



STŘEDNÍ ŠKOLA PRŮMYSLOVÁ
A UMĚLECKÁ, OPAVA

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Automatické obchodování na Forex trzích pomocí umělé inteligence

Tomáš Schablický

Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování

Třída: IT4

Školní rok: 2024/2025

Poděkování

Rád bych vyjádřil své poděkování panu Mgr. Markovi Lučnému za pomoc a zpětnou vazbu, kterou mi poskytoval během tvorby této závěrečné práce. Také bych rád poděkoval Matěji Zimmerovi za jeho práci na tomto projektu a ochotu mi pomáhat.

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2024

podpis autora práce

ABSTRAKT

Projekt představuje webovou aplikaci pro obchodování na forexových trzích s využitím metod posilovaného učení. Aplikace umožňuje manuální obchodování prostřednictvím REST API, které propojuje uživatele s platformou MetaAPI. Uživatelé mohou odesílat příkazy k otevření a uzavření pozic a sledovat aktivní obchody.

Automatizované obchodování je implementováno pomocí modelu posilovaného učení, který poskytuje doporučení k obchodním akcím (BUY/SELL) na základě analýzy tržních dat. Model je vytvořen pomocí TensorFlow a kombinuje neuronové sítě CNN a LSTM s mechanismem pozornosti. Prostředí pro trénink zahrnuje technické indikátory, jako jsou klouzavé průměry, volatilita a objemové indikátory.

Data z trhu jsou sbírána prostřednictvím integrace s MetaAPI pomocí Celery a Redis serveru v intervalech jedné minuty a ukládána do databáze. Aplikace také obsahuje vizualizace tržních dat a výkonu modelu prostřednictvím jednoduchých grafů.

ABSTRACT

The project introduces a web application for forex trading utilizing reinforcement learning methods. The application enables manual trading via REST API, connecting users with the MetaAPI platform. Users can send commands to open and close positions and monitor active trades.

Automated trading is implemented through a reinforcement learning model that provides trading action recommendations (BUY/SELL) based on market data analysis. The model is developed using TensorFlow and combines CNN and LSTM neural networks with an attention mechanism. The training environment includes technical indicators such as moving averages, volatility, and volume indicators.

Market data is collected through MetaAPI integration using Celery and Redis servers at one-minute intervals and stored in a database. The application also features visualizations of market data and model performance using simple graphs.

Obsah:

Úvod.....	4
1 Teoretická a metodická východiska.....	6
1.1 Přehled forexového trhu.....	6
1.2 Metody posilovaného učení.....	6
1.3 Technické indikátory.....	7
1.4 Výběr metod a nástrojů.....	10
2 Využité technologie.....	11
2.1 TensorFlow.....	11
2.2 Django.....	12
2.3 Celery a Redis.....	12
2.4 Tailwind CSS.....	13
2.5 MetaAPI.....	13
3 Způsoby řešení a použité postupy.....	14
3.1 Implementace obchodního systému.....	14
3.2 Parametry systému.....	16
3.3 Schémata a obrázky.....	16
3.4 Použité příkazy.....	18
4 Výsledky řešení, výstupy, uživatelský manuál.....	20
4.1 Výčet splněných cílů.....	20
4.2 Nesplněné cíle.....	20
4.3 Výstupy a prokázání funkčnosti.....	21
4.4 Uživatelský manuál.....	21
4.5 Obrázky a schémata.....	22
Závěr.....	24

Úvod

Tato práce se zaměřuje na vývoj a implementaci systému pro obchodování na forexovém trhu, který kombinuje manuální a automatizované obchodní strategie s využitím umělé inteligence (AI). Projekt byl realizován ve spolupráci dvou studentů: já a Matěj Zimmer jsme společně pracovali na všech aspektech projektu, přičemž každý z nás se zaměřil na různé části vývoje.

Já jsem se soustředil především na backendovou část systému. Zajišťoval jsem získávání a zpracování dat z forexového trhu, implementaci algoritmů pro predikce cen a vývoj AI modelu, který má schopnost se učit z historických dat a následně generovat obchodní doporučení. Mým cílem bylo optimalizovat přesnost predikcí a zajistit efektivní komunikaci mezi databází a uživatelským rozhraním.

Matěj Zimmer se zaměřil hlavně na frontendovou část projektu. Pracoval s Django modely, navrhoval a implementoval vzhled webové aplikace a zajišťoval uživatelsky přívětivé rozhraní pro zobrazení vývoje cen a výsledků predikcí. Díky jeho práci je aplikace vizuálně přehledná a snadno ovladatelná.

Téma jsme si vybrali, protože jsme se v minulosti snažili sami obchodovat na forexovém trhu, avšak bez výraznějšího úspěchu. Uvědomili jsme si, že obchodování na tomto trhu je velmi náročné a vyžaduje rychlá a přesná rozhodnutí, což může být pro jednotlivce komplikované. Z toho důvodu jsme se rozhodli přenechat rozhodovací proces umělé inteligenci a vytvořit systém, který se dokáže učit z historických dat a automaticky navrhopvat obchodní příkazy.

Výsledkem našeho projektu je komplexní aplikace, která kombinuje historická data, pokročilé algoritmy a uživatelské rozhraní pro zobrazení vývoje trhu. AI model analyzuje historická data a na jejich základě navrhuje optimální obchodní strategie. Tento systém má potenciál snížit lidskou chybu a zvýšit efektivitu obchodování na forexovém trhu.

Hlavní cíle naší práce se soustředí na vývoj webové aplikace zaměřené na obchodování na forexovém trhu, která zahrnuje různé funkcionality pro uživatele. Mezi klíčové součásti aplikace patří:

- Implementace systému pro manuální obchodování na forexu.
- Vývoj AI modelu, který nejen poskytuje doporučení obchodních strategií, ale také automaticky provádí obchodní příkazy.
- Vizualizace dat a statistik pro analýzu obchodních rozhodnutí.
- Chat mezi uživateli pro sdílení zkušeností a strategií.
- Leaderboard, který zobrazuje nejúspěšnější obchodníky.
- Sekce s novinkami a aktuálními informacemi z forexového trhu.

