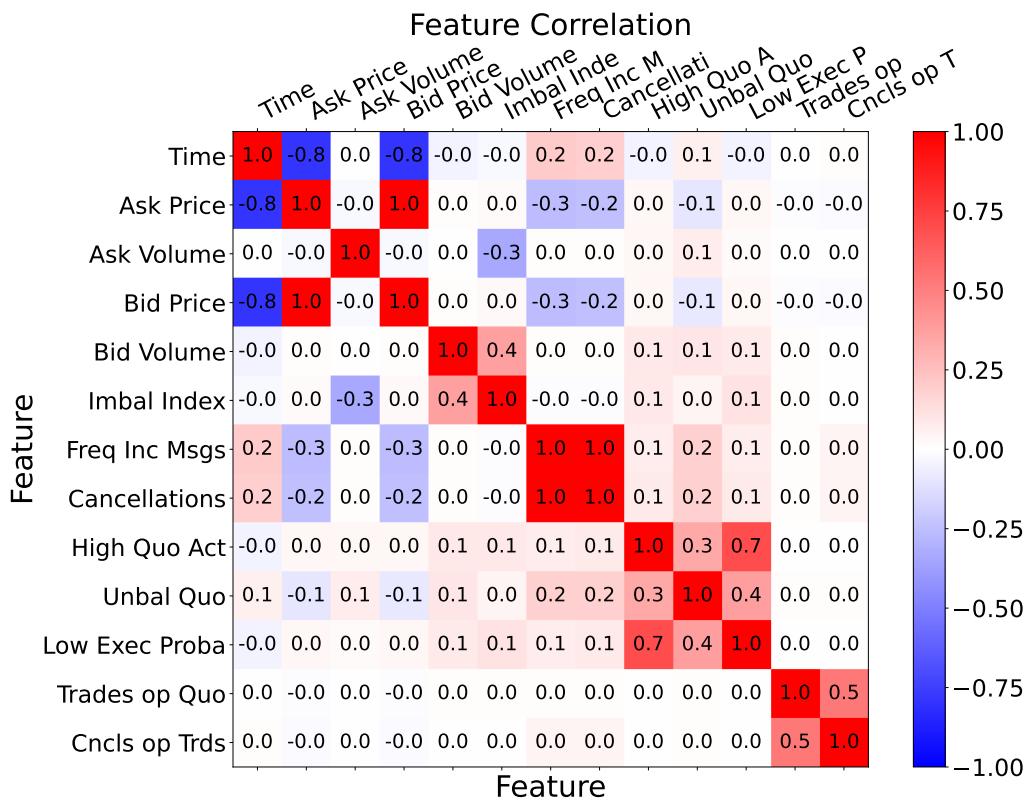
### 4.2.1 Korelační analýza

Korelační analýza odhalila, že mezi některými příznaky existují silné korelace. Korelační matice ukazuje na několik významných vztahů, které lze očekávat vzhledem k povaze dat. Tato matice je vyobrazena na Obrázku 4.3, z které vycházejí následující závěry. Například ceny (Ask Price 1 a Bid Price 1) jsou téměř perfektně korelovány, což naznačuje, že je možné použít pro analýzu jednu z těchto cen, nebo ideálně *Mid Price* (průměr z nabídkové a poptávkové ceny).

Dále byly identifikovány korelace mezi *Imbalance Index* a objemy objednávek (Ask Volume 1 a Bid Volume 1), což je přirozené, neboť *imbalance index* je odvozený z těchto objemů. Frekvence příchozích zpráv a frekvence zpráv o rušení objednávek jsou rovněž silně korelovány, což je očekávané, jelikož rušení objednávek je podmnožinou všech příchozích zpráv.

Další korelace se objevují mezi proměnnými jako *High Quoting Activity*, *Unbalanced Quoting*, a *Low Execution Probability*, které mají podobné výpočtové základy. Pro případy *spoofingu* (manipulace trhem) jsou výborným indikátorem korelace mezi *Trades Oppose Quotes* a *Cancels Oppose Trades*, což signalizuje nepřirozené chování mezi zveřejněnými objednávkami a skutečnými obchody.

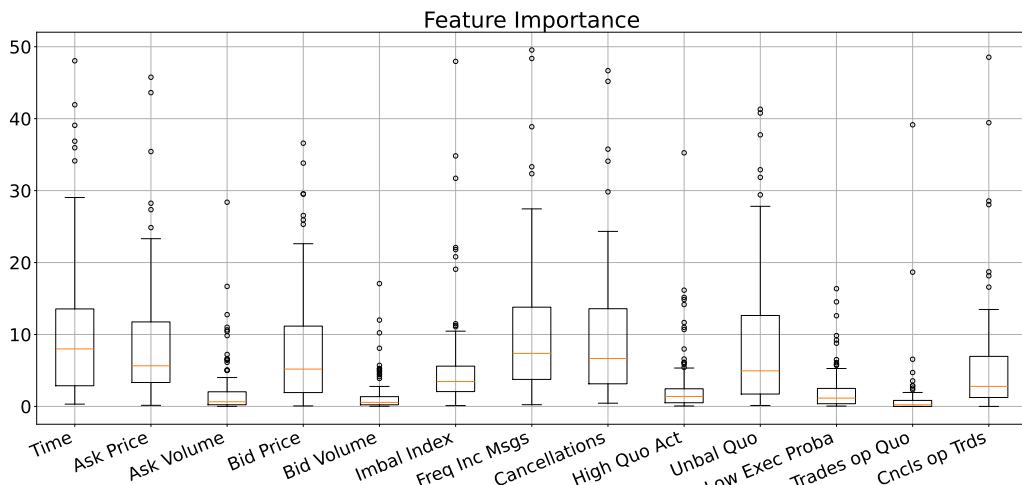
Tato analýza byla provedena na všech datových souborech, přičemž výsledky korelací byly konzistentní napříč jednotlivými datovými sadami.



Obrázek 4.3: Korelační matice (Security ID 4128839)

## 4.2.2 Důležitost příznaků

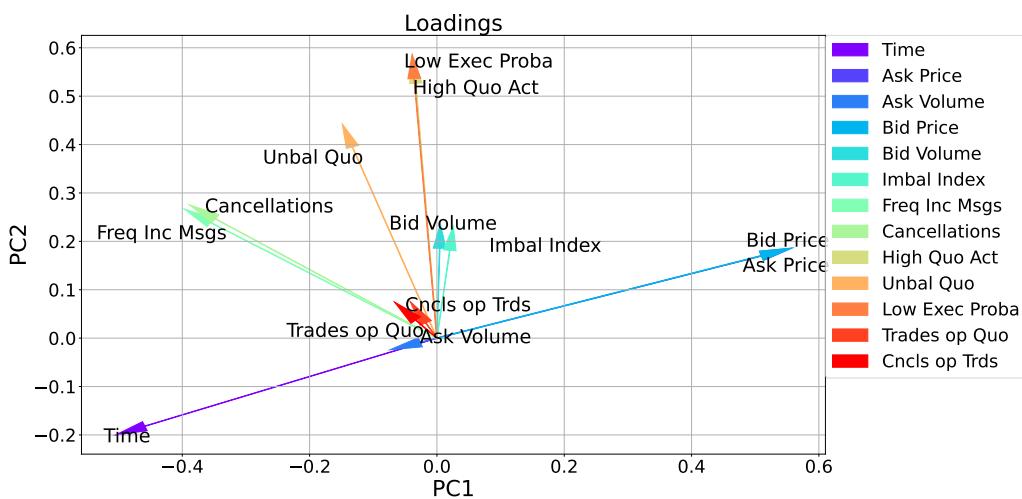
Pro hodnocení důležitosti příznaků byl využit algoritmus *DIFFI*, který odhalil klíčové proměnné pro model *izolačního lesa* (*Isolation Forest*). Výsledky ukázaly, že mezi nejméně důležité příznaky patří Ask Volume 1, Bid Volume 1, High Quoting Activity, Low Execution Probability a Trades Oppose Quotes. Tento závěr byl proveden na základě Obrázku 4.4, přičemž analýza byla opět provedena na všech datových souborech a výsledky byly vždy podobné napříč jednotlivými datovými soubory.



Obrázek 4.4: Boxplot důležitosti příznaků podle algoritmu *DIFFI* – spuštěno stokrát pro zajištění stabilních výsledků (Security ID 4128839)

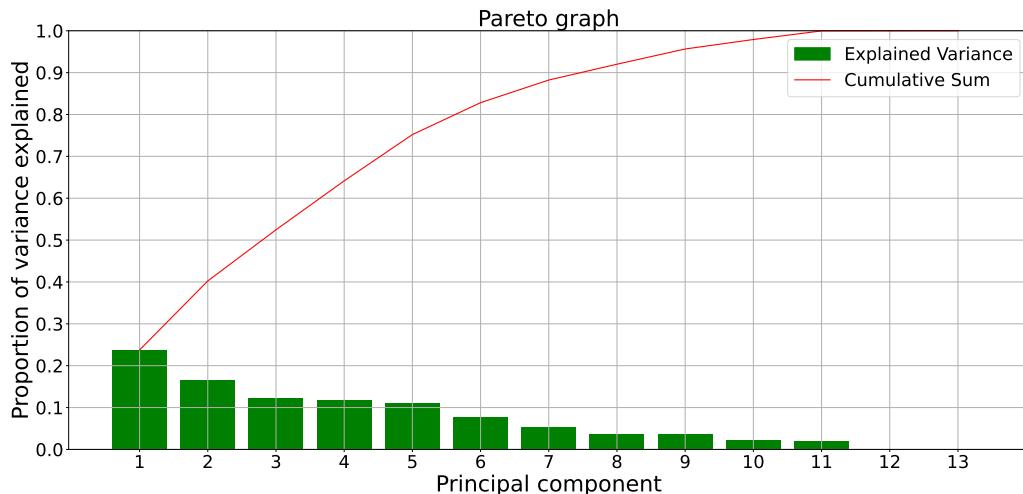
### 4.2.3 Analýza hlavních komponent

V rámci *analýzy hlavních komponent* (PCA) byly vizualizovány **koeficienty hlavních komponent** (*Loadings*) v prostoru první a druhé komponenty. V tomto grafu jsou jasné viditelné shluky, které odpovídají těm, jež byly popsány v předchozí korelační analýze – viz Obrázek 4.5. PCA poskytlo užitečný pohled na to, jak jsou data rozložena v nižší dimenzi, a jak jsou různé atributy seskupeny podle podobnosti.



Obrázek 4.5: Koeficienty hlavních komponent jako vektory v prostoru první a druhé komponenty po aplikaci PCA na celá data (Security ID 4128839)

Z **Paretova diagramu** (Obrázek 4.6) bylo zjištěno, že pro zachování alespoň 90 % informací je nutné zachovat minimálně 7 komponent. Nicméně, jak bylo uvedeno v Kapitole 3.1.3, cílem není použít PCA na celý dataset, ale spíše aplikovat PCA na specifické shluky příznaků identifikované v korelační analýze, což je také podpořeno výsledky z předchozí a této kapitoly.



Obrázek 4.6: Paretův diagram zobrazující procento zachované informace (vysvětleného rozptylu) při aplikaci PCA na celá data (Security ID **4128839**)

#### 4.2.4 Výsledná redukce dimenze

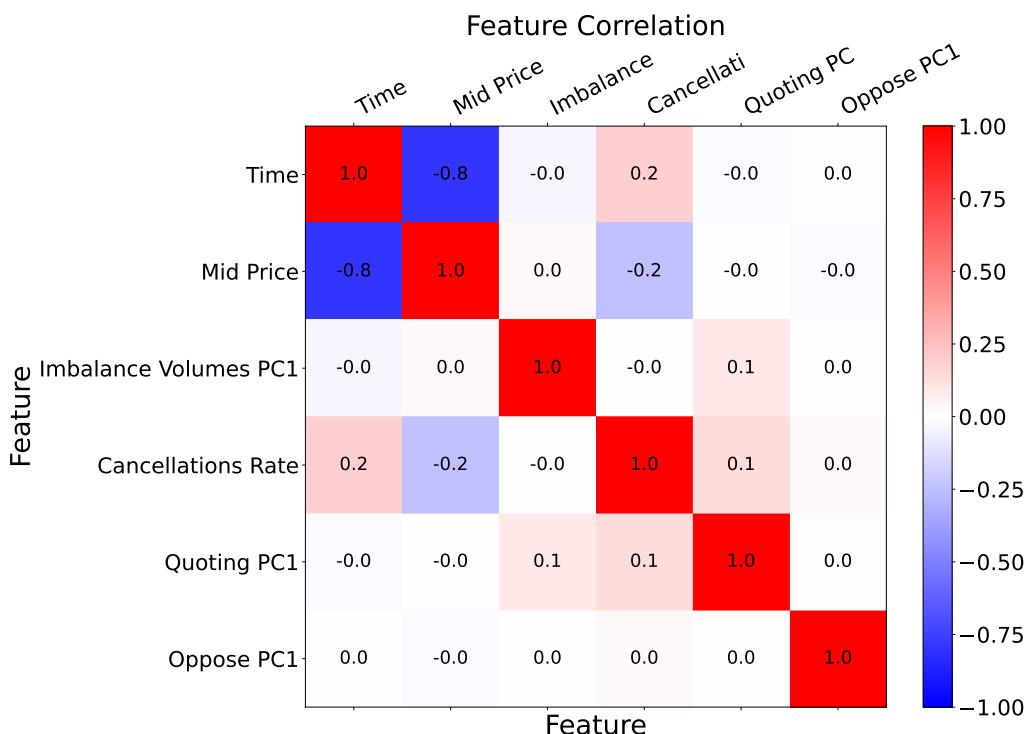
Na základě výsledků výše zmíněných analýz došlo k aplikaci redukce dimenze, která vedla k výběru klíčových komponent a transformacím. Zde je přehled všech změn:

- **Ask Price 1** a **Bid Price 1** byly nahrazeny **Mid Price** (průměr obou cen), čímž došlo k výraznému zjednodušení bez ztráty relevantních informací o cenové hladině.
- **Imbalance Index**, **Bid Volume 1** a **Ask Volume 1** byly zredukovány do první *hlavní komponenty* užitím PCA, která díky tomu, že *imbalance index* je vypočítán z objemů zachovává většinu své interpretability. Tato komponenta vyjadřuje objem objednávek a jejich nerovnováhu na trhu.
- **Cancellation Rate** byl zachován, jelikož tato proměnná je klíčová pro detekci *spoofingu* – manipulace s objednávkami na trhu.
- **Frequency of Incoming Messages** byla vyhozena, neboť byla silně korelována s frekvencí zpráv o rušení objednávek. Zachována byla pouze ta, která se ukázala jako kontextově významnější – *Cancellation Rate*.

- **High Quoting Activity, Unbalanced Quoting a Low Execution Probability** byly sloučeny do první *hlavní komponenty* užitím PCA. Díky podobnosti výpočtů těchto příznaků opět došlo k částečnému zachování interpretability – jedná se o komponentu, která souvisí s objemy objednávek.
- **Trades Oppose Quotes a Cancels Oppose Trades** byly také sjednoceny do první *hlavní komponenty* užitím PCA, tato komponenta odráží specifické tržní chování, kde na jedné straně trhu se objednávky rapidně přidávají a mazají a na druhé straně dochází k uzavíraní obchodů – typické chování manipulátorů užívajících strategii *spoofingu*.

Po aplikaci této redukce dimenze se prostor dat zúžil na 6 dimenzí, z nichž jedna je časová a dalších 5 jsou časově závislé proměnné. Korelační matice po redukci na Obrázku 4.7 ukazuje, že i po snížení dimenze zůstávají některé komponenty slabě korelovány, ale bylo dosaženo kvalitní redukce s maximálním zachováním interpretability a klíčových informací pro detekci *spoofingu*.

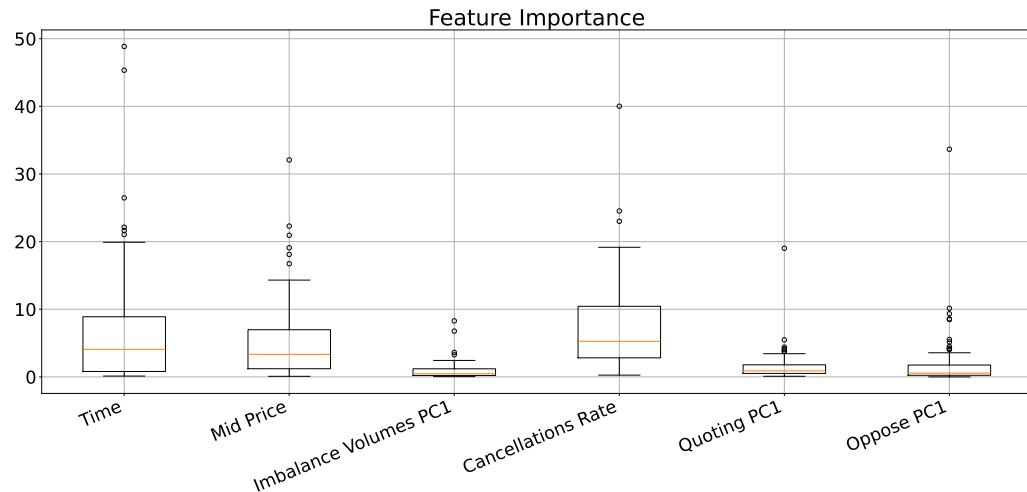
Z důvodu lepšího pochopení struktur v datech jsou v Příloze B uvedeny také obrázky přes všechny dimenze v závislosti na čase po zredukovaní dimenzí.