V následujících kapitolách bude podrobně rozebráno fungování webové aplikace, včetně systémů pro manuální a automatizované obchodování, použití technických indikátorů a AI modelu pro predikce cenových pohybů. Bude také představen návrh vizualizací pro lepší přehled o vývoji obchodních příkazů, funkčnost chatu a leaderboardu. Dále budou popsány použité technologie, proces sběru a zpracování dat a plány pro rozšíření systému v budoucnosti.

1 TEORETICKÁ A METODICKÁ VÝCHODISKA

Tato kapitola se zaměřuje na teoretické a metodické základy používané v rámci tohoto projektu zaměřeného na obchodování na forexovém trhu. Cílem je objasnit klíčové pojmy, vymezit základní teoretické rámce a představit metodiky a technologie, které jsou aplikovány v navrhovaném systému.

1.1 Přehled forexového trhu

Forexový trh (Foreign Exchange) je decentralizovaný trh pro výměnu měn, který funguje 24 hodin denně, 5 dní v týdnu. Tento trh je klíčovým prvkem globální ekonomiky, kde měnové páry jako EUR/USD, GBP/JPY a další jsou obchodovány mezi různými účastníky, včetně bank, vládních institucí, korporací a individuálních obchodníků. Ceny měn se na tomto trhu neustále mění v závislosti na ekonomických, politických a sociálních faktorech, což představuje příležitosti pro spekulanty, kteří hledají zisk z těchto změn. Obchodování na forexu je vysoce rizikové a vyžaduje nejen technické a fundamentální analýzy, ale i správné řízení rizik.

1.2 Metody posilovaného učení

Pro predikci a optimalizaci obchodních rozhodnutí v tomto projektu je využívána metoda posilovaného učení (Reinforcement Learning, RL). Posilované učení je typ strojového učení, kde model (agent) se učí pomocí zpětné vazby z prostředí, v tomto případě z obchodních dat. Agent provádí akce (např. nákup, prodej, držení pozice) na základě stavu trhu a dostává odměny nebo tresty podle toho, jak úspěšně jeho akce vedou k dosažení cíle (např. zisku). Jednou z nejdůležitějších vlastností našeho modelu je nepřetržité učení. Po každém uzavřeném obchodu nastává trénování což v praxi zvyšuje přesnost a profitabilitu modelu.

V našem projektu je aplikována metoda Q-learningu, AI agent si postupně **buduje tabulku Q-hodnot (Q-Table)**, která říká, jakou hodnotu má každá možná akce v daném stavu prostředí. Cílem je **maximalizovat celkovou odměnu (reward)** za své akce.

- **Stav (State, S):** Reprezentuje aktuální situaci v prostředí (např. aktuální cena EUR/USD na forexu).
- **Akce (Action, A):** Možné kroky, které agent může provést (např. koupit, prodat nebo neudělat nic).

- **Odměna (Reward, R):** Hodnota, kterou agent obdrží po provedení akce (např. zisk nebo ztráta z obchodu).

Agent se učí pomocí aktualizace Q-hodnot podle následující rovnice:

$$Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_{a'} Q(S', a') - Q(S, A)]$$

Kde:

- $Q(S, A)$ = Q-hodnota pro konkrétní stav a akci
- α = **rychlost učení (learning rate)**, tj. jak rychle AI přizpůsobuje své znalosti
- R = **okamžitá odměna**
- γ = **diskontní faktor, který určuje, jak moc AI bere v úvahu budoucí odměny**
- $\max_{a'} Q(S', a')$ = **maximální očekávaná budoucí odměna ze všech možných akcí v novém stavu**

1.3 Technické indikátory

Technické indikátory hrají klíčovou roli při analýze tržních dat a rozhodování o obchodních příkazech. V rámci tohoto projektu jsou používány různé technické indikátory, jako jsou Simple Moving Average (SMA), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD) a další. Tyto indikátory pomáhají modelu identifikovat trendy, momenty překoupenosti nebo přeprodanosti a signály pro potenciální obchody.

- **SMA – Simple Moving Average (Jednoduchý klouzavý průměr):** je jedním z nejjednodušších a nejpoužívanějších technických indikátorů v obchodování na finančních trzích.

SMA se počítá jako průměrná hodnota uzavíracích cen za určité časové období. Výsledkem je čára, která se promítá do cenového grafu a sleduje pohyb trhu.

Vzorec pro výpočet SMA:

$$SMA = \frac{C_1 + C_2 + \dots + C_n}{n}$$

Kde:

C_1, C_2, \dots, C_n jsou uzavírací ceny za jednotlivé období (např.: minuta, hodina

n je počet období, ze kterých se SMA počítá.

- RSI – Relative Strength Index (Index relativní síly): je oscilátorový technický indikátor, který měří rychlost a změnu cenových pohybů. Patří mezi nejpopulárnější nástroje používané obchodníky k analýze finančních trhů, včetně forexu. RSI pomáhá určit, zda je trh **překoupený** nebo **přeprodaný**, což naznačuje možné budoucí obrácení trendu.

RSI se pohybuje v rozmezí **0 až 100**.

Hodnota **nad 70** obvykle značí, že je trh **překoupený** a může dojít k poklesu.

Hodnota **pod 30** značí, že je trh **přeprodaný** a může dojít k růstu.

Vzorec pro výpočet RSI:

$$RSI = 100 - \left(\frac{100}{1 + RS} \right)$$

Kde:

$$RS = \frac{\text{průměrný zisk}}{\text{průměrná ztráta}}$$

Průměrný zisk a průměrná ztráta se obvykle počítají za **14 období** (např. 14 hodin nebo minut), což je standardní nastavení.

Jak interpretovat RSI?