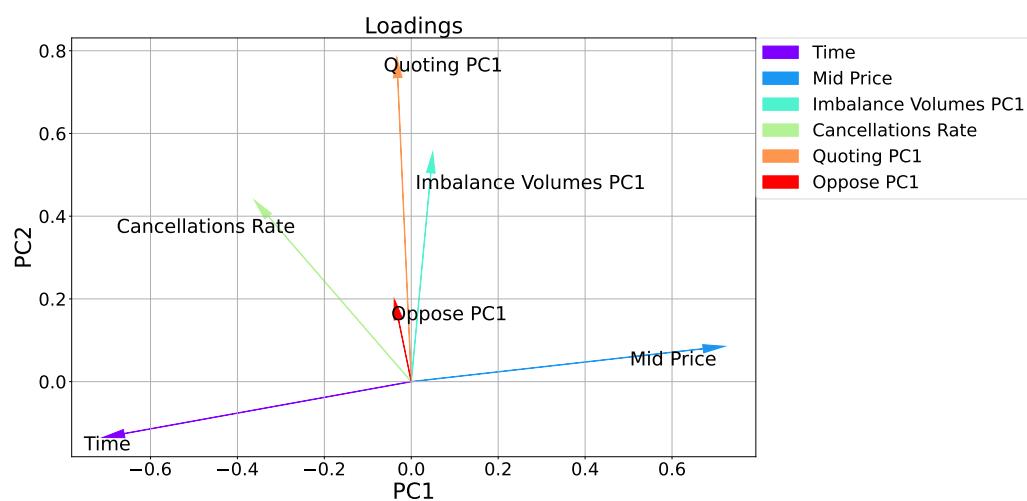
Obrázek 4.7: Korelační matice po redukci dimenzí (Security ID 4128839)

#### 4 Výsledky a diskuze

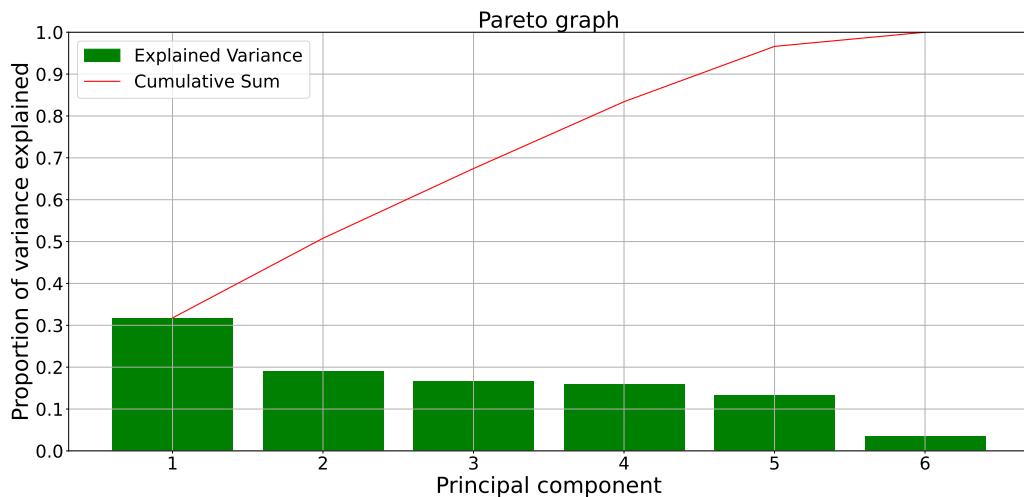
Dále jsou k dispozici Obrázky 4.8, 4.9 a 4.10, které dále podporují smysl a úspěšnost redukce dimenzií.



Obrázek 4.8: Boxplot důležitosti příznaků podle algoritmu *DIFFI* po redukci dimenzií – spuštěno stokrát pro zajištění stabilních výsledků (Security ID **4128839**)



Obrázek 4.9: Koeficienty hlavních komponent jako vektory v prostoru první a druhé komponenty po aplikaci PCA na celá data po redukci dimenzií (Security ID **4128839**)



Obrázek 4.10: Paretov diagram zobrazující procento zachované informace (vysvětleného rozptylu) při aplikaci PCA na celá data po redukci dimenzí (Security ID **4128839**)

## 4.3 Prohledávání prostoru hyperparametrů

Jak již bylo zmíněno v Kapitole 3.2.2, proces **prohledávání prostoru hyperparametrů** (*Grid Search*) probíhal na infrastruktuře **MetaCentrum**. V rámci tohoto procesu bylo otestováno celkem **31** různých kombinací hyperparametrů pro referenční modely a **135** kombinací pro modely založené na neuronových sítích. Každá kombinace byla vyhodnocena pomocí *5-fold křížové validace* na šesti různých datových sadách, což vedlo k velmi rozsáhlému experimentálnímu prostoru.

Celkem bylo na **MetaCentru** vygenerováno a zpracováno více než **2 800 unikátních úloh** (včetně testovacích a přípravných běhů). Odhadovaná výpočetní náročnost těchto experimentů odpovídá přibližně **250 CPU dní**. Výsledky byly průběžně zaznamenávány a vizualizovány pomocí platformy **Weights & Biases**, a to zejména pomocí metrik *EM* a *MV*, které jsou detailně popsány v Kapitole 3.2.3. U neuronových modelů bylo hlavně nahlízeno na hodnoty ztrátových funkcí na validačních datech.

Vzhledem k velkému množství dostupných vizualizací z experimentů musela být do celé této kapitoly zařazena pouze reprezentativní podmnožina obrázků. Kompletní sada výstupů z procesu *prohledávání prostoru hyperparametrů* je dostupná v repozitáři ve složce `/img/grid_search`.

## Referenční modely

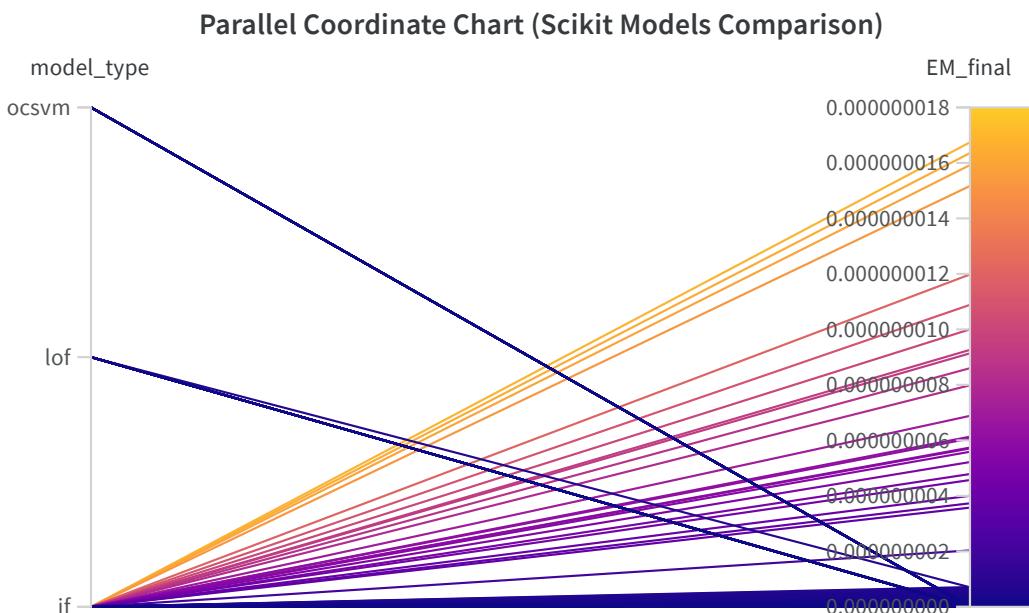
Všechny laděné hyperparametry s jejich konkrétními volenými hodnotami jsou uvedeny v následující Tabulce 4.3

Tabulka 4.3: Výčet laděných hyperparametrů a jejich testovaných hodnot (referenční modely)

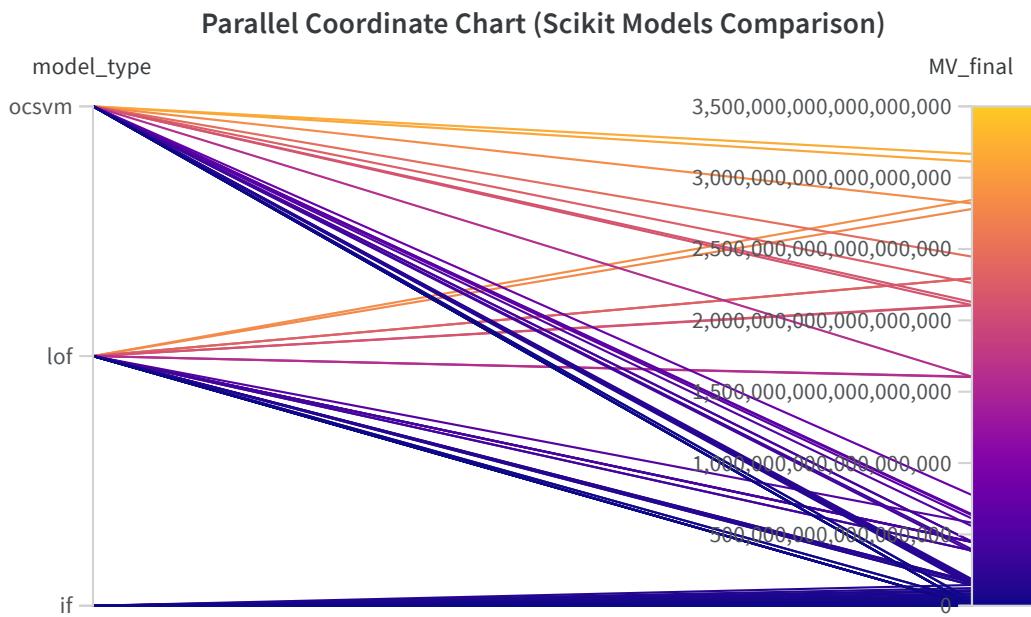
Hyperparametr	Testované hodnoty (odděleny čárkou)
n_estimators	50, 100, 200
max_samples	auto, 1024, 0.1
max_features	0.5, 0.75, 1.0
gamma	scale, auto
n_neighbors	16, 32

Z referenčních modelů dosahoval konzistentně nejlepších výsledků algoritmus *izolačního lesu* (*Isolation Forest, IF*). Naproti tomu *lokální faktor odlehlosti* (*Local Outlier Factor, LOF*) a *jednotřídní SVM* (*One-Class SVM, OCSVM*) vykazovaly výrazně horší výkonnost. Na Obrázcích 4.11 a 4.12 je vidět srovnání všech běhů v rámci experimentu – jednotlivé křivky představují jednotlivé běhy trénování.

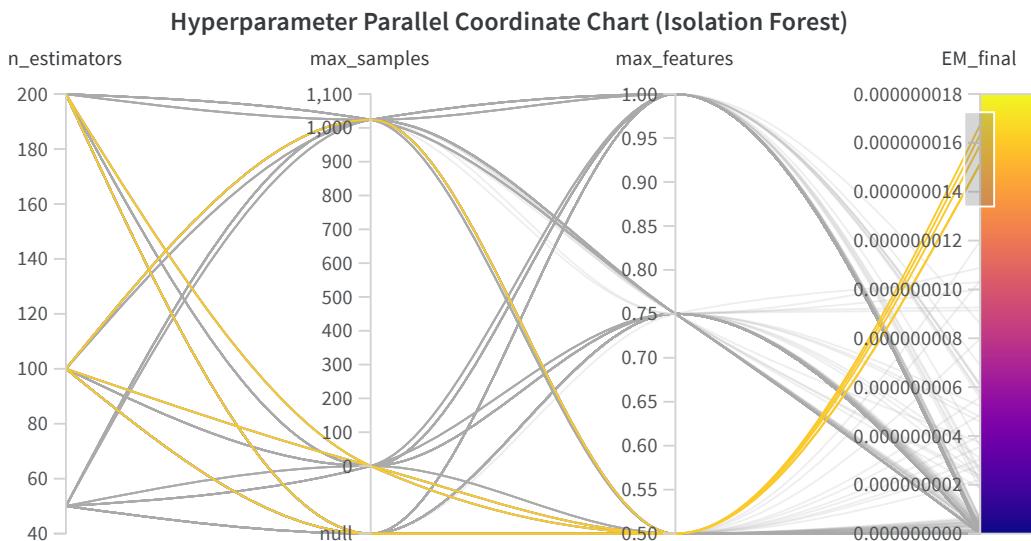
■ Připomenutí: žádoucí jsou vysoké hodnoty EM a nízké hodnoty MV!

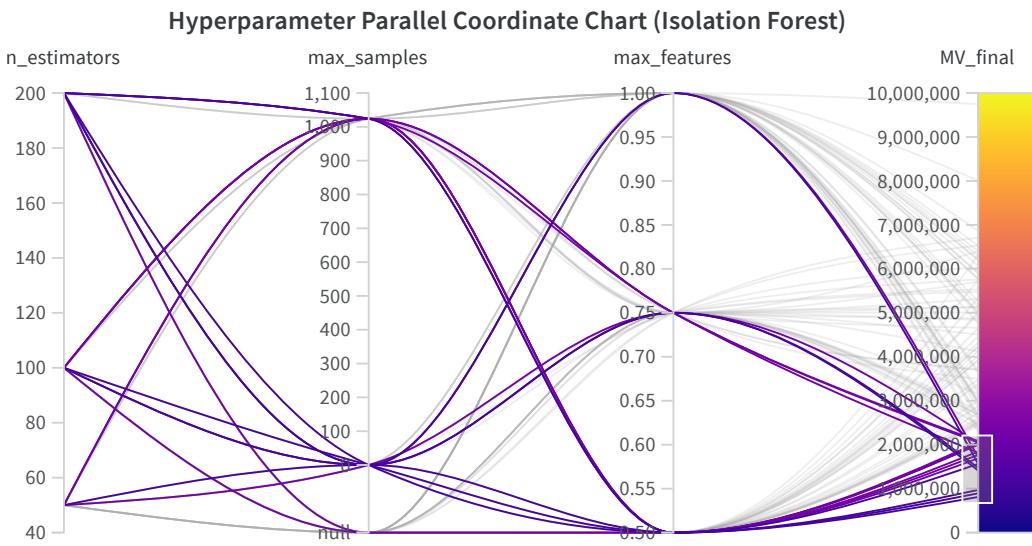


Obrázek 4.11: Srovnání všech referenčních modelů na metrice *EM*


 Obrázek 4.12: Srovnání všech referenčních modelů na metrice  $MV$ 

Výběr nejlepšího nastavení hyperparametrů probíhal vizuální analýzou – příklad na Obrázcích 4.13 a 4.14 demonstruje výběr nejlepších nastavení pro model *IF*. Na obrázcích je vidět interaktivní filtrační okno, které výrazně zlehčovalo výběr nejlepších modelů.


 Obrázek 4.13: Filtrované zobrazení všech běhů modelu *IF* na metrice  $EM$



Obrázek 4.14: Filtrované zobrazení všech běhů modelu *IF* na metrice  $MV$

Průměrně nejlépe hodnocené nastavení hyperparametrů pro algoritmus *IF* bylo následující:

- `n_estimators` = 100
- `max_samples` = 0.1
- `max_features` = 0.5

Pro OCSVM a *LOF* byly průměrně nejlepší nalezené hodnoty následující:

- OCSVM: `gamma` = `scale`
- LOF: `n_neighbors` = 32

Vzhledem k očekávané nižší výkonnosti modelů OCSVM a *LOF* nebylo prohledávání realizováno tak podrobně jako u modelu *IF*. Dále je třeba zmínit, že implementace *LOF* a OCSVM v rámci knihovny **Scikit-learn** neposkytuje tak široké možnosti nastavení hyperparametrů jako v případě *IF*.

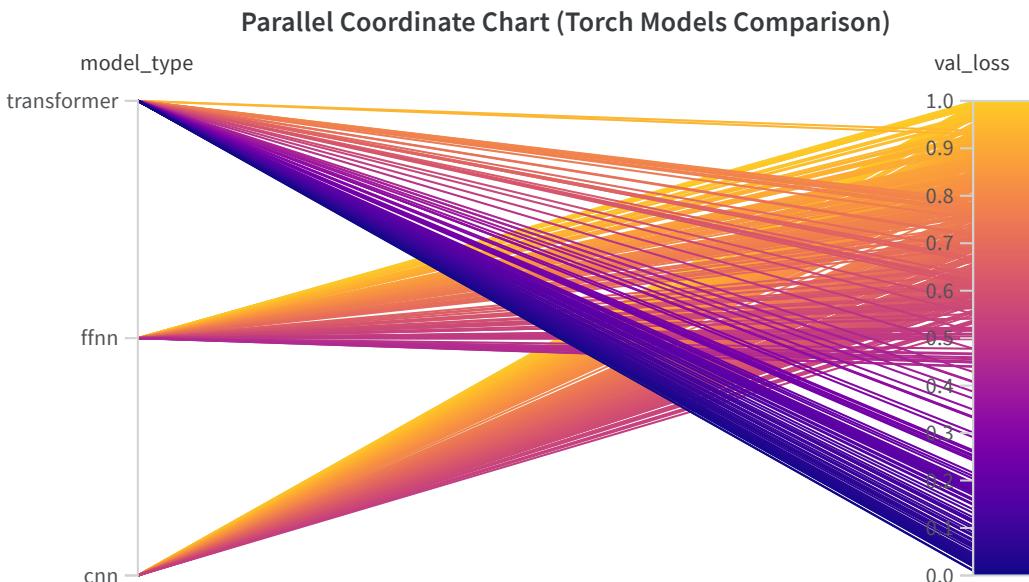
## Neuronové modely

Všechny laděné hyperparametry s jejich konkrétními volenými hodnotami jsou uvedeny v následující Tabulce 4.4

Tabulka 4.4: Výčet laděných hyperparametrů a jejich testovaných hodnot (neuronové modely)

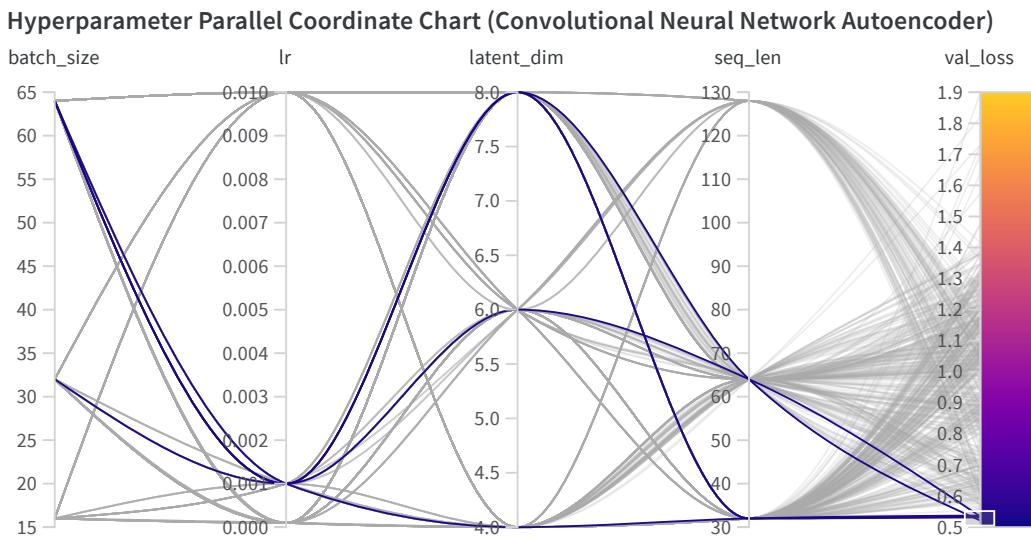
Hyperparametr	Testované hodnoty (odděleny čárkou)
batch_size	16, 32, 64
learning_rate	1e-2, 1e-3, 1e-4
latent_dim	4, 6, 8
seq_len	32, 64, 128

Modely *autoenkodérů* výrazně překonaly referenční algoritmy na základě *EM* a *MV* metrik. Z tohoto důvodu byla pro budoucí porovnávání napříč modely využita logaritmická škála – v textu na tuto skutečnost bude upozorněno, až to bude relevantní. Nejlepších výsledků dosahovala architektura *transformer*, která vykazovala nejnižší hodnoty ztrátové funkce na validačních datech – viz Obrázek 4.15. Dopředné neuronové sítě (*FFNN*) a konvoluční neuronové sítě (*CNN*) dosahovaly navzájem srovnatelných výsledků.