RSI hodnota	Interpretace	Akce
0-30	Přeprodaný trh	Možný růst
30-70	Neutrální zóna	Žádný signál
70-100	Překoupený trh	Možný pokles

V projektu forexového obchodování je RSI používán jako jeden z klíčových technických indikátorů. Pomáhá AI modelu identifikovat potenciální obraty trhu, což zlepšuje přesnost doporučení pro nákup nebo prodej.

- **MACD – Moving Average Convergence Divergence:** je populární technický indikátor používaný k identifikaci směru a síly trendu a také ke generování obchodních signálů. Tento indikátor měří rozdíl mezi dvěma exponenciálními klouzavými průměry (EMA) a poskytuje obchodníkům přehled o tom, kdy je trh v rostoucím nebo klesajícím trendu.

MACD se skládá ze tří hlavních komponent:

- **MACD Line (hlavní linie)** – rozdíl mezi dvěma EMA (obvykle EMA 12 a EMA 26).
- **Signal Line (signální linie)** – EMA hlavní linie (obvykle EMA 9).
- **Histogram** – grafický rozdíl mezi MACD linií a signální linií.

Vzorce:

$$MACD\ Line = EMA_{12} - EMA_{26}$$

$$Signal\ Line = EMA_9\ hlavní\ linie$$

$$Histogram = MACD\ Line - Signal\ Line$$

Jak interpretovat MACD?

Signál	Interpretace
MACD Line překříží Signal Line směrem nahoru	Býčí signál – doporučení k nákupu.
MACD Line překříží Signal Line směrem dolů	Medvědí signál – doporučení k prodeji.
Histogram roste	Trend nabírá na síle.
Histogram klesá	Trend slábne.

1.4 Výběr metod a nástrojů

Pro tento projekt byly zvoleny metody, které umožňují efektivní analýzu a predikci tržního chování na forexovém trhu. Hlavním přístupem je kombinace posilovaného učení a technických indikátorů, které poskytují robustní rámec pro analýzu dat a obchodní rozhodování. Důvodem pro výběr posilovaného učení je jeho schopnost neustále se adaptovat na měnící se tržní podmínky a optimalizovat obchodní strategie na základě historických dat a zpětné vazby.

Pro implementaci modelu byly vybrány následující technologie: TensorFlow pro trénink modelu, Django pro vytvoření REST API a interakci s obchodní platformou a Celery pro asynchronní sběr dat. Tyto nástroje byly zvoleny na základě jejich robustnosti, flexibility a dostupnosti v rámci komunitního vývoje.

2 VYUŽITÉ TECHNOLOGIE

V této kapitole jsou popsány technologie, které byly použity pro vývoj a implementaci systému pro obchodování na forexovém trhu. Tyto technologie byly vybrány na základě jejich schopnosti efektivně podporovat potřeby projektu, a to jak z hlediska výkonu, tak i flexibility a škálovatelnosti.

2.1 TensorFlow

TensorFlow je open-source framework pro strojové učení a umělou inteligenci, který byl vybrán pro trénování a implementaci AI modelu. TensorFlow je vysoce optimalizován pro práci s velkými objemy dat a umožňuje efektivní trénování modelů na CPU a GPU. Pro tento projekt byl TensorFlow zvolen zejména kvůli jeho schopnosti podporovat různé typy modelů, včetně hlubokých neuronových sítí, což je nezbytné pro implementaci složitých modelů strojového učení, jako jsou modely typu CNN, LSTM a Attention, které jsou součástí tohoto projektu.

Klíčové parametry:

- **Podpora GPU a distribuovaného trénování:** TensorFlow umožňuje rychlé trénování modelů na grafických kartách, což je nezbytné pro zpracování velkých objemů historických dat a trénování složitých modelů.
- **Flexibilita:** Tento framework podporuje jak nízkoúrovňové API pro detailní kontrolu, tak i vysokoúrovňové API (např. Keras) pro rychlý vývoj modelů.
- **Široká podpora komunitních knihoven:** TensorFlow má silnou komunitní podporu, což usnadňuje implementaci pokročilých funkcí a zajišťuje stálý vývoj nových nástrojů.
- **LSTM (Long Short-Term Memory):** je druh rekurentní neuronové sítě (RNN), která je navržena tak, aby si pamatovala dlouhodobé závislosti v datech. Oproti běžným RNN řeší problém mizení gradientu, což umožňuje pracovat s dlouhými časovými řadami.
- **CNN (Convolutional Neural Network):** je typ neuronové sítě, která je nejčastěji využívána pro analýzu obrazů. Dokáže rozpoznávat vzory a struktury v datech pomocí konvolučních vrstev, které hledají filtry nebo charakteristické znaky.
- **Attention Mechanismus:** je technika používaná v moderních neuronových sítích, která umožňuje soustředit se na klíčové části dat při zpracování dlouhých sekvencí. Tento mechanismus je zásadní v pokročilých modelech, jako jsou Transformer (např. GPT), protože řeší problém, kdy důležité informace mohou být daleko od sebe v sekvenci.

2.2 Django

Django je webový framework napsaný v Pythonu, který byl použit pro vývoj REST API pro komunikaci mezi uživatelským rozhraním a obchodní platformou. Django je vysoce efektivní pro rychlý vývoj webových aplikací a poskytuje robustní nástroje pro autentizaci, správu databází a bezpečnostní opatření. V tomto projektu je Django klíčovým komponentem pro zajištění komunikace mezi uživatelem, backendem a obchodním prostředím.

Klíčové parametry:

- **Rychlý vývoj:** Django je navrženo tak, aby usnadnilo rychlý vývoj aplikací, což je ideální pro prototypování a implementaci komplexních systémů v krátkém čase.
- **Vestavěná autentizace a správa uživatelů:** Django poskytuje snadné způsoby, jak implementovat bezpečnou autentizaci uživatelů a správu jejich oprávnění, což je zásadní pro aplikace, které spravují citlivé obchodní data.
- **Podpora REST API:** Django REST framework je rozšíření Django, které usnadňuje vytváření API pro mobilní nebo webové aplikace.