Obrázek 4.15: Srovnání všech neuronových modelů na hodnotě ztrátové funkce na validačních datech (filtrování hodnot – maximálně 1.0)

Výběr optimálního nastavení pro každý neuronový model probíhal obdobně jako u referenčních modelů – příklad na Obrázku 4.16 demonstruje výběr nejlepšího nastavení pro model *CNN* z úhlu pohledu ztrátové funkce.



Obrázek 4.16: Filtrované mapování nastavení hyperparametrů všech běhů modelu CNN na ztrátovou funkci

Optimální nastavení pro jednotlivé architektury bylo následující:

- **FFNN** (461 trénovatelných parametrů):

`batch_size = 32`

`learning_rate = 1e-3`

`latent_dim = 8`

- **CNN** (785 trénovatelných parametrů):

`batch_size = 32`

`learning_rate = 1e-3`

`latent_dim = 8`

`seq_len = 64`

- **Transformer** (838 765 trénovatelných parametrů):

`batch_size = 64 nebo 16`

`learning_rate = 1e-4`

`seq_len = 32 nebo 128`

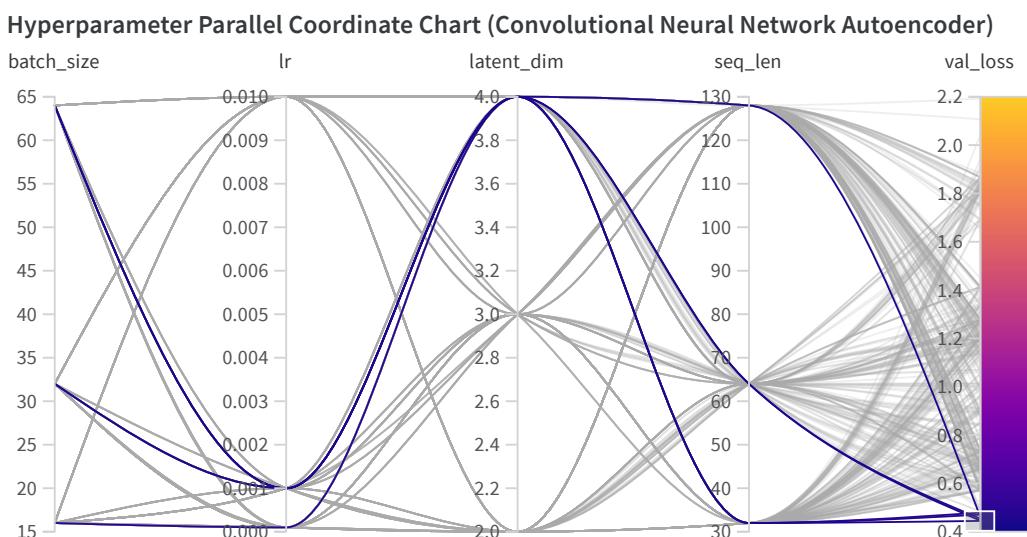
Je důležité upozornit, že modely *transformer* mají několikanásobně více parametrů než *FFNN* a *CNN*. Přímé porovnávání jejich výkonnosti je proto nutné interpretovat s jistou mírou opatrnosti.

#### 4.3.1 Prohledávání prostoru hyperparametrů po redukci dimenze vstupních dat

U *transformeru* byly nalezeny původně dva nejlepší modely – buď méně dlouhých sekvencí, nebo více kratších sekvencí. Dalšími experimenty nakonec vyšlo najev, že více kratších sekvencí pro tuto konkrétní úlohu funguje obecně lépe – ve výčtu výše je tato konfigurace označena tučně.

### 4.3.1 Prohledávání prostoru hyperparametrů po redukci dimenze vstupních dat

Po aplikaci metody *redukce dimenze vstupních dat* bylo *prohledávání prostoru hyperparametrů* provedeno znovu. Na základě vizuální analýzy výsledků ve **WandB** lze konstatovat, že u neuronových modelů došlo ke znatelnému **zlepšení** výkonu, především v oblasti hodnot validační ztráty – pro vizuální srovnání slouží Obrázky 4.16 a 4.17. Zatímco *transformer* již předtím vykazoval nízké hodnoty, po redukci dimenze se mu přiblížily i ostatní architektury. Navíc díky snížení rozměru vstupních dat byl celý proces trénování výrazně **rychlejší**.



Obrázek 4.17: Filtrované mapování nastavení hyperparametrů všech běhů modelu CNN na ztrátovou funkci po redukci dimenzionality vstupních dat

### Referenční modely

Všechny laděné hyperparametry s jejich konkrétními volenými hodnotami jsou stejně jako před redukcí dimenze vstupních dat, viz Tabulka 4.3.

Obecně lze tvrdit, že redukce dimenze měla pozitivní vliv na časovou výkonnost *referenčních modelů*. Na základě metrik *EM* a *MV* lze však konstatovat, že redukce dimenze měla spíše negativní vliv na shlukovací schopnosti těchto modelů. V některých případech byly výsledky evaluací horší než na původních datech.

Průběh vizuální analýzy je ekvivalentní s popisem z kapitoly výše o referenčních modelech bez redukce dimenze dat. Nejlepší konfigurace pro jednotlivé modely na datech s redukovanou dimenzionalitou byly následující:

- **IF:**

```
n_estimators = 100  
max_samples = 0.1  
max_features = 0.5
```

- **OCSVM:**

```
gamma = scale
```

- **LOF:**

```
n_neighbors = 32
```

## Neuronové modely

Laděné hyperparametry s jejich konkrétními volenými hodnotami jsou stejné jako před redukcí dimenze vstupních dat s výjimkou `latent_dim` – po redukci bylo voleno z hodnot 2, 3 a 4, viz Tabulka 4.5.

Tabulka 4.5: Výčet laděných hyperparametrů a jejich testovaných hodnot po redukci dimenze vstupních dat (neuronové modely)

Hyperparametr	Testované hodnoty (odděleny čárkou)
<code>batch_size</code>	16, 32, 64
<code>learning_rate</code>	1e-2, 1e-3, 1e-4
<code>latent_dim</code>	2, 3, 4
<code>seq_len</code>	32, 64, 128

Redukce dimenze se pozitivně promítla do výkonnosti všech neuronových modelů, přičemž došlo ke znatelnému snížení hodnot ztrátové funkce. Navíc byl celý proces trénování rychlejší díky menšímu objemu dat.

Výběr optimálních nastavení hyperparametrů opět probíhal dle výše popsaného postupu o neuronových modelech bez redukce dimenze dat. Nejlepší konfigurace hyperparametrů pro jednotlivé modely na redukovaných datech byly:

- **FFNN** (120 trénovatelných parametrů):

```
batch_size = 16
learning_rate = 1e-3
latent_dim = 4
```

- **CNN** (194 trénovatelných parametrů):

```
batch_size = 32
learning_rate = 1e-3
latent_dim = 4
seq_len = 64
```

- **Transformer** (838 310 trénovatelných parametrů):

```
batch_size = 32
learning_rate = 1e-4
seq_len = 64
```

## 4.4 Ověření nejlepších hyperparametrů

Předchozí kapitola se zabývala výběrem optimálních hodnot hyperparametrů jednotlivých modelů. Tento výběr byl proveden analýzou přibližně 10 až 20 nejúspěšnějších běhů každého modelu, z nichž bylo určeno dominantní nastavení. V některých případech však výběr nebyl jednoznačný, jak ukazuje například model *transformer* na neredukovaných datech.

Cílem následujících experimentů je ověřit, že zvolené konfigurace skutečně představují obecně robustní nastavení modelů. Pro každý z nejlepších modelů byla provedena opakovaná evaluace s deseti různými *počátečními hodnotami* (*Seed*) generátoru pseudonáhodných čísel. Tímto způsobem byla minimalizována náhodná variabilita a ověřena stabilita dosažených výsledků napříč různými inicializacemi.

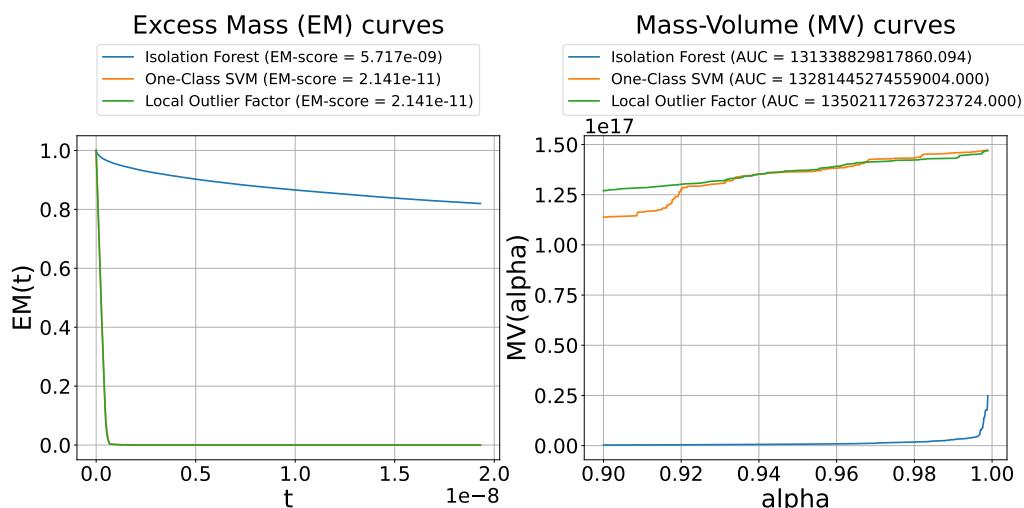
Výsledky těchto ověřovacích experimentů potvrzují, že zvolené konfigurace vykazují konzistentní výkon a lze je považovat za obecně vhodné. Pro následné porovnání výkonnosti modelů byly implementovány vizualizace metrik *EM* a *MV*, inspirované přístupem uvedeným v práci [47]. Tyto vizualizace umožňují detailní srovnání jednotlivých přístupů. Současně byla v rámci platformy **Weights & Biases** využita možnost agregace výsledků — jak napříč iteracemi *5-fold křízové validace*, tak i napříč celou testovací sadou (šesti dny a produkty).

Vzhledem k množství dostupných výstupů byla do této kapitoly opět zařazena pouze reprezentativní podmnožina výsledků. Kompletní sada vizualizací je dostupná v repozitáři ve složkách `/img/grid_search` a `/img/eval`. Na všech následujících obrázcích je použito Security ID **457882**.

#### 4.4.1 Srovnání referenčních modelů

Jak již bylo zmíněno v předchozích částech, model *IF* byl jediným referenčním modelem, který vykazoval relevantní výsledky. Ostatní referenční přístupy – *OCSVM* a *LOF* – výrazně zaostávaly.

Obrázek 4.18 znázorňuje porovnání těchto tří referenčních modelů pomocí křivek metrik *EM* a *MV*. Z průběhu křivek je zřejmá vysoká efektivita modelu *IF* a naopak nevhodnost ostatních dvou přístupů pro řešenou úlohu.

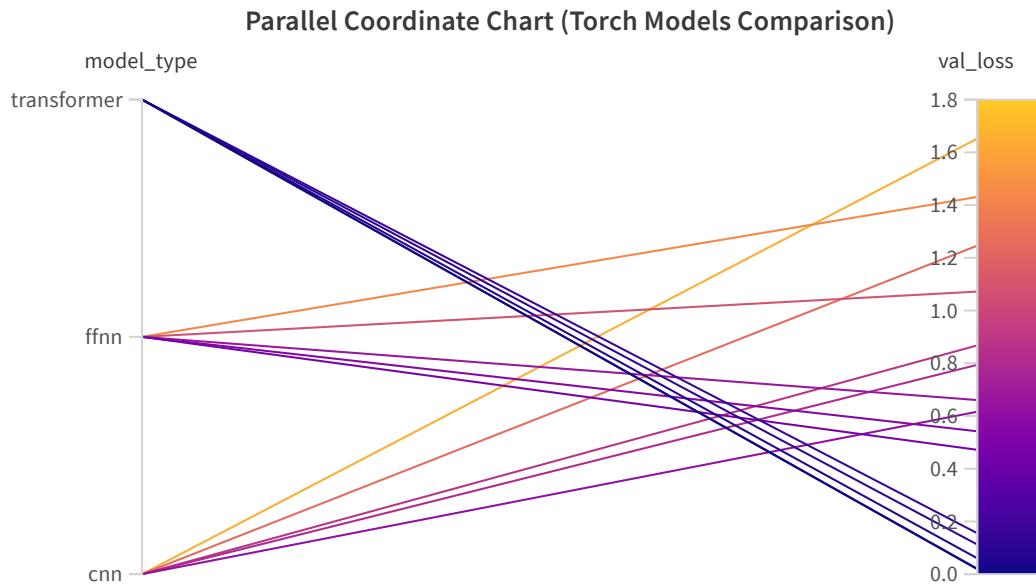


Obrázek 4.18: Porovnání nejlepších referenčních modelů pomocí křivek *EM* a *MV* (Security ID **457882**)

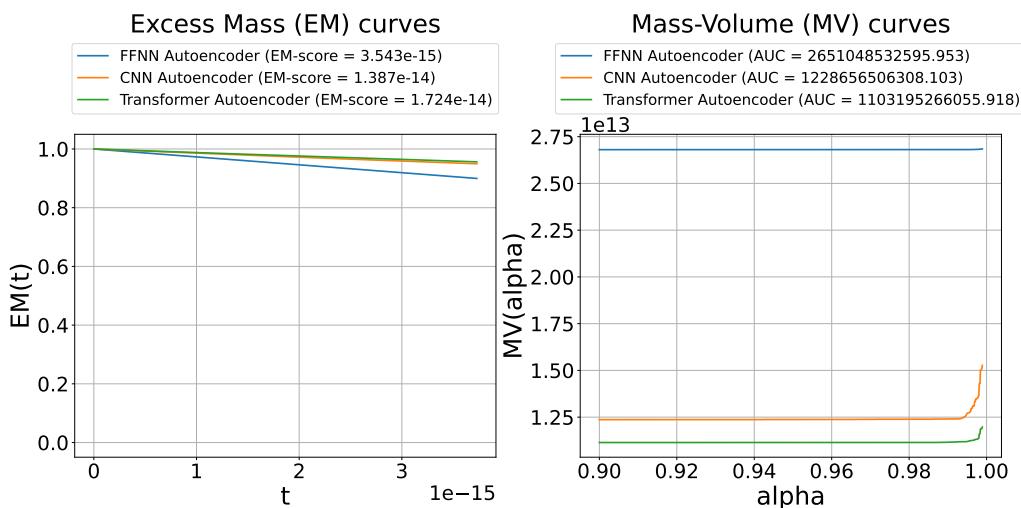
#### 4.4.2 Srovnání neuronových modelů

Jak bylo diskutováno dříve, *transformer* díky svému vysokému počtu trénovatelných parametrů zpravidla překonává ostatní architektury. Přesto je překvapivé, že *dopředné neuronové sítě (FFNN)* a *konvoluční neuronové sítě (CNN)* dosahují velmi vzájemně podobné úrovně výkonnosti.

V této části jsou uvedeny dva obrázky: první (Obrázek 4.19) zachycuje hodnoty validační ztrátové funkce nejlepších konfigurací modelů, druhý (Obrázek 4.20) porovnává neuronové modely na základě metrik *EM* a *MV*.



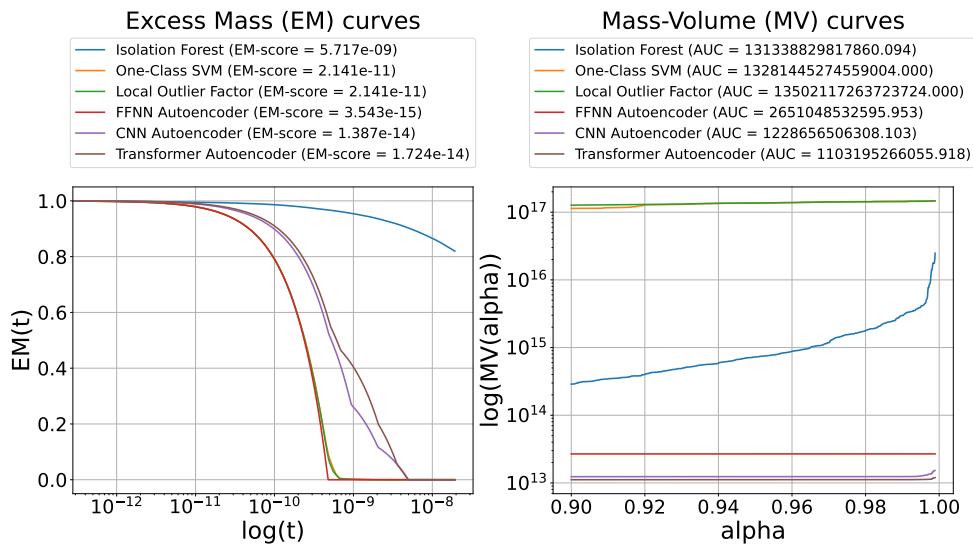
Obrázek 4.19: Porovnání nejlepších konfigurací jednotlivých architektur autoenkovodérů z hlediska validační ztrátové funkce (agregace napříč iteracemi 5-fold křížové validace)



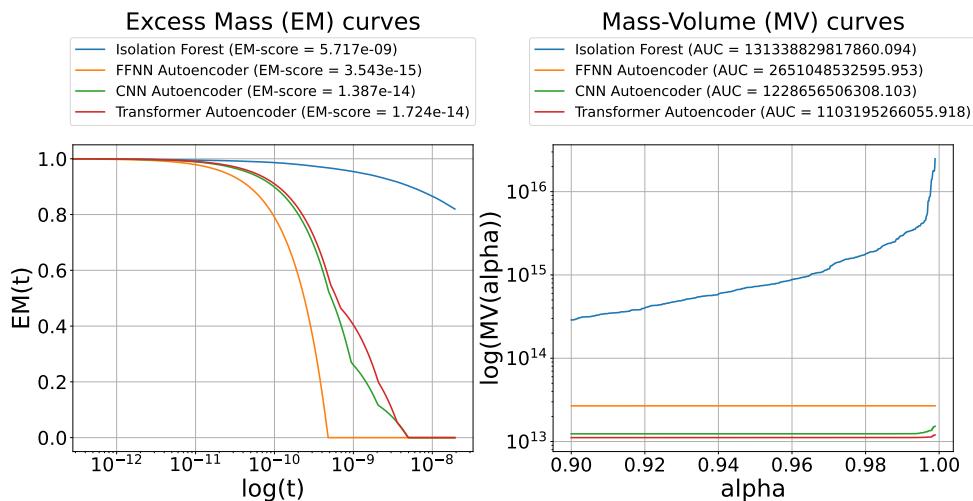
Obrázek 4.20: Porovnání nejlepších neuronových modelů pomocí křivek *EM* a *MV* (Security ID **4578882**)

### 4.4.3 Srovnání všech modelů

Vzhledem k tomu, že referenční modely neoptimalizují ztrátovou funkci, je možné srovnávat všechny modely pouze prostřednictvím metrik  $EM$  a  $MV$ . Vzhledem k velkým rozdílům mezi dvěma skupinami modelů byla pro Obrázky 4.21 a 4.22 zvolena semi-logaritmická měřítka (u  $EM$  je logaritmické měřítko na vodorovné ose, u  $MV$  je logaritmické měřítko na ose svislé). Jelikož *OCSVM* a *LOF* výrazně zaostávají, byly v Obrázku 4.22 tyto modely vynechány, aby bylo možné detailněji porovnat výkonnéjší přístupy.



Obrázek 4.21: Srovnání všech modelů na křivkách  $EM$  a  $MV$  (semi-logaritmické měřítka) (Security ID 4578882)



Obrázek 4.22: Porovnání modelů *IF*, *FFNN*, *CNN* a *Transformer* pomocí křivek  $EM$  a  $MV$  (semi-logaritmické měřítka) (Security ID 4578882)

Z uvedených obrázků je patrné, že model *IF* překonal i *autoenkovodérové* přístupy v rámci metriky *EM*. To naznačuje, že dokáže nejfektivněji ze všech modelů koncentrovat běžná (*normální*) data do kompaktních oblastí. V metrice *MV* již tak dominantní není, přesto si tento model udržel svou relevanci a byl zařazen do finálních výsledků.

## 4.5 Detekované anomálie

Výstupy jednotlivých modelů byly pomocí *prahování* převedeny na binární klasifikaci časových značek – tedy na rozhodnutí, zda se daný okamžik v čase jeví jako *anomální*. Použitý práh vycházel z předpokládané míry ***kontaminace*** dat anomáliemi, která byla nastavena na 1 %. To znamená, že 1 % nejvyšších hodnot skóre anomálnosti bylo označeno jako anomálie.

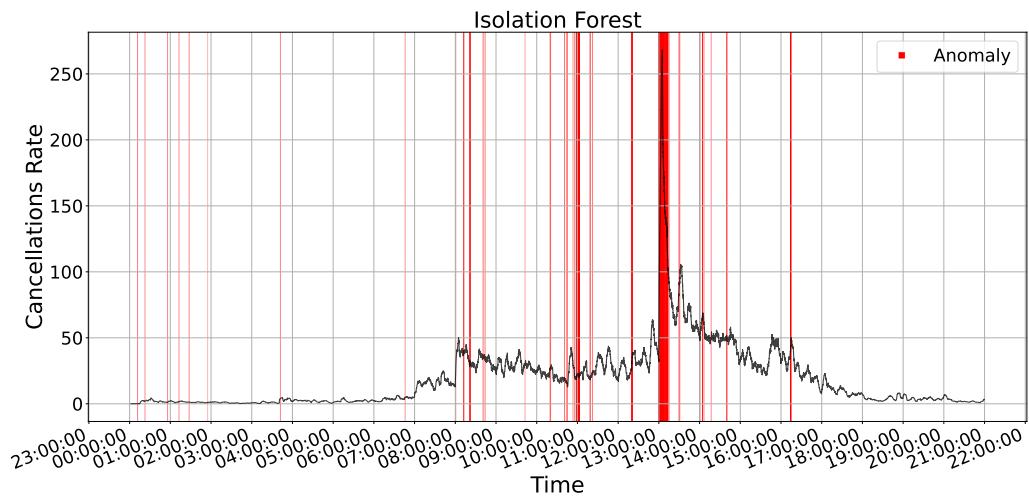
Pro účely vizualizace byla tato detekce znázorněna pomocí červených pruhů v časových grafech. Intenzita červené barvy byla dále upravena na základě normalizovaných hodnot skóre anomálnosti, které byly převedeny do rozsahu [0, 1] a použity jako alfa kanál. Díky tomu je možné vizuálně odlišit místa s vysokou mírou jistoty od těch, kde byl model méně přesvědčený – výrazně červené oblasti tedy značí silné podezření na přítomnost anomálie, zatímco méně syté oblasti mohou být výsledkem hromadění více slabších indikací.

Jelikož bylo vygenerováno velké množství vizualizací (řádově stovky), byl pro ilustraci zvolen reprezentativní případ se Security ID **5578483**, konkrétně v dimenzi *Cancellations Rate*. Tato dimenze byla empiricky identifikována jako klíčová pro detekci *spoofingu*, navíc se vyskytuje jak v původních datech, tak i v datech po redukci dimenzionality, což umožňuje jejich přímé porovnání. Kompletní sada výstupů je dostupná v repozitáři ve složce `/img/anomaly_detection`.

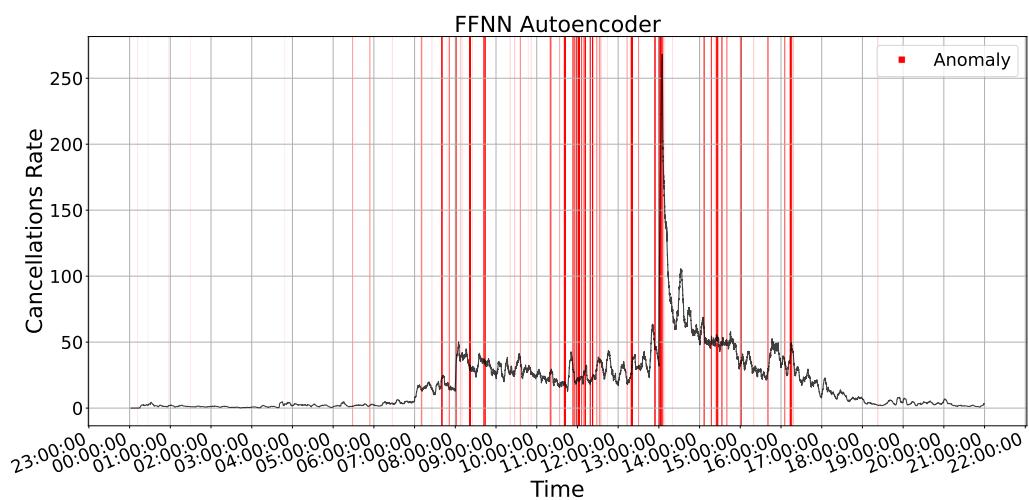
Detekce pomocí modelů *OCSVM* a *LOF* nejsou v této sekci uvedeny, neboť jejich výstupy nedávaly smysluplné výsledky – modely často označovaly například začátky či konce dnů, což lze považovat za chybnou interpretaci anomálnosti. Tyto přístupy se tedy ukázaly jako nevhodné pro řešený problém.

V následující sérii čtyř grafů (Obrázky 4.23, 4.24, 4.25 a 4.26) je možné pozorovat, že modely *FFNN* a *CNN* vykazují velmi podobné výsledky, liší se pouze v drobných nuancích. Za nejlepší v rámci této úlohy se však jeví modely *IF* a *Transformer*, které poskytují nejvíce konzistentní a přesvědčivé detekce.

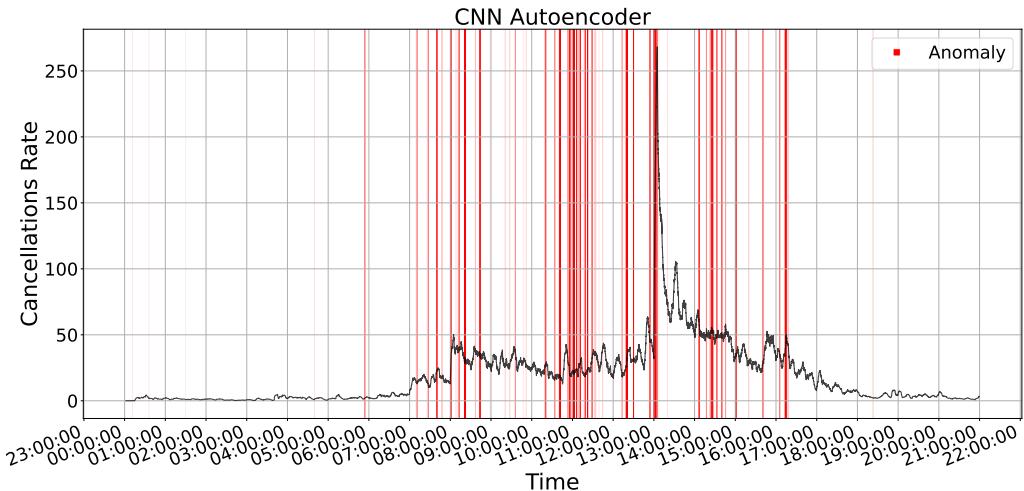
#### 4 Výsledky a diskuze



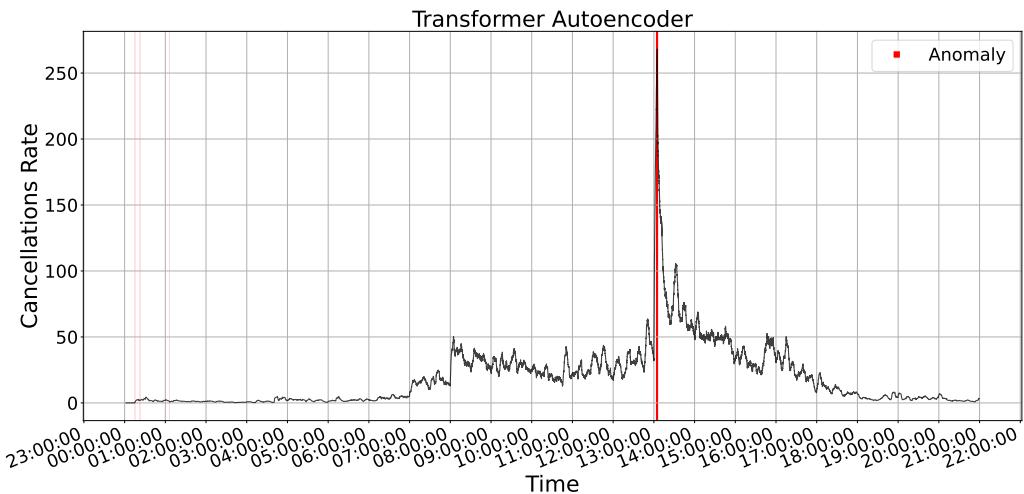
Obrázek 4.23: Detekované anomálie modelem IF (Security ID **5578483**)



Obrázek 4.24: Detekované anomálie modelem FFNN (Security ID **5578483**)



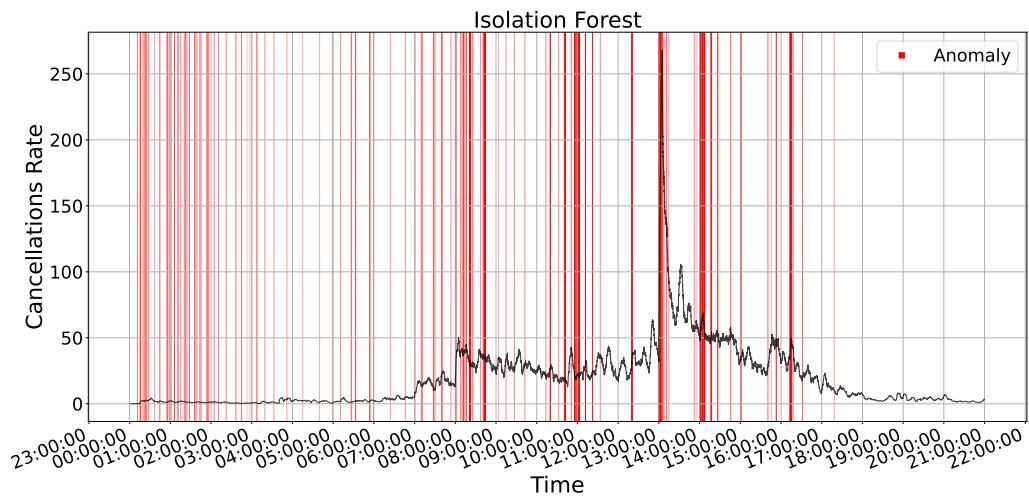
Obrázek 4.25: Detekované anomálie modelem CNN (Security ID 5578483)



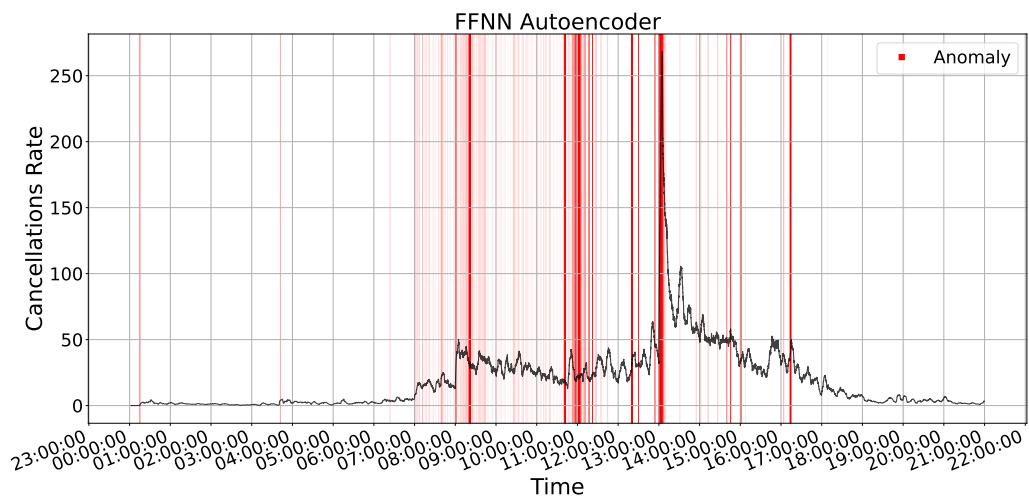
Obrázek 4.26: Detekované anomálie modelem Transformer (Security ID 5578483)

Druhá série grafů (Obrázky 4.27, 4.28, 4.29 a 4.30) zobrazuje výsledky na datech po redukci dimenzionality. Ve všech případech modely detekují zdánlivě větší množství anomálií s vyšší mírou jistoty. Vzhledem k absenci anotací však není možné jednoznačně určit, zda se skutečně jedná o relevantní anomálie. Za zmínu stojí, že modely *IF* a *Transformer* často označují jako anomální oblast začátku dne. Tento jev má své opodstatnění – na začátku dne dochází k načítání stavů z předchozího dne, což způsobuje prudké změny v datech. I přesto, že se tyto změny mohou jevit jako anomálie, pro detekci *spoofingu* se pravděpodobně nejedná o oblast s vysokou relevancí. Nelze však zcela vyloučit opak – právě v těchto obdobích by se mohly skrývat největší pokusy o manipulace.