2.3 Celery a Redis

Pro asynchronní zpracování úloh a sběr dat z obchodní platformy bylo použito **Celery** ve spojení s **Redis**. **Celery** je knihovna pro asynchronní zpracování úloh v Pythonu, která umožňuje spouštění úloh na pozadí, což je nezbytné pro sběr dat v reálném čase z forexového trhu. **Redis** funguje jako message broker, který umožňuje efektivní správu úkolů mezi různými pracovními procesy.

Klíčové parametry:

- **Asynchronní zpracování:** Celery umožňuje provádět úlohy na pozadí, což zajišťuje, že aplikace není blokována při sběru a analýze dat.
- **Škálovatelnost:** Redis je vysoce výkonný nástroj pro správu fronty úkolů, což umožňuje horizontální škálování systému pro zpracování větších objemů dat.
- **Podpora více backendů:** Celery podporuje různé message brokery a úložné systémy, což dává flexibilitu při výběru nástrojů pro konkrétní projekt.

2.4 Tailwind CSS

Pro návrh frontendového rozhraní bylo použito **Tailwind CSS**, moderní utility-first CSS framework, který umožňuje rychlé a efektivní vytváření responzivních a vizuálně atraktivních webových aplikací. Tailwind CSS byl vybrán pro svou jednoduchost, flexibilitu a schopnost přizpůsobit design bez potřeby psaní rozsáhlého vlastního CSS.

Klíčové parametry:

- **Rychlá tvorba rozhraní:** Tailwind poskytuje množství předdefinovaných CSS tříd, což umožňuje rychlé vytváření komplexních uživatelských rozhraní bez nutnosti psaní vlastního kódu.
- **Modularita:** Tento framework je vysoce modulární, což znamená, že lze snadno přizpůsobit design bez zbytečných komplikací.
- **Responzivní design:** Tailwind usnadňuje tvorbu rozhraní, které se automaticky přizpůsobí různým zařízením, což je zásadní pro moderní aplikace.

2.5 MetaAPI

MetaAPI je služba, která umožňuje přístup k historickým i aktuálním datům o forexovém trhu prostřednictvím API. Pro tento projekt byla MetaAPI zvolena pro svou schopnost poskytovat data v reálném čase a s minimální latencí, což je klíčové pro obchodní strategie, které vyžadují aktuální informace o cenových pohybech. Rovněž je výhodou cenová dostupnost, protože je veškeré používání MetaAPI zcela zdarma.

Klíčové parametry:

- **Rychlý přístup k datům:** MetaAPI poskytuje data v reálném čase s nízkou latencí, což je nezbytné pro modely strojového učení a automatizované obchodování.
- **Podpora různých timeframe:** MetaAPI umožňuje získat data v různých časových rámcích (např. 1 minuta, 5 minut), což je důležité pro analýzu a predikci cenových pohybů na forexovém trhu.

3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

V této kapitole jsou popsány konkrétní způsoby řešení úkolu, které byly aplikovány při vývoji systému pro obchodování na forexovém trhu. Vysvětlíme použité postupy, jakým způsobem byly implementovány klíčové komponenty systému, a jak byly testovány jejich funkčnosti. Dále budou uvedeny parametry systému, včetně výstupů modelu, a uvedeme schémata, obrázky a příklady kódu, které ilustrují realizaci jednotlivých částí systému.

3.1 Implementace obchodního systému

Celý obchodní systém je rozdělen na dvě hlavní části: manuální obchodování a automatizované obchodování řízené AI modelem. Základním postupem bylo nejprve implementovat manuální obchodování, které následně sloužilo jako základ pro integraci AI modelu pro automatizovanou obchodní strategii.

- **Manuální obchodování:** Uživatelé mohou zadávat obchodní příkazy prostřednictvím webového rozhraní, které je napojeno na backendovou část aplikace pomocí REST API. Pro odesílání obchodních příkazů byly využity HTTP metody jako POST (pro otevření pozice) a GET (pro získání relevantních dat).
- **Automatizované obchodování:** Pro implementaci automatizovaného obchodování byl použit model posilovaného učení (Q-learning). Tento model používá technické indikátory, jako jsou SMA, RSI a MACD, pro generování doporučení o obchodních akcích. Model se učí z historických dat a postupně optimalizuje své obchodní rozhodnutí.

Příklad kódu pro predikci obchodní akce pomocí AI modelu:

```
async def step(self, action, current_price, data_fetcher, symbol):
    reward = 0
    done = False
    save_model = False

    try:
        positions = await data_fetcher.get_positions()
        current_position = next((pos for pos in positions if pos['symbol'] == symbol), None)
        logger.debug(f"Current position check: {current_position}")

        if not current_position and not self.position:
            if action in [0, 1]: # BUY or SELL
                action_type = "ORDER_TYPE_BUY" if action == 0 else "ORDER_TYPE_SELL"
                result = await data_fetcher.execute_trade(action_type, symbol)
                if result and 'orderId' in result:
                    self.position = 'long' if action == 0 else 'short'
                    self.position_id = result['orderId']
                    self.entry_price = current_price
                    self.trades_count += 1
                    logger.info(f"Trade executed: {action_type}, ID: {result['orderId']}")
                else:
                    logger.error("Trade execution failed")

            elif current_position and action == 3: # CLOSE
                profit = float(current_position['unrealizedProfit'])
                logger.info(f"Attempting to close position, current P/L: {profit}")

                result = await data_fetcher.close_position(current_position['id'])
                if result:
                    reward = profit
                    self.total_profit += reward
                    self.position = None
                    self.position_id = None
                else:
                    logger.error("Position closing failed")

            if reward != 0:
                self.trades_count += 1
                if self.trades_count % self.save_threshold == 0:
                    save_model = True

        return reward, done, save_model

    except Exception as e:
        logger.exception("Error in trading step")
        return 0, False, False
```