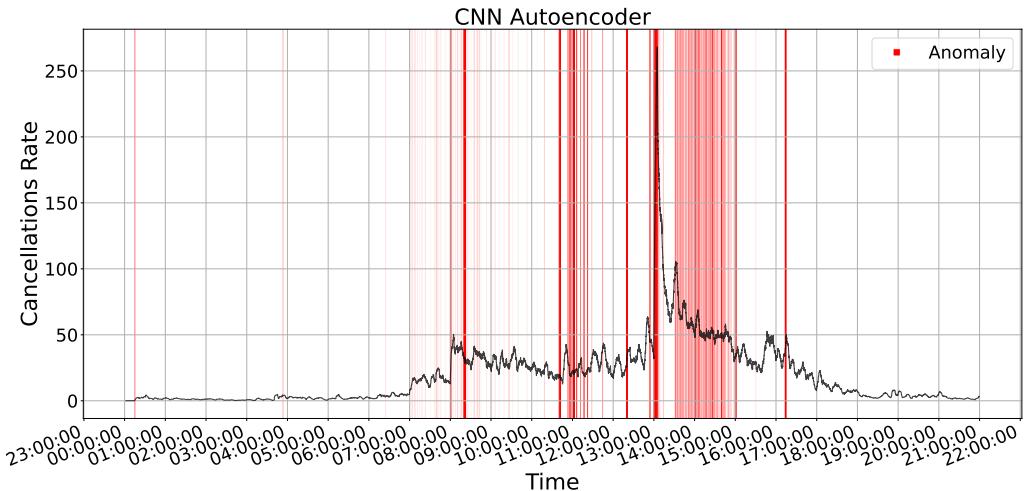
#### 4 Výsledky a diskuze



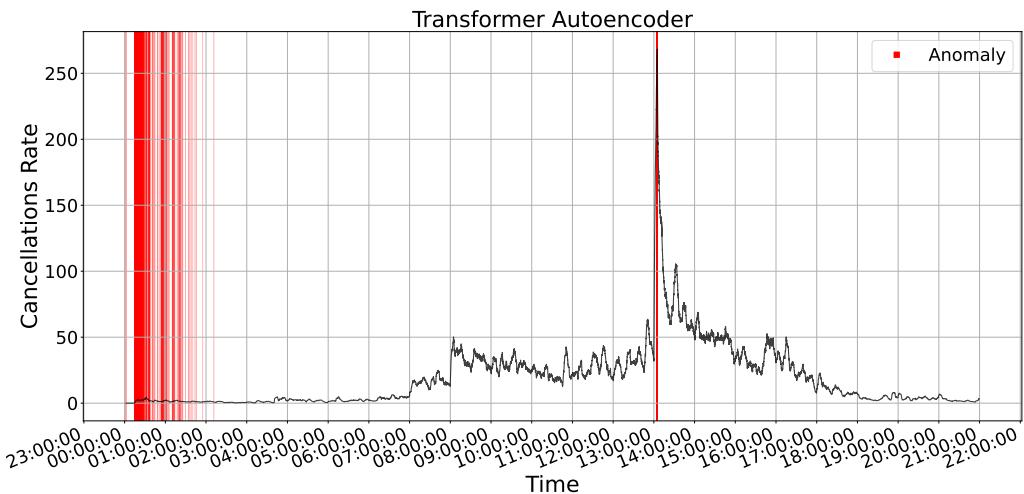
Obrázek 4.27: Detekované anomálie modelem *IF* na redukovaných datech  
(Security ID **5578483**)



Obrázek 4.28: Detekované anomálie modelem *FFNN* na redukovaných datech  
(Security ID **5578483**)



Obrázek 4.29: Detekované anomálie modelem *CNN* na redukovaných datech  
(Security ID **5578483**)

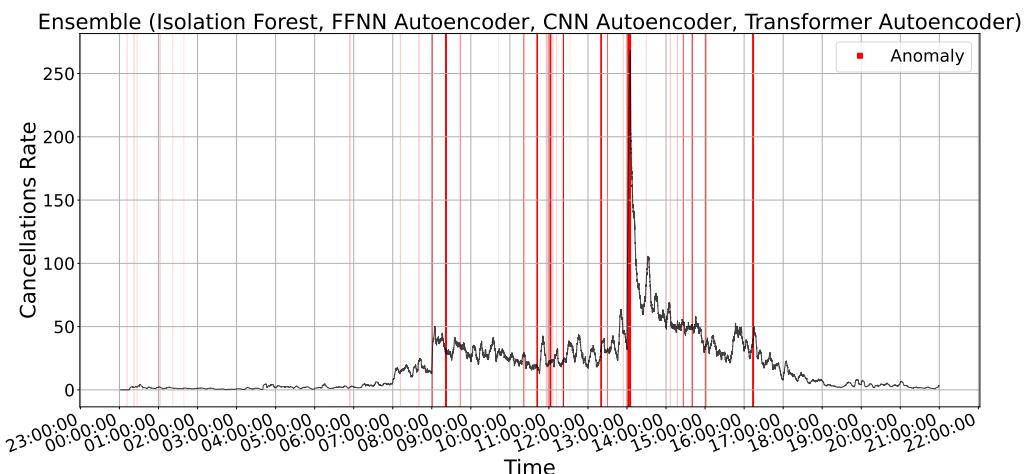


Obrázek 4.30: Detekované anomálie modelem *Transformer* na redukovaných datech  
(Security ID **5578483**)

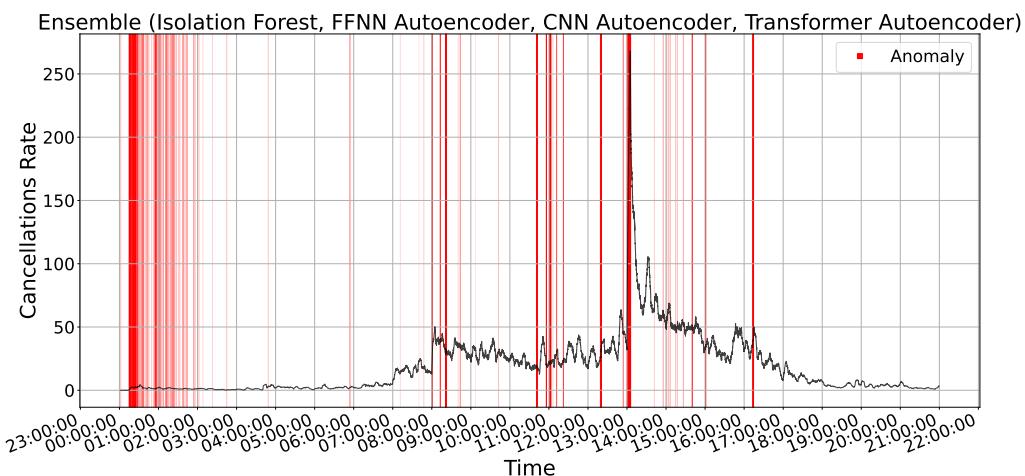
### 4.5.1 Kombinace modelů

V závěru práce byl vytvořen **soubor modelů** (*Ensemble*) složený z výstupů čtyř nejúspěšnějších modelů. Testována byla varianta váženého průměru skóre jednotlivých modelů, přičemž váhy byly odvozeny na základě dosažených hodnot metrik *EM* a *MV*. Výsledky se však ukázaly jako velmi podobné těm, kterých bylo dosaženo pomocí jednoduchého neváženého průměru. Z důvodu srozumitelnosti a nižší výpočetní složitosti byl proto zvolen tento jednodušší přístup – normalizace skóre jednotlivých modelů, jejich zprůměrování a výsledný průměr byl považován za skóre *souborového modelu*.

Na následujících Obrázcích 4.31 a 4.32 jsou zobrazeny výsledky této kombinace opět pro Security ID **5578483** a dimenzi **Cancellations Rate**. Místa s výrazně sytou červenou barvou odpovídají vysoké shodě názorů jednotlivých modelů – tedy vyšší míře jistoty o výskytu anomálie. Tato shoda může indikovat i výskyt *spoofingu*.

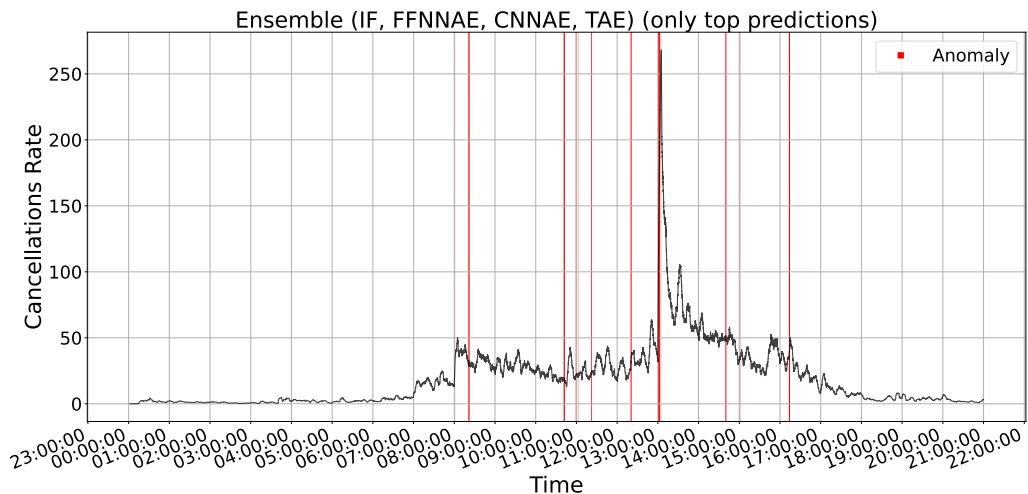


Obrázek 4.31: Detekované anomálie souborem modelů (Security ID **5578483**)

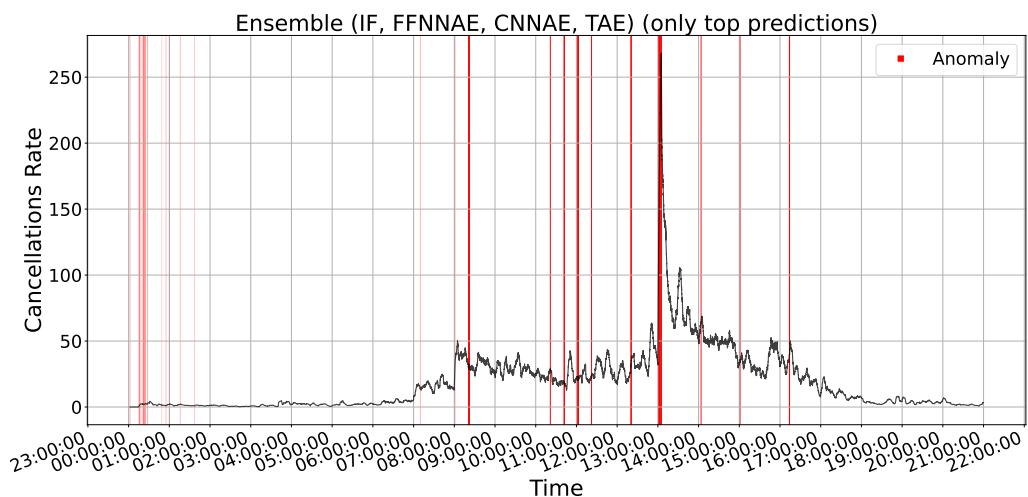


Obrázek 4.32: Detekované anomálie souborem modelů na redukovaných datech (Security ID **5578483**)

Pro dosažení vyšší selektivity byla navíc výsledná skóre přefiltrována pomocí 99,9. percentilu, což znamená, že byly vizualizovány pouze nejvýraznější detekce – tedy 0,1 % časových značek s nejvyšším skóre – Obrázky 4.33 a 4.34. Tyto výsledky byly následně předány expertům z **Deutsche Börse AG** k detailnímu přezkoumání a vyhodnocení, zda se skutečně může jednat o *spoofing*. Vzhledem k neanotované povaze vstupních dat však v rámci této práce bohužel nebylo možné stanovit přesné hodnoty falešně pozitivních či falešně negativních detekcí.



Obrázek 4.33: Přefiltrované detekce anomálií souborem modelů (Security ID **5578483**)



Obrázek 4.34: Přefiltrované detekce anomálií souborem modelů na redukovaných datech (Security ID **5578483**)



# Závěr

5

Those who can imagine anything, can create the impossible.

---

*Alan Turing (1912–1954), Mathematician, Logician,  
Cryptanalyst, Father of Computer Science, Known for  
breaking the Enigma Code*

Cílem této práce bylo navrhnut a experimentálně ověřit metody pro detekci anomálií ve *vysokofrekvenčních obchodních datech* s důrazem na rozpoznání manipulační techniky známé jako *spoofing*. Vzhledem k neexistenci anotovaných datových sad pro tento specifický typ finančního podvodu bylo zvoleno řešení založené na metodách učení bez učitele.

Po důkladném předzpracování dat a analýze relevantních metrik bylo navrženo a implementováno šest hlavních přístupů ke generování skóre anomálnosti: izolační les (*Isolation Forest, IF*), lokální faktor odlehlosti (*Local Outlier Factor, LOF*), jednotřídní SVM (*One-Class SVM, OCSVM*), autoenkovodér s plně propojenou architekturou (*FFNN*), konvoluční autoenkovodér (*CNN*) a autoenkovodér s *Transformer* architekturou. Na základě metrik *Excess Mass (EM)* a *Mass Volume (MV)* byly identifikovány modely s nejvyššími prediktivními schopnostmi. Modely *OCSVM* a *LOF* nebyly pro daný problém vhodné, neboť vykazovaly chybnou interpretaci anomálnosti.

Výsledky jednotlivých modelů byly převedeny na binární rozhodnutí pomocí *prahování* a následně vizualizovány. Za zvlášť cennou se ukázala schopnost modelů *IF* a *Transformer* zachytit konzistentní a přesvědčivé indikace možných anomálií, zejména v dimenzi *Cancellations Rate*, která je typicky spojována se *spoofingem*.

Následná kombinace výstupů nejúspěšnějších modelů do jednoho *souborového modelu (Ensemble)* vedla k vytvoření robustnějšího nástroje pro detekci anomálií. Tato kombinace byla provedena pomocí jednoduchého zprůměrování

normalizovaných skóre, což poskytlo kvalitní výsledky bez zbytečného navýšení výpočetní složitosti.

Z hlediska praktického využití lze konstatovat, že navržený přístup umožňuje efektivní zúžení prostoru pro následnou manuální analýzu – tedy slouží jako nástroj pro výběr potenciálně rizikových oblastí v datech. Výsledky byly předány expertům z **Deutsche Börse AG**, kteří je využijí jako podpůrný podklad při manuálním vyhodnocení podezřelých obchodních vzorců.

Hlavním omezením práce zůstává absence přesného ohodnocení výkonnosti detekce ve smyslu typických metrik jako *přesnost* (ať už ve smyslu *Accuracy*, tak i *Precision*), *úplnost* (*Recall*) či *F1 skóre*, neboť vstupní data neobsahovala anotace známých případů *spoofingu*. Pro budoucí výzkum by bylo vhodné zaměřit se na vytvoření nebo získání takových dat, případně využít simulovaných datových sad s předem definovanými anomáliemi.

Navržený rámec je však flexibilní a snadno rozšířitelný o další modely i metriky. Ukázalo se, že i v prostředí s minimem předchozích znalostí o výskytu anomálií lze dosáhnout užitečných výstupů, které mají potenciál výrazně přispět k odhalování nekalých praktik na finančních trzích.

# Uživatelská příručka

A

Celý projekt je veřejně dostupný na platformě **GitHub** na adrese [https://github.com/SpeekeR99/DP\\_2024\\_2025\\_Zappe](https://github.com/SpeekeR99/DP_2024_2025_Zappe)

Smlouva o poskytnutí dat pro akademické účely od **Deutsche Börse AG** ze dne 14. 11. 2022 nedovoluje zveřejnění vstupních dat (*Non-Disclosure Agreement*).

Případná aktualizace podmínek by byla zveřejněna v repozitáři v adresáři data.

Většina uvedených výpisů v této kapitole je pro operační systém **Linux**. Pro jiné operační systémy (**Windows**, **macOS**) je však postup analogický.

## Předpoklady

Pro správné spuštění programu je zapotřebí mít nainstalovaný:

- Python 3 (doporučená verze **Python 3.11**)
- Uživatelem preferovaný **správce balíčků** (doporučen je standardní **pip**)

Interpret jazyka **Python** si lze stáhnout z oficiálních stránek <https://www.python.org>. Standardní **správce balíčků pip** je nedílnou součástí instalace.

## Naklonování repozitáře

Jelikož repozitář obsahuje tzv. *podmoduly*, je zapotřebí projekt správně naklonovat tzv. *rekurzivním klonováním*. Ukázka správného postupu je uvedena ve Výpisu A.1.

Výpis A.1: Ukázkový výpis při klonování repozitáře

```
1 uzivatel@pocitac:~$ git clone --recursive  
      https://github.com/SpeekeR99/DP_2024_2025_Zappe.git  
2 ... (výpis průběhu klonování)  
3 uzivatel@pocitac:~$
```

## Vytvoření a aktivace virtuálního prostředí

Doporučuje se vytvořit si pro projekt samostatné virtuální prostředí, ve kterém budou instalovány všechny závislosti. Tím se předejde možným konfliktům s jinými projekty či globálně nainstalovanými balíčky.

Ukázka vytvoření a aktivace virtuálního prostředí pomocí standardního modulu `venv` je uvedena ve Výpisech A.2 a A.3.

Výpis A.2: Ukázkový výpis tvorby a aktivace virtuálního prostředí (**Windows**)

```
1 C:\Users\Uzivatel\DP_2024_2025_Zappe>python -m venv venv  
2  
3 C:\Users\Uzivatel\DP_2024_2025_Zappe>call venv\Scripts\activate.bat  
4  
5 (venv) C:\Users\Uzivatel\DP_2024_2025_Zappe>
```

Výpis A.3: Ukázkový výpis tvorby a aktivace virtuálního prostředí (**Linux**)

```
1 uzivatel@pocitac:~/DP_2024_2025_Zappe$ python3 -m venv venv  
2 uzivatel@pocitac:~/DP_2024_2025_Zappe$ source venv/bin/activate  
3 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

**Linux:** aktivaci virtuálního prostředí je umožněno používání příkazů `python` a `pip` namísto `python3` a `pip3`.

## Instalace závislostí

Po úspěšném naklonování se v *kořenovém adresáři* projektu nachází soubor `requirements.txt`, který obsahuje potřebné knihovny. Jejich instalaci lze provést příkazem uvedeným ve Výpisu A.4.

Výpis A.4: Ukázkový výpis instalace závislostí

```
1 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$ pip install -r  
     requirements.txt  
2 ... (výpis průběhu instalace závislostí)  
3 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

## Spuštění

Jelikož je aplikace napsána v interpretovaném jazyce **Python**, není nutné program nejdříve kompilovat.

**Veškeré skripty předpokládají spuštění z *kořenového adresáře* projektu!** Při spuštění z jiného adresáře může dojít k chybám kvůli relativním cestám.

## Stažení dat

Nejprve je nutné získat vstupní data – v rámci této práce byl využit proprietární nástroj A7. Skript `/src/A7/download_eobi.py` slouží ke stažení požadovaného souboru ve formátu `JSON` a očekává čtyři parametry v následujícím pořadí:

1. **Market ID** – např. XEUR
2. **Datum** ve formátu YYYYMMDD – např. 20191202
3. **Market Segment ID** – např. 688
4. **Security ID** – např. 4128839

Ukázka úspěšného spuštění je uvedena ve Výpisu A.5. Po úspěšném dokončení by se měl v adresáři `/data` objevit stažený soubor.

Pro korektní fungování skriptů komunikujících s A7 API je nutné upravit zdrojový kód na rádcích 25 a 26, kde se nachází uživatelské ID a API klíč. Alternativně lze tyto údaje uložit do souboru `a7token.txt` v kořenovém adresáři projektu.