3.2 Parametry systému

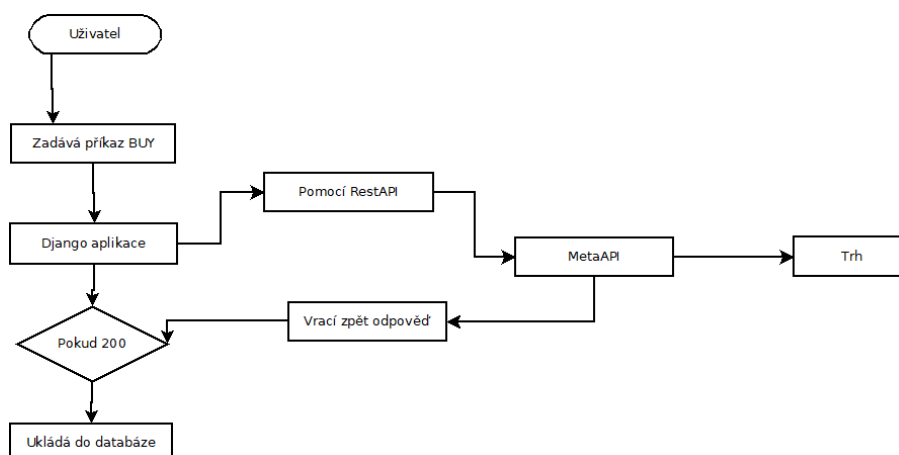
Systém je navržen tak, aby umožňoval flexibilní a efektivní obchodování s měnovými páry na forexovém trhu. Parametry systému zahrnují:

- **Funkcionalita:** Uživatelé mohou manuálně otevírat a uzavírat pozice na základě doporučení AI modelu nebo jejich vlastního rozhodnutí. Model pro automatizované obchodování se zaměřuje na maximalizaci zisku pomocí analýzy tržních dat a technických indikátorů.
- **Performance:** Systém je schopen zpracovat několik požadavků za sekundu pro otevření a uzavření pozic, což umožňuje obchodování v reálném čase.

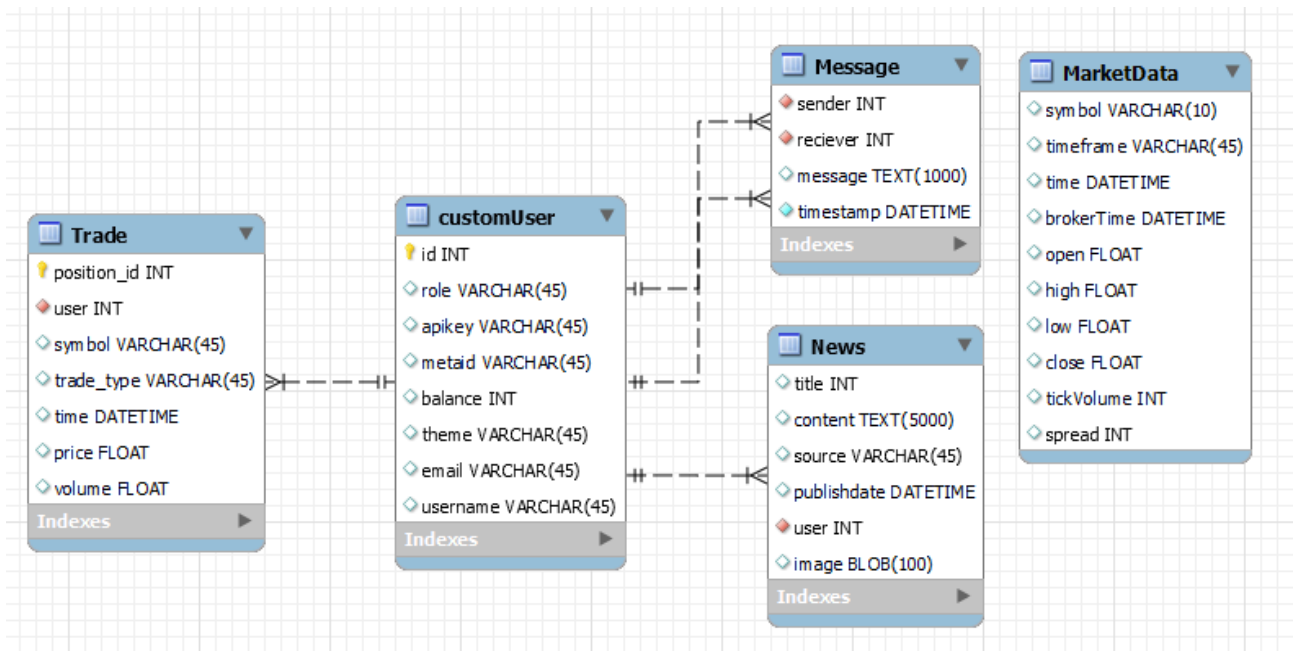
3.3 Schémata a obrázky

Pro lepší ilustraci byla vytvořena schémata, která znázorňují tok dat mezi jednotlivými částmi systému a vzorcové diagramy pro vizualizaci obchodního procesu. Následuje příklad schématu, které znázorňuje interakci mezi frontendovým rozhraním, backendem a obchodní platformou:

- **Schéma obchodního procesu:**
 - Uživatel zadá obchodní příkaz prostřednictvím webového rozhraní.
 - Backend přijme požadavek a ověří platnost.
 - Pokud je vše v pořádku, příkaz je odeslán na obchodní platformu přes REST API.
 - Systém monitoruje stav otevřených pozic a na základě tržních dat a AI modelu generuje doporučení pro jejich uzavření.