Výpis A.5: Ukázkový výpis spuštění stažení dat

```

1 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$ python
      src/A7/download_eobi.py XEUR 20191202 688 4128839
2 ... (výpis průběhu stahování)
3 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

## Předzpracování dat

Po úspěšném stažení vstupního `JSON` souboru je nutné provést jeho převod a případné rozšíření dat. K tomu slouží dva skripty umístěném ve složce `/src/data_preprocess`:

- `json-detailed2lobster.py` – převádí vstupní `JSON` soubor surových zpráv do formátu `LOBSTER`.
- `augment_lobster.py` – provádí doplnění a rozšíření dat – výpočet dodatečných metrik.

Oba skripty očekávají čtyři základní parametry ve stejném pořadí jako u předchozího kroku:

1. **Market ID** — např. XEUR
2. **Datum** ve formátu YYYYMMDD — např. 20191202
3. **Market Segment ID** — např. 688
4. **Security ID** — např. 4128839

Spuštění těchto skriptů je obdobné jako v předchozím Výpisu A.5. Výsledné soubory budou uloženy do složky /data, kde slouží jako vstup pro další fázi zpracování – trénování modelů.

## Spuštění trénování modelů

Pro trénování detekčních modelů slouží dva hlavní skripty umístěné ve složce /src/anomaly\_detection/models:

- `autoencoder.py` — trénuje varianty *autoenkovodérů* (plně propojený, konvoluční, transformer).
- `if_ocsvm_lof.py` — spouští modely typu *izolační les*, *jednotřídní SVM* a *lokální faktor odlehlosti*.

Spuštění skriptů vyžaduje několik parametrů, které určují jednak konkrétní datový soubor, jednak konfiguraci samotného modelu.

Mezi společné parametry patří:

- `--market_id` — např. XEUR
- `--date` — datum ve formátu YYYYMMDD, např. 20191202
- `--market_segment_id` — např. 688
- `--security_id` — např. 4128839

Dále se parametry liší podle typu modelu.

### Příklad parametrisace pro autoenkovodér.

- `--model_type` — typ modelu: ffnn, cnn nebo transformer
- `--epochs`, `--kfolds`, `--batch_size`, `--lr` — běžné trénovací parametry
- `--seq_len`, `--latent_dim` — specifické pro sekvenční modely
- `--seed` — pro zajištění reprodukovatelnosti

## Příklad parametrizace pro IF, OCSVM a LOF.

- `--model_type` – typ modelu: `if`, `ocsvm` nebo `lof`
- `--kfolds` – počet iterací pro křížovou validaci
- Parametry specifické pro jednotlivé modely:
  - IF: `--n_estimators`, `--max_samples`, `--max_features`
  - OCSVM: `--gamma`
  - LOF: `--n_neighbors`
- `--seed` – pro zajištění reprodukovatelnosti

Výsledky experimentů se automaticky ukládají do složky `/res`, zatímco natrénované modely jsou serializovány a ukládány do složky `/models` ve formátu `.pckl`.

Ukázkové spuštění obou skriptů může vypadat např. jako ve Výpisech A.6 a A.7.

Výpis A.6: Ukázkové spuštění modelu *CNN Autoencoder*

```

1 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$ python
src/anomaly_detection/models/autoencoder.py --market_id XEUR --date
20191202 --market_segment_id 688 --security_id 4128839 --model_type
cnn --epochs 500 --kfolds 5 --batch_size 32 --lr 1e-3 --seq_len 64 --
latent_dim 4
2 ... (výpis průběhu trénování)
3 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

Výpis A.7: Ukázkové spuštění modelu *Isolation Forest*

```

1 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$ python
src/anomaly_detection/models/if_ocsvm_lof.py --market_id XEUR --date
20191202 --market_segment_id 688 --security_id 4128839 --model_type
if --kfolds 5 --n_estimators 100 --max_samples auto --max_features
1.0 --seed 42
2 ... (výpis průběhu trénování)
3 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

## Soubor modelů a vizualizace

Kombinaci výsledků všech modelů spolu s jejich vizualizací zajišťuje skript `/src/anomaly_detection/models/ensemble.py`. Tento skript umožňuje načíst uložené výsledky z předchozího kroku, dále je kombinovat a generovat vizuální výstupy pro snadnější interpretaci chování jednotlivých modelů i celého *souboru modelů*.

Skript akceptuje široké spektrum parametrů, které zahrnují jak identifikaci konkrétního datového souboru, tak konfigurace jednotlivých modelů. Většina parametrů odpovídá těm, které byly použity při trénování modelů. Nově zde přibývají následující parametry:

- `--no_if`
- `--no_ocsvm`
- `--no_lof`
- `--no_ffnn`
- `--no_cnn`
- `--no_transformer`

Parametry s prefixem `no_` slouží k deaktivaci daného modelu. Díky tomu je možné snadno testovat různé kombinace metod bez nutnosti úprav zdrojového kódu.

Skript načítá vstupní data ze složky `/data` a mezivýsledky z trénování ze složky `/res`. Výstupní vizualizace jsou automaticky ukládány do příslušných pod adresářů složky `/img`, např. `/img/anomaly_detection`, `/img/eval` apod. Příklad spuštění `ensemble.py` na výše popsaných natrénovaných modelech vypadá dle vzoru Výpisu A.8.

Výpis A.8: Ukázkové spuštění souboru modelů