- Schéma databázového modelu



3.4 Použité příkazy

Pro komunikaci mezi backendem a obchodní platformou byly použity různé HTTP metody a příkazy. Například pro otevření pozice se použil POST požadavek, jak je uvedeno níže:

```
def execute_trade(user, symbol, trade_type, volume, take_profit=None):
    try:

        if not user.metaid:
            raise ValueError("MetaID is missing or invalid")

        url =
f"https://mt-client-api-v1.london.agiliumtrade.ai/users/current/accounts/{user.metaid}/trade"

        logger.debug(f"Trade URL: {url}")

        headers = {
            "Content-Type": "application/json",
            "Accept": "application/json",
            "auth-token": user.apikey,
        }

        payload = {
            "actionType": trade_type,
            "symbol": symbol,
            "volume": volume,
        }

        logger.debug(f"Headers: {headers}")
        logger.debug(f"Payload: {payload}")

        response = requests.post(url, headers=headers, json=payload)

        if response.status_code == 200:
            result = response.json()
            Trade.objects.create(
                user=user,
                symbol=symbol,
                trade_type=trade_type,
                volume=volume,
                price=result.get('price', 0),
                position_id=result.get('positionId')
            )

            return result
```

A pro uzavření pozice:

```
def close_trade(user, position_id):
    try:
        logger.info(f"Closing trade for user: {user.username}, API Key: {user.apikey}, MetaID: {user.metaid}")

        if not user.metaid:
            raise ValueError("MetaID is missing or invalid")
        position_url=f"https://mt-client-api-v1.london.agiliumtrade.ai/users/current/accounts/{user.metaid}/positions/{position_id}"
        headers = {
            "Accept": "application/json",
            "auth-token": user.apikey,
        }

        position_response = requests.get(position_url, headers=headers)

        if position_response.status_code == 200:
            position_data = position_response.json()
            profit = position_data.get('profit', 0)

            # Update the trade with the profit
            trade = Trade.objects.get(position_id=position_id)
            trade.price = profit
            trade.save()
        else:
            error_message = position_response.text
            logger.error(f"Error fetching position details: {error_message}")
            raise Exception(f"Error fetching position details: {error_message}")

        url = f"https://mt-client-api-v1.london.agiliumtrade.ai/users/current/accounts/{user.metaid}/trade"

        headers = {
            "Content-Type": "application/json",
            "Accept": "application/json",
            "auth-token": user.apikey,
        }

        payload = {
            "actionType": "POSITION_CLOSE_ID",
            "positionId": position_id,
        }

        response = requests.post(url, headers=headers, json=payload)

        if response.status_code == 200:
            result = response.json()
            logger.info(f"Trade closed successfully: {result}")
            return result
```

4 VÝSLEDKY ŘEŠENÍ, VÝSTUPY, UŽIVATELSKÝ MANUÁL

V této kapitole se zaměříme na zhodnocení dosažených cílů, přehled výsledků, které byly dosaženy během implementace projektu, a popis finálních výstupů. Také bude uveden uživatelský manuál pro interakci s obchodním systémem, jak pro manuální, tak pro automatizované obchodování.

4.1 Výčet splněných cílů

Během vývoje projektu byly splněny následující cíle:

- **Implementace manuálního obchodování:** Byla vytvořena funkční webová aplikace umožňující uživatelům zadávat obchodní příkazy. Tato část byla úspěšně testována a ověřena v reálných podmínkách.
- **Doporučení AI modelu pro obchodování:** Model posilovaného učení byl navržen, natrénován na historických datech a implementován do systému pro generování obchodních doporučení. Model dokázal úspěšně předpovědět vhodné obchodní akce (BUY, SELL, HOLD).
- **Vizualizace základních statistik:** Byly vytvořeny grafy pro vizualizaci vývoje ceny měnových párů, zisků a ztrát a dalších relevantních statistik. Tyto vizualizace byly integrovány do uživatelského rozhraní.

4.2 Nesplněné cíle

- **Plná automatizace obchodních strategií:** Tento cíl nebyl zcela dosažen. Systém automatizovaného obchodování je v plenkách a je zapotřebí ještě optimalizovat metody pro zlepšení predikcí a zajištění stabilního zisku.
- **Pokročilé vizualizace:** Ačkoli byly vytvořeny základní vizualizace, budoucí plány zahrnují interaktivní grafy a další metriky pro lepší analýzu a uživatelský zážitek. Tento aspekt projektu nebyl ve své plnosti implementován.

4.3 Výstupy a prokázání funkčnosti

Systém, jak je aktuálně implementován, vykazuje stabilní výkon a schopnost generovat obchodní doporučení na základě historických a aktuálních dat.

- **Testování modelu:** AI model, využívající metody posilovaného učení, se ukázal jako funkční pro předpověď trhu na základě tržních podmínek. Byla ověřena jeho schopnost učit se z historických dat a vygenerovat rozhodnutí pro obchodování.
- **Prokázání funkčnosti obchodního systému:** Manuální obchodování bylo úspěšně testováno s několika reálnými obchodními příkazy. Automatizované obchodování zatím probíhá v testovacím režimu, ale již generuje smysluplné predikce pro obchodování.
- **Výsledné parametry systému:** Systém je navržen pro zpracování obchodních příkazů v reálném čase s latencí menší než 1 sekunda. AI model dosahuje průměrné přesnosti ~30 % při rozhodování o nákupu nebo prodeji, což je v souladu s očekávanými výsledky pro první fázi vývoje.

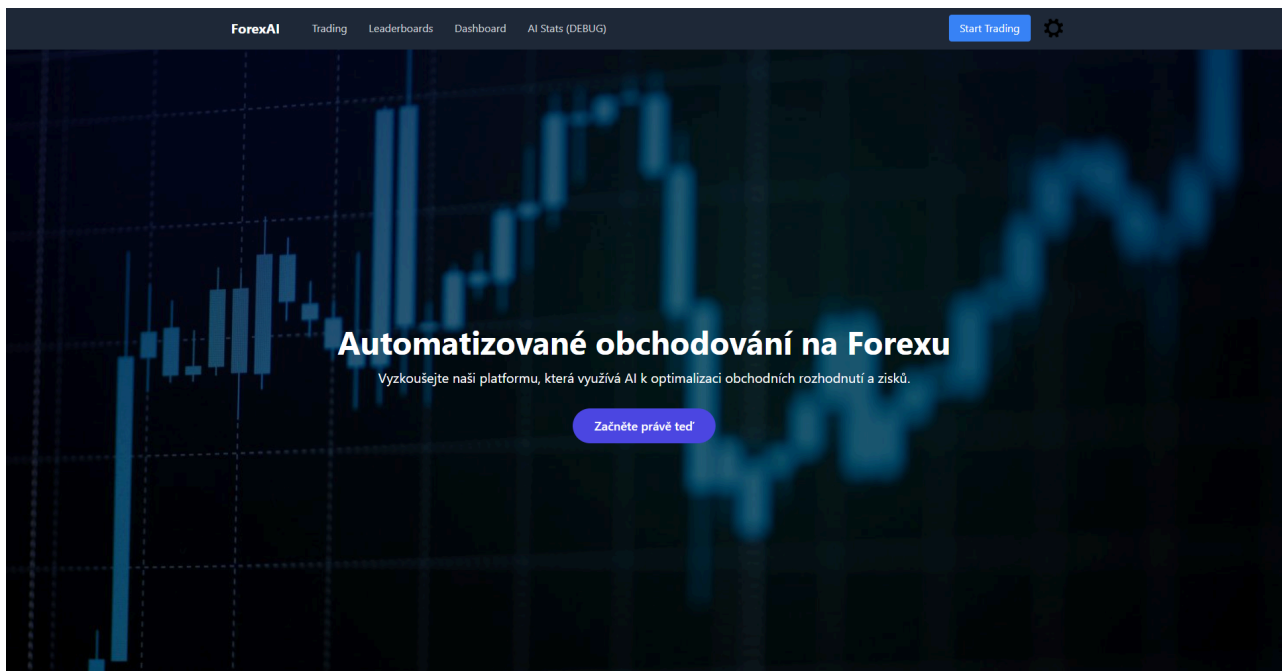
4.4 Uživatelský manuál

- **Registrace a přihlášení do systému:** Uživatel se musí zaregistrovat pomocí e-mailu a hesla. Po registraci je mu umožněn přístup k uživatelskému rozhraní. Poté může v nastavení uživatele zadat svůj API klíč a MetaID pro umožnění obchodování.
- **Manuální obchodování:** Uživatelé mohou otevřít obchodní pozici zadáním měnového páru, objemu a směru obchodu (BUY/SELL). Obchodní pozice mohou být uzavřeny kliknutím na odpovídající tlačítko.
- **Automatizované obchodování:** Tato část systému umožňuje uživatelům sledovat doporučení modelu. Uživatel se může podle něj rozhodnout a slouží zatím pouze jako nápověda.
- **Vizualizace dat:** Uživatel může zobrazit graf, který zobrazuje historický vývoj ceny.

4.5 Obrázky a schémata

Pro lepší orientaci jsou v manuálu přiloženy obrázky, které znázorňují rozhraní aplikace, kroky pro otevření a uzavření obchodní pozice.

Obrázek 1:



Zobrazuje uvítací stránku, toto je první co uživatel vidí po navštívení stránky.

Obrázek 2:

This is a registration form titled 'Create Account' with the subtitle 'Join our trading platform'. It contains four input fields: 'Username', 'Email', 'Password', and 'Confirm Password'. Below the 'Confirm Password' field is a blue 'Create Account' button. At the bottom, there is a link 'Already have an account?' followed by a green 'Sign In' button.

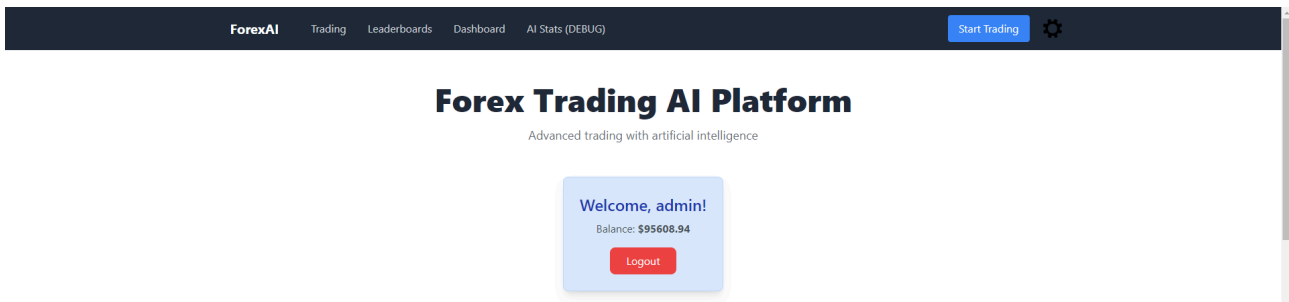
Zobrazuje formulář pro registraci.

Obrázek 3:

This is a login form titled 'Welcome Back' with the subtitle 'Sign in to your account'. It contains two input fields: 'Username' and 'Password'. Below the 'Password' field is a blue 'Sign In' button. At the bottom, there is a link 'Don't have an account?' followed by a green 'Create Account' button.

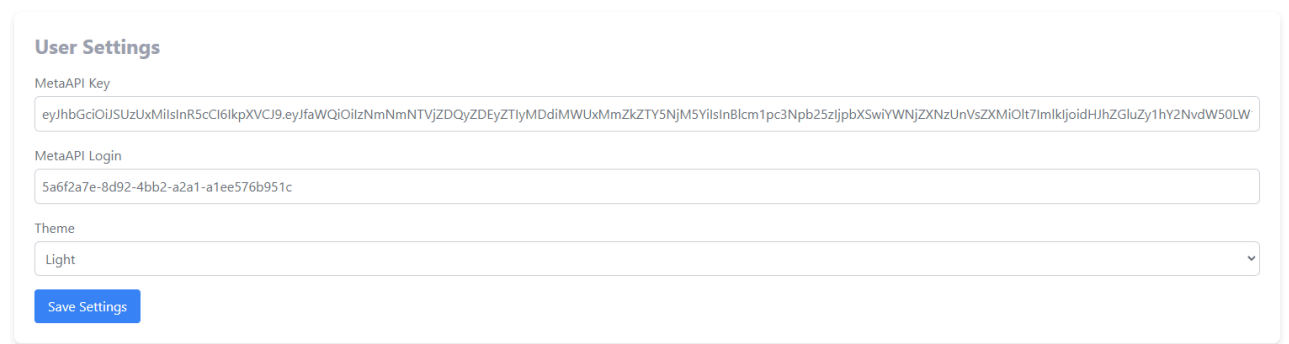
Zobrazuje formulář pro přihlášení.