```
1 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$ python
    src/anomaly_detection/models/ensemble.py --market_id XEUR --date
    20210319 --market_segment_id 688 --security_id 5578483 --epochs 500 --
    kfolds 5 --if_n_estimators 100 --if_max_samples 0.1 --
    if_max_features 0.5 --no_ocsvm true --no_lof true --cnn_batch_size
    32 --cnn_lr 1e-3 --cnn_seq_len 64 --no_ffnn true --no_transformer
    true
2 (venv) uzivatel@pocitac:~/DP_2024_2025_Zappe$
```

# Pohled přes všechny dimenze

B

Všechny obrázky v této příloze jsou demonstrovány na Security ID **4128839**.

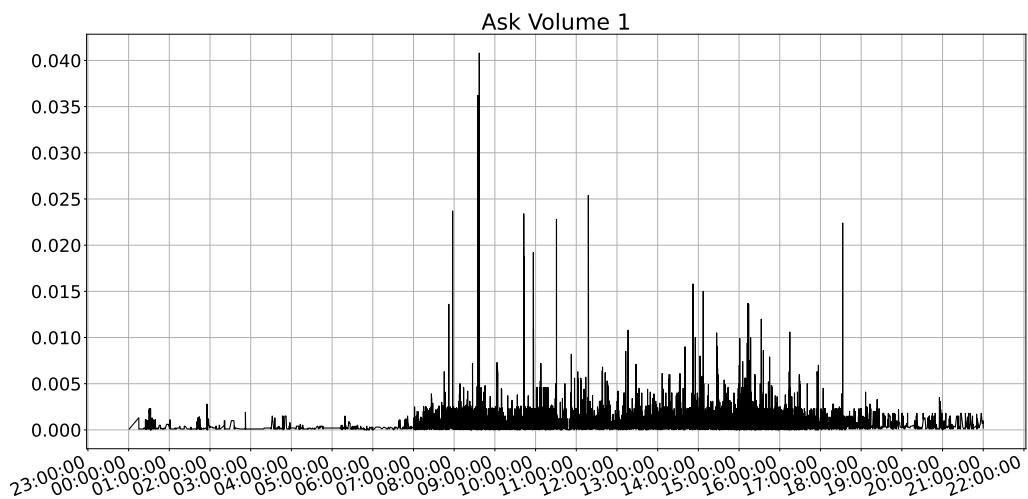
## Původní data



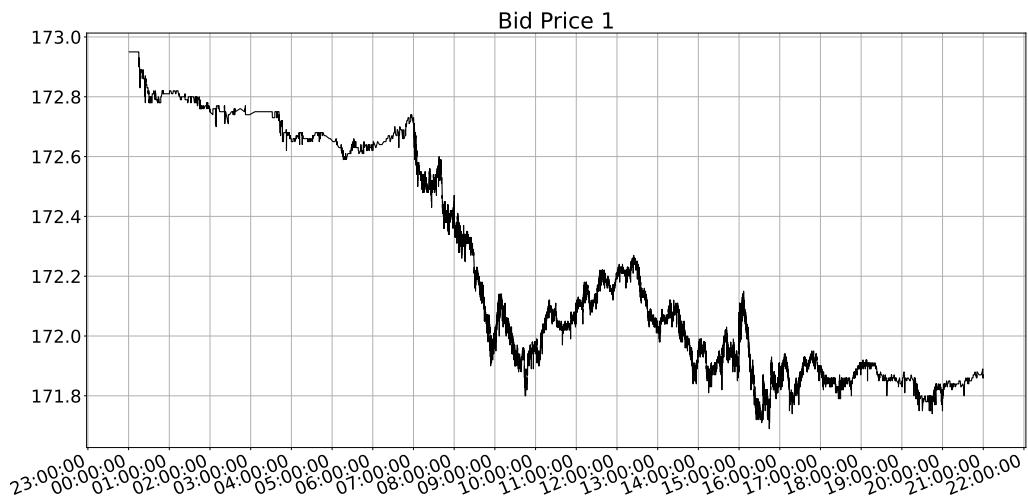
Obrázek B.1: Příklad průběhu *nejlepší poptávkové ceny* v čase (Security ID **4128839**)

*B Pohled přes všechny dimenze*

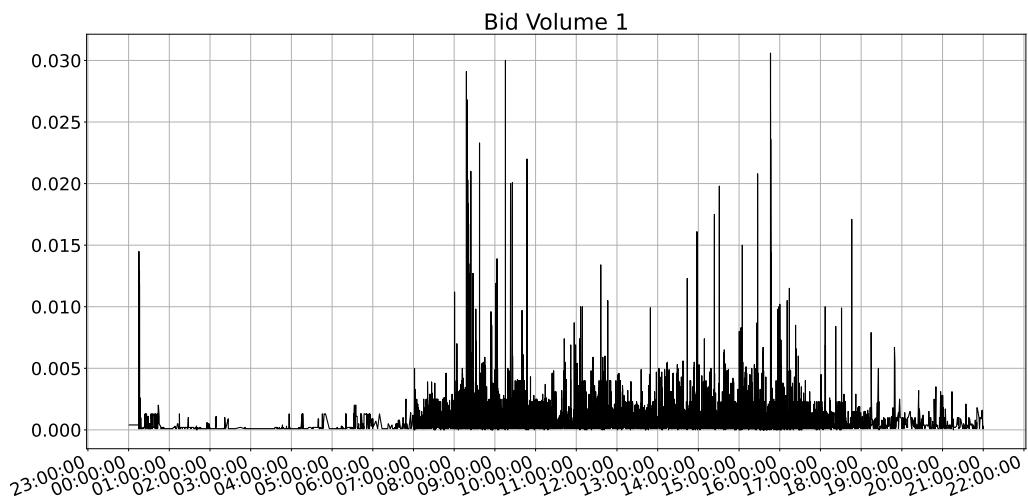
---



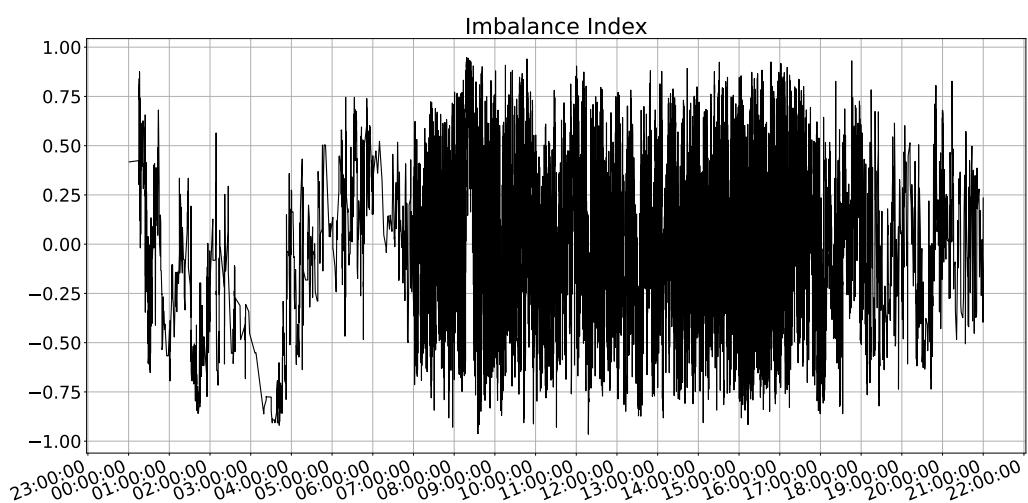
Obrázek B.2: Příklad průběhu *objemu nejlepších poptávek* v čase (Security ID **4128839**)



Obrázek B.3: Příklad průběhu *nejlepší nabídkové ceny* v čase (Security ID **4128839**)



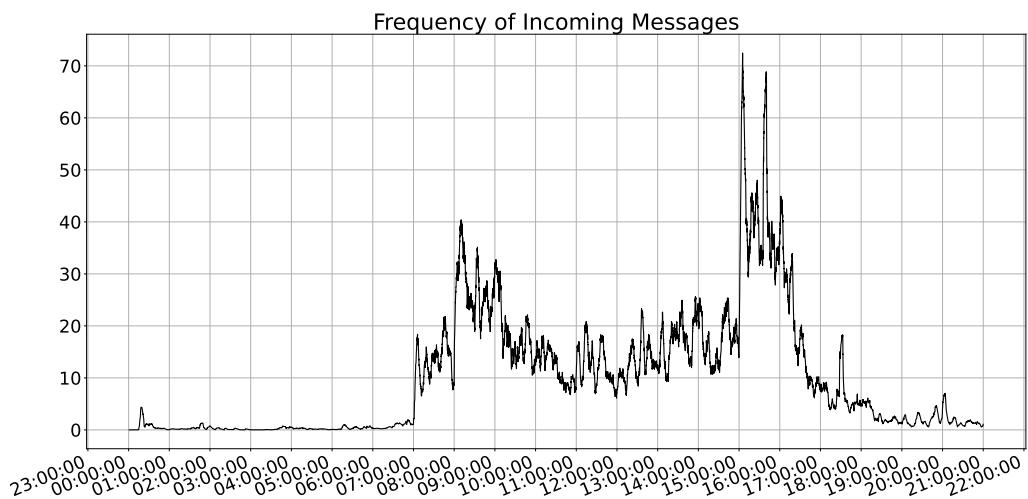
Obrázek B.4: Příklad průběhu objemu nejlepších nabídek v čase (Security ID **4128839**)



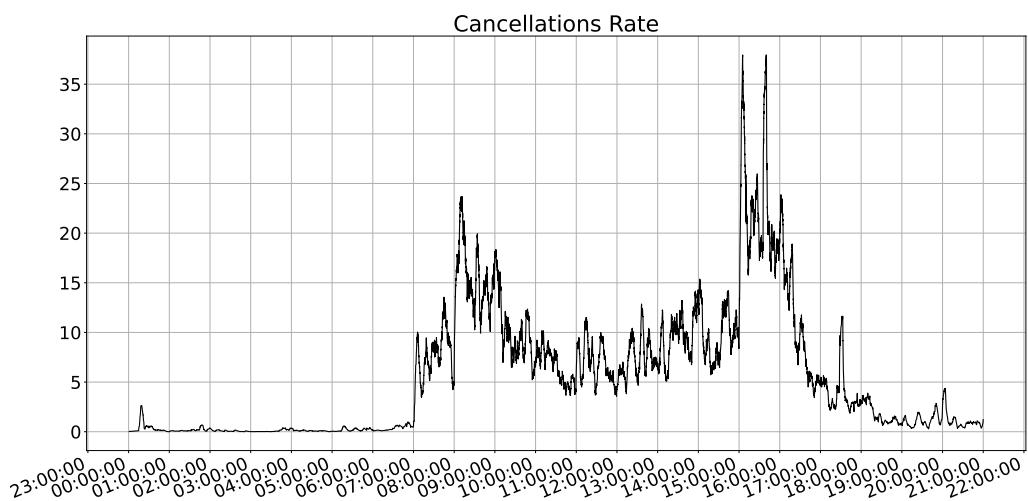
Obrázek B.5: Příklad průběhu imbalance indexu v čase (Security ID **4128839**)

*B Pohled přes všechny dimenze*

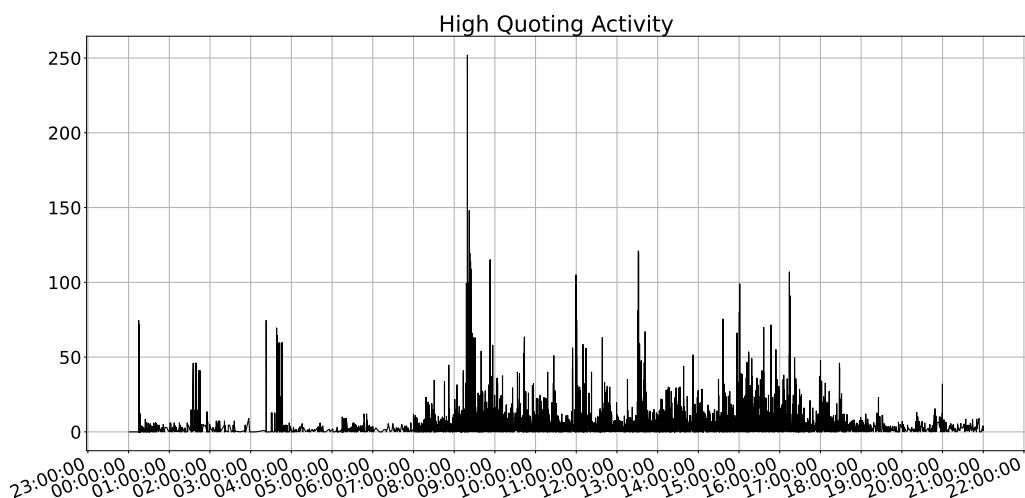
---



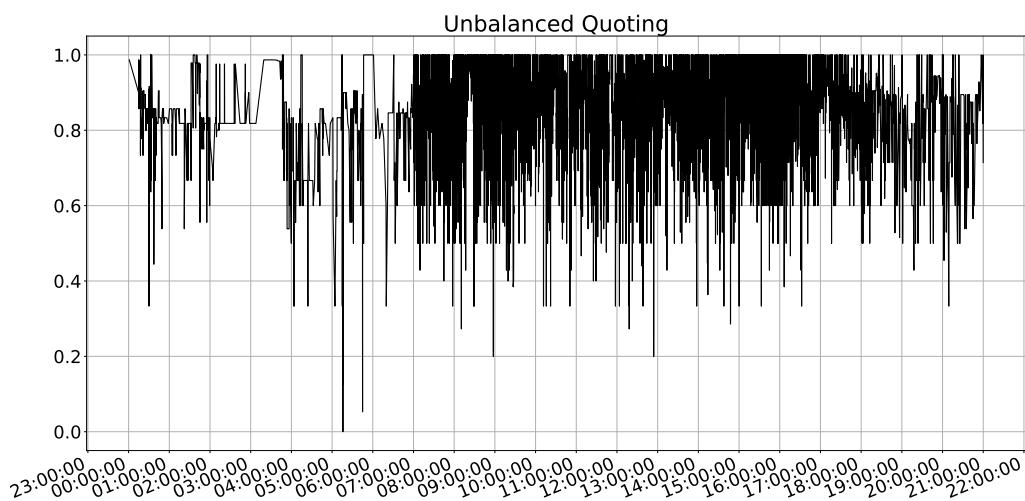
Obrázek B.6: Příklad průběhu frekvence příchozích zpráv v čase (Security ID **4128839**)



Obrázek B.7: Příklad průběhu frekvence rušení objednávek v čase (Security ID **4128839**)



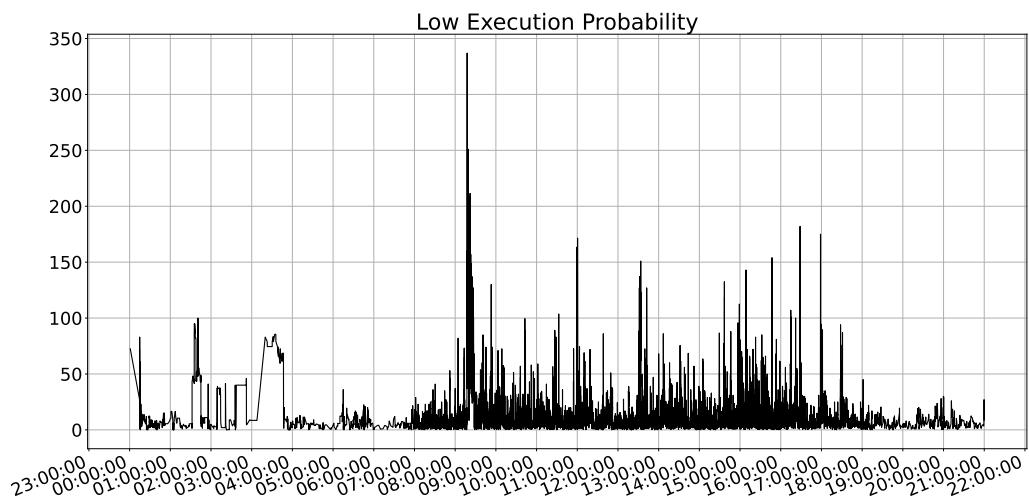
Obrázek B.8: Příklad průběhu dimenze „High Quoting Activity“ v čase (Security ID **4128839**)



Obrázek B.9: Příklad průběhu dimenze „Unbalanced Quoting“ v čase (Security ID **4128839**)

*B Pohled přes všechny dimenze*

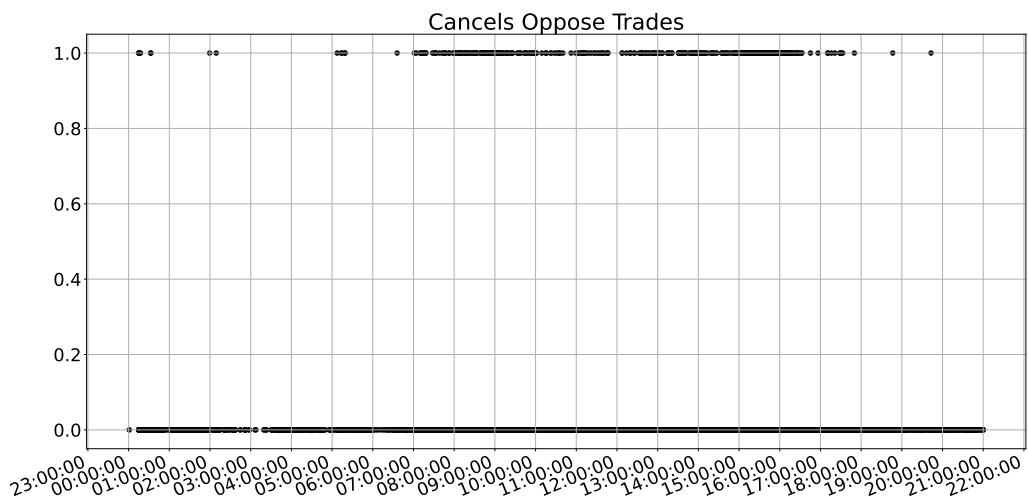
---



Obrázek B.10: Příklad průběhu dimenze „Low Execution Probability“ v čase  
(Security ID **4128839**)

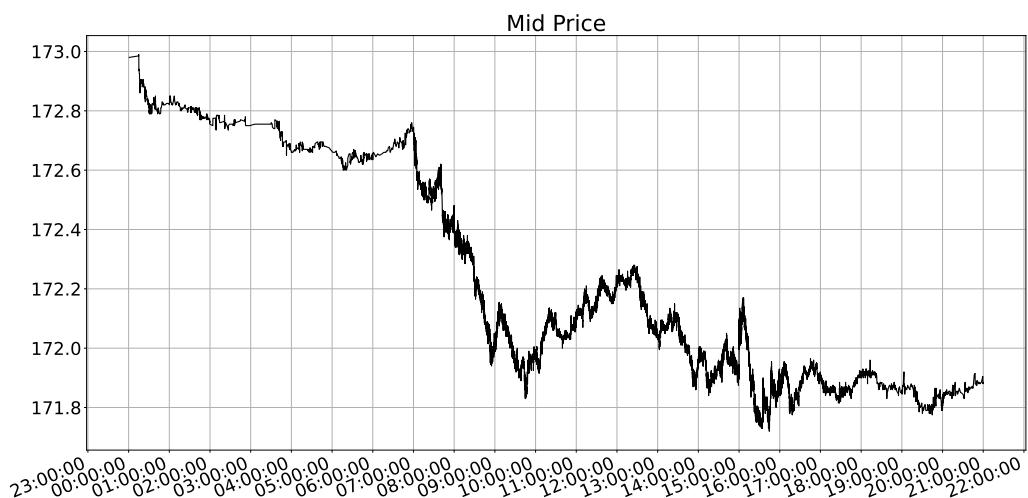


Obrázek B.11: Příklad průběhu dimenze „Trades Oppose Quotes“ v čase  
(Security ID **4128839**)



Obrázek B.12: Příklad průběhu dimenze „Cancels Oppose Trades“ v čase  
(Security ID **4128839**)

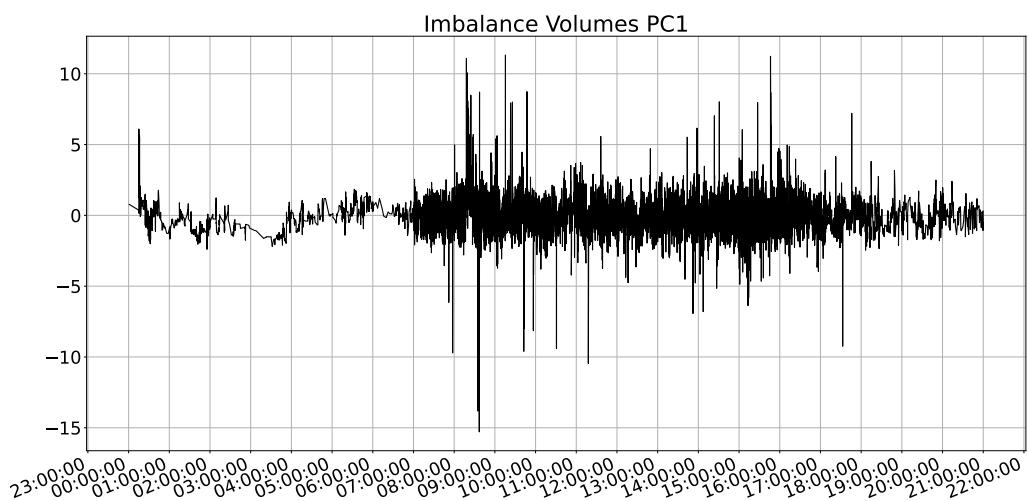
## Po redukci dimenzi



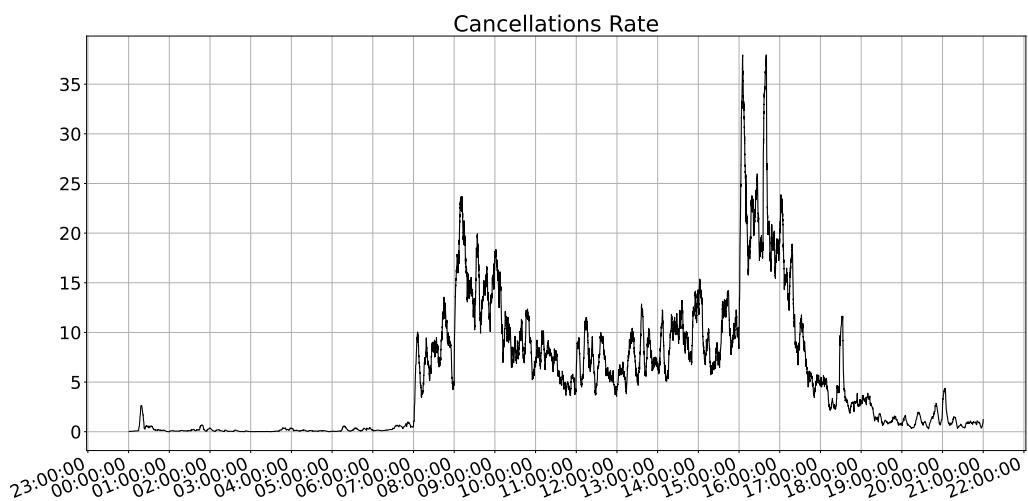
Obrázek B.13: Příklad průběhu střední ceny v čase (Security ID **4128839**)

*B Pohled přes všechny dimenze*

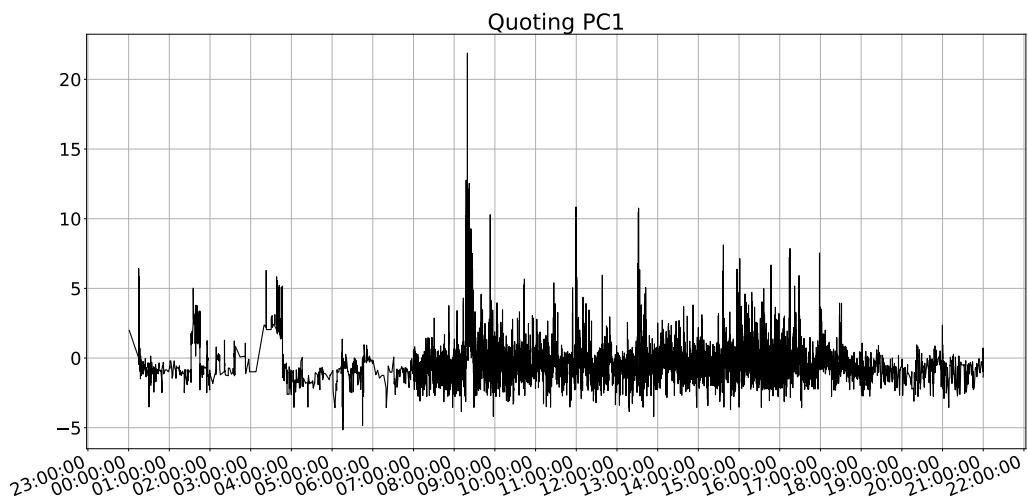
---



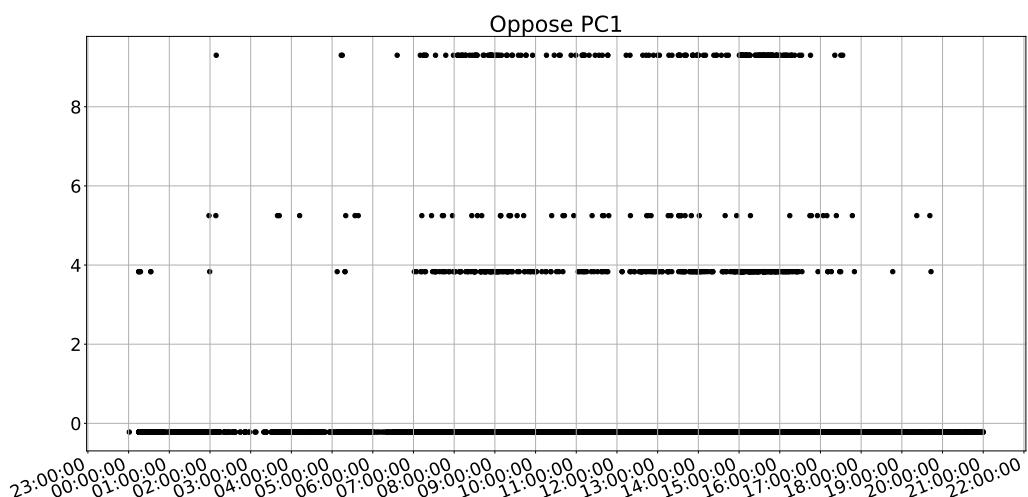
Obrázek B.14: Příklad průběhu dimenze „*Imbalance Volumes PC1*“ v čase  
(Security ID **4128839**)



Obrázek B.15: Příklad průběhu frekvence rušení objednávek v čase (Security ID **4128839**)



Obrázek B.16: Příklad průběhu dimenze „Quoting PC1“ v čase (Security ID **4128839**)



Obrázek B.17: Příklad průběhu dimenze „Oppose PC1“ v čase (Security ID **4128839**)



# Bibliografie

1. RIESCHEL, Emilia. *Detecting spoofing in financial markets: An unsupervised anomaly detection approach*. 2024. Dostupné z URL: <https://www.diva-portal.org/smash/get/diva2:1885077/FULLTEXT01.pdf>.
2. VERYZHENKO, Iryna; NASRALLAH, Nohade; GARCIA, Henri. Detecting Spoofing in High-Frequency Trading Using Machine Learning Techniques. *Institut Louis Bachelier Working Paper*. 2024. Dostupné z URL: <https://www.institutlouisbachelier.org/wp-content/uploads/2024/03/detecting-spoofing-in-high-frequency-trading-using-machine-learning-techniques.pdf>.
3. TUCCELLA, Jean-Noël; NADLER, Philip; SERBAN, Ovidiu. *Protecting Retail Investors from Order Book Spoofing using a GRU-based Detection Model*. 2021. Dostupné z DOI: 10.48550/arXiv.2110.03687.
4. LI, Maoxi; SHU, Mengying; LU, Tianyu. *Anomaly Pattern Detection in High-Frequency Trading Using Graph Neural Networks*. 2024. Dostupné z DOI: 10.70393/6a69656173.323430.
5. COLLABORATION, HighLO. *HighLO: High Energy Physics Tools in Limit Order Book Analysis*. 2024. Dostupné z URL: <https://www.highlo.org>. [Citováno: 3. 2. 2025].
6. DEBIE, Philippe et al. Unravelling the JPMorgan spoofing case using particle physics visualization methods. *European Financial Management*. 2023, roč. 29, č. 1, s. 288–326. Dostupné z DOI: 10.1111/eufm.12353.
7. ABERGEL, F.; ANANE, M.; CHAKRABORTI, A.; JEDIDI, A.; TOKE, I.M. *Limit Order Books*. Cambridge University Press, 2016. PHYSICS OF SOCIETY: ECONOPHYSI. ISBN 9781107163980. Dostupné z DOI: 10.1017/CBO9781316683040.
8. GOULD, Martin D. et al. Limit order books. *Quantitative Finance*. 2013, roč. 13, č. 11, s. 1709–1742. Dostupné z DOI: 10.1080/14697688.2013.803148.

9. SCHWARTZ, R.A.; SIPRESS, G.M.; WEBER, B.W. *Mastering the Art of Equity Trading Through Simulation, + Web-Based Software: The TraderEx Course*. Wiley, 2010. Wiley Trading. ISBN 9780470464854. Dostupné z DOI: 10.1002/9781119198253.
10. GROUP, Deutsche Börse. *Enhanced Order Book Interface Manual, Version 8.1.1*. Deutsche Börse AG, 2020. Ver. 8.1.1. Dostupné z URL: [https://www.eurexchange.com/resource/blob/2128190/1c3ff499decf4bc0516e5a0e6b2c1af9/data/T7\\_EOBI\\_Manual\\_v\\_8\\_1\\_1.pdf](https://www.eurexchange.com/resource/blob/2128190/1c3ff499decf4bc0516e5a0e6b2c1af9/data/T7_EOBI_Manual_v_8_1_1.pdf). Build Version 81.6.10.ga-81006010-57 [Citováno: 7. 4. 2025].
11. STENFORS, Alexis; DORAGHI, Mehrdaad; SOVIANY, Cristina; SUSAI, Masa-yuki; VAKILI, Kaveh. Cross-market spoofing. *Journal of International Financial Markets, Institutions and Money*. 2023, roč. 83, s. 101735. ISSN 1042-4431. Dostupné z DOI: 10.1016/j.intfin.2023.101735.
12. STENFORS, Alexis; DILSHANI, Kaveesha; GUO, Andy; MERE, Peter. Detecting the Risk of Cross-Product Manipulation in the EUREX Fixed Income Futures Market. *SSRN Electronic Journal*. 2024. Dostupné z DOI: 10.2139/ssrn.4546523.
13. CARTEA, Álvaro; JAIMUNGAL, Sebastian; WANG, Yixuan. Spoofing and Price Manipulation in Order Driven Markets. *SSRN Electronic Journal*. 2019. Dostupné z DOI: 10.2139/ssrn.3431139.
14. GROUP, Deutsche Börse. *Deutsche Börse*. [B.r.]. Dostupné z URL: <https://www.deutsche-boerse.com>. [Citováno: 22. 1. 2025].
15. GROUP, Deutsche Börse. *Deutsche Börse A7*. [B.r.]. Dostupné z URL: <https://a7.deutsche-boerse.com>. [Citováno: 22. 1. 2025].
16. JSON.ORG. *JSON: JavaScript Object Notation*. 1999. Dostupné z URL: <https://www.json.org/json-en.html>. [Citováno: 22. 1. 2025].
17. CONGRESS, Library of. *CSV, Comma Separated Values (strict form as described in RFC 4180)*. 2024. Dostupné z URL: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000323.shtml>. [Citováno: 22. 1. 2025].
18. DATA, Lobster. *Data Structure for LOBSTER*. 2018. Dostupné z URL: <https://lobsterdata.com/info/DataStructure.php>. [Citováno: 22. 1. 2025].
19. SOUKUP, Martin. *Experimentální server na podporu výzkumu v oblasti EOBI dat*. 2024. Dostupné z URL: <http://hdl.handle.net/11025/57109>. Západočeská univerzita v Plzni. Vedoucí práce: Ing. Kamil Ekštein, Ph.D.

20. DUMAS, Maxime; MCGUFFIN, Michael; LEMIEUX, Victoria. Financevis.net - A Visual Survey of Financial Data Visualizations. In: *Poster Abstracts of IEEE Conference on Visualization*. 2014. Dostupné z URL: <http://financevis.net/>. [Citováno: 23. 1. 2025].
21. DUMAS, Maxime; MCGUFFIN, Michael; LEMIEUX, Victoria. *Leveraging Order Book Heatmaps and Trade Order Flow for Market Trend Analysis*. 2025. Dostupné z URL: <https://blog.amberdata.io/leveraging-order-book-heatmaps-and-trade-order-flow-for-market-trend-analysis>. [Citováno: 23. 1. 2025].
22. BOOKMAP. *Bookmap - A Futures and Stocks Trading Platform*. 2025. Dostupné z URL: <https://bookmap.com/en>. [Citováno: 22. 1. 2025].
23. QUANTCONNECT. *QuantConnect - The World's Leading Algorithmic Trading Platform*. 2025. Dostupné z URL: <https://www.quantconnect.com/>. [Citováno: 22. 1. 2025].
24. NINJATRADER. *NinjaTrader - Better Futures Trading Starts Now*. 2025. Dostupné z URL: <https://nintrader.com/>. [Citováno: 22. 1. 2025].
25. TRADINGVIEW. *Lightweight Charts*. 2025. Dostupné z URL: <https://www.tradingview.com/lightweight-charts/>. [Citováno: 22. 1. 2025].
26. SNEIDERMAN, Robby. A Quick Introduction to Time Series Analysis. 2021. Dostupné z URL: <https://towardsdatascience.com/a-quick-introduction-to-time-series-analysis-d86e4ff5fdd>.
27. HAMILTON, J.D. *Time Series Analysis*. Princeton University Press, 2020. ISBN 9780691218632. Dostupné z DOI: 10.2307/j.ctv14jx6sm.
28. BLÁZQUEZ-GARCÍA, Ane; CONDE, Angel; MORI, Usue; LOZANO, Jose A. A Review on Outlier/Anomaly Detection in Time Series Data. *ACM Comput. Surv.* 2021, roč. 54, č. 3. ISSN 0360-0300. Dostupné z DOI: 10.1145/3444690.
29. MONTGOMERY, John D. Spoofing, Market Manipulation, and the Limit-Order Book. *SSRN Electronic Journal*. 2016. Dostupné z DOI: 10.2139/ssrn.2780579.
30. COMMISSION, Commodity Futures Trading. *CFTC Orders JPMorgan to Pay Record \$920 Million for Spoofing and Manipulation*. 2020. Dostupné z URL: <https://www.cftc.gov/PressRoom/PressReleases/8260-20>. Press Release No. 8260-20 [Citováno: 21. 1. 2025].

31. JUSTICE, U.S. Department of. *Deutsche Bank Agrees to Pay Over \$130 Million to Resolve Foreign Corrupt Practices Act and Fraud*. 2025. Dostupné z URL: <https://www.justice.gov/opa/pr/deutsche-bank-agrees-pay-over-130-million-resolve-foreign-corrupt-practices-act-and-fraud>. [Citováno: 21. 1. 2025].
32. COMMISSION, Commodity Futures Trading. *CFTC Orders The Bank of Nova Scotia to Pay \$127.4 Million for Spoofing, False Statements, Compliance and Supervision Violations*. 2020. Dostupné z URL: <https://www.cftc.gov/PressRoom/PressReleases/8220-20>. Press Release No. 8220-20 [Citováno: 21. 1. 2025].
33. JUSTICE, U.S. Department of. *Tower Research Capital LLC Agrees to Pay \$67 Million in Connection with Commodities Fraud Scheme*. 2025. Dostupné z URL: <https://www.justice.gov/opa/pr/tower-research-capital-llc-agrees-pay-67-million-connection-commodities-fraud-scheme>. [Citováno: 21. 1. 2025].
34. ILLINOIS, U.S. Attorney's Office for the Northern District of. *High-Frequency Trader Sentenced to Three Years in Prison for Disrupting Futures Market*. 2025. Dostupné z URL: <https://www.justice.gov/usao-ndil/pr/high-frequency-trader-sentenced-three-years-prison-disrupting-futures-market-first>. [Citováno: 21. 1. 2025].
35. JUSTICE, U.S. Department of. *United States v. Navinder Singh Sarao*. 2023. Dostupné z URL: <https://www.justice.gov/criminal/criminal-vns/united-states-v-navinder-singh-sarao>. Court Docket No.: 1:15-cr-00075 (N.D. Illinois) [Citováno: 21. 1. 2025].
36. JUNG, A. *Machine Learning: The Basics*. Springer Nature Singapore, 2022. Machine Learning: Foundations, Methodologies, and Applications. ISBN 9789811681936. Dostupné z URL: <https://books.google.cz/books?id=1IBaEAAAQBAJ>.
37. MURPHY, K.P. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. Adaptive Computation and Machine Learning series. ISBN 9780262018029. Dostupné z URL: <https://books.google.cz/books?id=NZP6AQAAQBAJ>.
38. LIU, Fei Tony; TING, Kai Ming; ZHOU, Zhi-Hua. Isolation Forest. In: *2008 Eighth IEEE International Conference on Data Mining*. 2008, s. 413–422. Dostupné z DOI: 10.1109/ICDM.2008.17.
39. SCHÖLKOPF, Bernhard; PLATT, John C.; SHawe-Taylor, John; SMOLA, Alex J.; WILLIAMSON, Robert C. Estimating the Support of a High-Dimensional Distribution. *Neural Computation*. 2001, roč. 13, č. 7, s. 1443–1471. ISSN 0899-7667. Dostupné z DOI: 10.1162/089976601750264965.

40. BREUNIG, Markus M.; KRIEGEL, Hans-Peter; NG, Raymond T.; SANDER, Jörg. LOF: identifying density-based local outliers. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. Dallas, Texas, USA: Association for Computing Machinery, 2000, 93–104. SIGMOD '00. ISBN 1581132174. Dostupné z DOI: 10.1145/342009.335388.
41. HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the Dimensionality of Data with Neural Networks. *Science*. 2006, roč. 313, č. 5786, s. 504–507. Dostupné z DOI: 10.1126/science.1127647.
42. NIELSEN, Michael A. *Neural networks and deep learning*. Sv. 25. Determination press San Francisco, CA, USA, 2015.
43. VASWANI, Ashish et al. Attention is All you Need. In: GUYON, I. et al. (ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017, sv. 30. Dostupné z DOI: 10.48550/arXiv.1706.03762.
44. RUBISOV, Anton D. *Statistical Arbitrage Using Limit Order Book Imbalance*. 2015. Dostupné z URL: <https://api.semanticscholar.org/CorpusID:55130442>.
45. DO, Bao; PUTNINS, Talis. Detecting Layering and Spoofing in Markets. *SSRN Electronic Journal*. 2023. Dostupné z DOI: 10.2139/ssrn.4525036.
46. KŮSOVÁ, Martina. *Modelling and prediction of data in limit order books*. 2023. Dostupné z URL: <http://hdl.handle.net/11025/53795>. Západočeská univerzita v Plzni. Vedoucí práce: doc. Ing. Jan Pospíšil, Ph.D.
47. GOIX, Nicolas. How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms? *arXiv preprint arXiv:1607.01152*. 2016. Dostupné z DOI: 10.48550/arXiv.1607.01152.
48. POLONIK, Wolfgang. Measuring Mass Concentrations and Estimating Density Contour Clusters—An Excess Mass Approach. *The Annals of Statistics*. 1995, roč. 23, č. 3, s. 855–881. Dostupné z URL: <http://www.jstor.org/stable/2242426>.
49. POLONIK, Wolfgang. Minimum volume sets and generalized quantile processes. *Stochastic Processes and their Applications*. 1997, roč. 69, č. 1, s. 1–24. ISSN 0304-4149. Dostupné z DOI: 10.1016/S0304-4149(97)00028-8.



# Seznam obrázků

2.1	Příklad možné vizualizace knihy limitních objednávek pro konkrétní časový okamžik . . . . .	7
2.2	Příklad možné vizualizace knihy limitních objednávek jako <i>časové řady</i> v kombinaci s <i>teplotní mapou</i> . . . . .	11
2.3	Základní model autoenkodéru . . . . .	22
3.1	Základní snímek obrazovky z vizualizačního nástroje . . . . .	41
3.2	Snímek obrazovky z vizualizačního nástroje podchycující interaktivitu	42
3.3	Cenový graf s pěti cenovými úrovněmi, imbalance indexem a frekvencemi zpráv (všech a pouze těch o rušení objednávek) (Security ID 4128839) . . . . .	43
3.4	Přefiltrovaný cenový graf s jednou ( <i>nejlepší</i> ) cenovou úrovni ( <i>Top of Book</i> ) a frekvencemi příchozích zpráv (Security ID 4128839) . . . . .	43
4.1	Přefiltrovaný cenový graf s jednou ( <i>nejlepší</i> ) cenovou úrovni ( <i>Top of Book</i> ) a frekvencemi příchozích zpráv – červenou elipsou je označena <i>podezřelá oblast</i> (Security ID 4128839) . . . . .	48
4.2	Přiblížení na <i>podezřelou oblast</i> z Obrázku 4.1 . . . . .	48
4.3	Korelační matice (Security ID 4128839) . . . . .	50
4.4	Boxplot důležitosti příznaků podle algoritmu <i>DIFFI</i> – spuštěno stokrát pro zajištění stabilních výsledků (Security ID 4128839) . . . . .	51
4.5	Koefficienty hlavních komponent jako vektory v prostoru první a druhé komponenty po aplikaci PCA na celá data (Security ID 4128839) . . . . .	51
4.6	Paretův diagram zobrazující procento zachované informace (vysvětleného rozptylu) při aplikaci PCA na celá data (Security ID 4128839) . . . . .	52
4.7	Korelační matice po redukci dimenzí (Security ID 4128839) . . . . .	53
4.8	Boxplot důležitosti příznaků podle algoritmu <i>DIFFI</i> po redukci dimenzí – spuštěno stokrát pro zajištění stabilních výsledků (Security ID 4128839) . . . . .	54

4.9	Koeficienty hlavních komponent jako vektory v prostoru první a druhé komponenty po aplikaci PCA na celá data po redukci dimenzí ( <b>Security ID 4128839</b> ) . . . . .	54
4.10	Paretův diagram zobrazující procento zachované informace (vysvětleného rozptylu) při aplikaci PCA na celá data po redukci dimenzí ( <b>Security ID 4128839</b> ) . . . . .	55
4.11	Srovnání všech referenčních modelů na metrice <i>EM</i> . . . . .	56
4.12	Srovnání všech referenčních modelů na metrice <i>MV</i> . . . . .	57
4.13	Filtrované zobrazení všech běhů modelu <i>IF</i> na metrice <i>EM</i> . . . . .	57
4.14	Filtrované zobrazení všech běhů modelu <i>IF</i> na metrice <i>MV</i> . . . . .	58
4.15	Srovnání všech neuronových modelů na hodnotě ztrátové funkce na validačních datech (filtrování hodnot – maximálně 1.0) . . . . .	59
4.16	Filtrované mapování nastavení hyperparametrů všech běhů modelu <i>CNN</i> na <i>ztrátovou funkci</i> . . . . .	60
4.17	Filtrované mapování nastavení hyperparametrů všech běhů modelu <i>CNN</i> na <i>ztrátovou funkci</i> po redukci dimenzionality vstupních dat . . . . .	61
4.18	Porovnání nejlepších referenčních modelů pomocí křivek <i>EM</i> a <i>MV</i> ( <b>Security ID 4578882</b> ) . . . . .	64
4.19	Porovnání nejlepších konfigurací jednotlivých architektur <i>autoencoderů</i> z hlediska validační ztrátové funkce (agregace napříč iteracemi <i>5-fold křížové validace</i> ) . . . . .	65
4.20	Porovnání nejlepších neuronových modelů pomocí křivek <i>EM</i> a <i>MV</i> ( <b>Security ID 4578882</b> ) . . . . .	65
4.21	Srovnání všech modelů na křivkách <i>EM</i> a <i>MV</i> (semi-logaritmické měřítko) ( <b>Security ID 4578882</b> ) . . . . .	66
4.22	Porovnání modelů <i>IF</i> , <i>FFNN</i> , <i>CNN</i> a <i>Transformer</i> pomocí křivek <i>EM</i> a <i>MV</i> (semi-logaritmické měřítko) ( <b>Security ID 4578882</b> ) . . . . .	66
4.23	Detekované anomálie modelem <i>IF</i> ( <b>Security ID 5578483</b> ) . . . . .	68
4.24	Detekované anomálie modelem <i>FFNN</i> ( <b>Security ID 5578483</b> ) . . . . .	68
4.25	Detekované anomálie modelem <i>CNN</i> ( <b>Security ID 5578483</b> ) . . . . .	69
4.26	Detekované anomálie modelem <i>Transformer</i> ( <b>Security ID 5578483</b> ) . . . . .	69
4.27	Detekované anomálie modelem <i>IF</i> na redukovaných datech ( <b>Security ID 5578483</b> ) . . . . .	70
4.28	Detekované anomálie modelem <i>FFNN</i> na redukovaných datech ( <b>Security ID 5578483</b> ) . . . . .	70
4.29	Detekované anomálie modelem <i>CNN</i> na redukovaných datech ( <b>Security ID 5578483</b> ) . . . . .	71
4.30	Detekované anomálie modelem <i>Transformer</i> na redukovaných datech ( <b>Security ID 5578483</b> ) . . . . .	71
4.31	Detekované anomálie <i>souborem modelů</i> ( <b>Security ID 5578483</b> ) . . . . .	72

---

4.32	Detekované anomálie <i>souborem modelů</i> na redukovaných datech (Security ID <b>5578483</b> ) . . . . .	72
4.33	Přefiltrované detekce anomálií <i>souborem modelů</i> (Security ID <b>5578483</b> )	73
4.34	Přefiltrované detekce anomálií <i>souborem modelů</i> na redukovaných da- tech (Security ID <b>5578483</b> ) . . . . .	73
B.1	Příklad průběhu <i>nejlepší poptávkové ceny</i> v čase (Security ID <b>4128839</b> )	83
B.2	Příklad průběhu <i>objemu nejlepších poptávek</i> v čase (Security ID <b>4128839</b> )	84
B.3	Příklad průběhu <i>nejlepší nabídkové ceny</i> v čase (Security ID <b>4128839</b> )	84
B.4	Příklad průběhu <i>objemu nejlepších nabídek</i> v čase (Security ID <b>4128839</b> )	85
B.5	Příklad průběhu <i>imbalance indexu</i> v čase (Security ID <b>4128839</b> ) . . .	85
B.6	Příklad průběhu <i>frekvence příchozích zpráv</i> v čase (Security ID <b>4128839</b> )	86
B.7	Příklad průběhu <i>frekvence rušení objednávek</i> v čase (Security ID <b>4128839</b> ) . . . . .	86
B.8	Příklad průběhu dimenze „ <i>High Quoting Activity</i> “ v čase (Security ID <b>4128839</b> ) . . . . .	87
B.9	Příklad průběhu dimenze „ <i>Unbalanced Quoting</i> “ v čase (Security ID <b>4128839</b> ) . . . . .	87
B.10	Příklad průběhu dimenze „ <i>Low Execution Probability</i> “ v čase (Security ID <b>4128839</b> ) . . . . .	88
B.11	Příklad průběhu dimenze „ <i>Trades Oppose Quotes</i> “ v čase (Security ID <b>4128839</b> ) . . . . .	88
B.12	Příklad průběhu dimenze „ <i>Cancels Oppose Trades</i> “ v čase (Security ID <b>4128839</b> ) . . . . .	89
B.13	Příklad průběhu <i>střední ceny</i> v čase (Security ID <b>4128839</b> ) . . . . .	89
B.14	Příklad průběhu dimenze „ <i>Imbalance Volumes PC1</i> “ v čase (Security ID <b>4128839</b> ) . . . . .	90
B.15	Příklad průběhu <i>frekvence rušení objednávek</i> v čase (Security ID <b>4128839</b> ) . . . . .	90
B.16	Příklad průběhu dimenze „ <i>Quoting PC1</i> “ v čase (Security ID <b>4128839</b> )	91
B.17	Příklad průběhu dimenze „ <i>Oppose PC1</i> “ v čase (Security ID <b>4128839</b> )	91



# Seznam tabulek

2.1	Příklad <i>LOBSTER</i> formátu (do druhé úrovně cen a objemů) . . . . .	10
4.1	Konkrétní výběr dnů a produktů . . . . .	45
4.2	Přehled analyzovaných datových souborů . . . . .	46
4.3	Výčet laděných hyperparametrů a jejich testovaných hodnot (referenční modely) . . . . .	56
4.4	Výčet laděných hyperparametrů a jejich testovaných hodnot (neuronové modely) . . . . .	59
4.5	Výčet laděných hyperparametrů a jejich testovaných hodnot po redukci dimenze vstupních dat (neuronové modely) . . . . .	62



# Seznam výpisů

A.1	Ukázkový výpis při klonování repozitáře . . . . .	77
A.2	Ukázkový výpis tvorby a aktivace virtuálního prostředí ( <b>Windows</b> ) . . . . .	78
A.3	Ukázkový výpis tvorby a aktivace virtuálního prostředí ( <b>Linux</b> ) . . . . .	78
A.4	Ukázkový výpis instalace závislostí . . . . .	78
A.5	Ukázkový výpis spuštění stažení dat . . . . .	79
A.6	Ukázkové spuštění modelu <i>CNN Autoencoder</i> . . . . .	81
A.7	Ukázkové spuštění modelu <i>Isolation Forest</i> . . . . .	81
A.8	Ukázkové spuštění souboru modelů . . . . .	82