Obrázek 4:



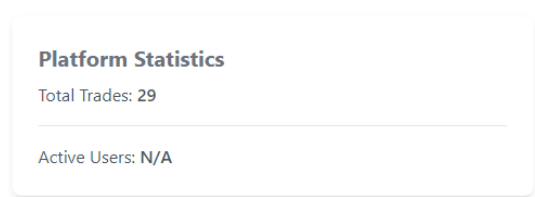
Zobrazuje uvítací obrazovku po přihlášení.

Obrázek 5:



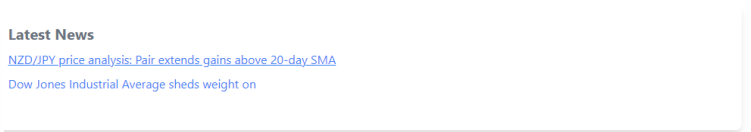
Zobrazuje uživatelská nastavení.

Obrázek 6:



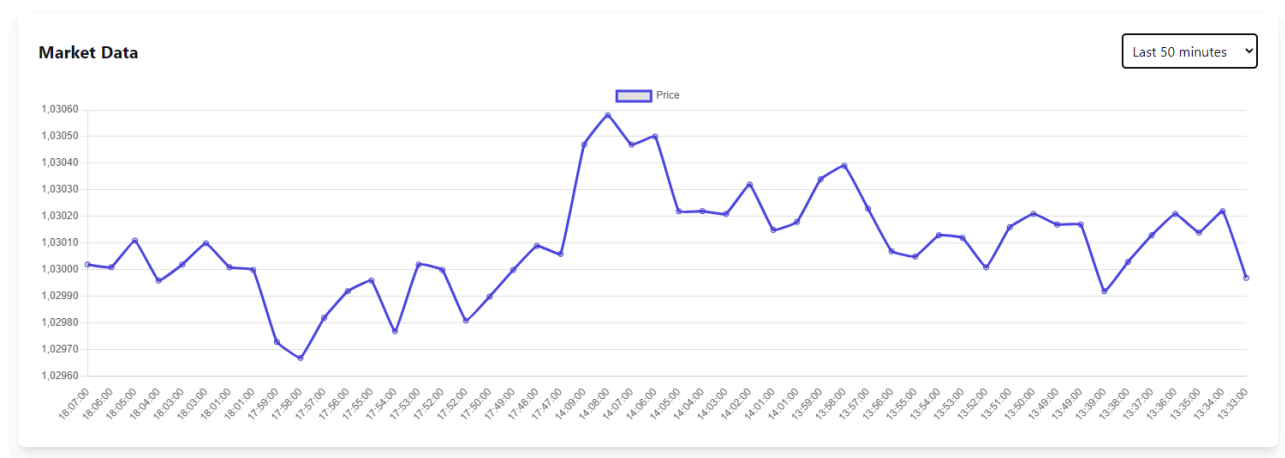
Zobrazuje statistiky.

Obrázek 7:



Zobrazuje nejnovější novinky.

Obrázek 8:



Zobrazuje graf s vývojem ceny.

Obrázek 9:

Current AI Opinion

SELL

Execute Trade

Symbol

Trade Type

Buy

Volume

Execute Trade

Open Trades











Symbol	Volume	Open Price	Current Price	Profit
No open trades				

Zobrazuje aktuální názor umělé inteligence, formulář pro otevření obchodu a otevřené pozice.

Obrázek 10:

Trading Leaderboards

Top performing traders on our platform

Top Traders				
Rank	Trader		Role	Balance
<div>1</div>	<div>A</div>	admin	<div>trader</div>	\$95558.94
<div>2</div>	<div>C</div>	cest	<div>trader</div>	\$3000.00
<div>3</div>	<div>M</div>	matej	<div>trader</div>	\$2000.00
4	<div>T</div>	tomas	<div>trader</div>	\$1500.00
5	<div>M</div>	maty	<div>trader</div>	\$1000.00
6	<div>A</div>	AI	<div>admin</div>	\$0.00
7	<div>J</div>	jetam	<div>trader</div>	\$0.00

Zobrazuje tabulku s nejúspěšnějšími uživateli.

ZÁVĚR

V rámci tohoto projektu jsme se zaměřili na vývoj systému pro forexové obchodování s důrazem na implementaci manuálního a automatizovaného obchodování s využitím umělé inteligence. Projekt splnil většinu stanovených cílů, včetně vytvoření plně funkčního rozhraní pro manuální obchodování, implementace základní automatizace řízené AI modelem a vizualizace obchodních statistik. AI model, využívající metody posilovaného učení, se ukázal být efektivní pro generování obchodních doporučení na základě historických dat, což prokázalo stabilitu a funkčnost systému.

I přesto, že systém byl úspěšně vyvinut a testován v reálných podmínkách, plná automatizace obchodních strategií není ještě zcela dokončena. V budoucnu je plánováno optimalizovat AI model pro zlepšení přesnosti předpovědí a dosažení vyšší profitability. Dalšími kroky budou i vylepšení vizualizací, jejich interaktivita a rozšíření o pokročilé metriky, což pomůže uživatelům lépe analyzovat trhy a provádět obchodní rozhodnutí.

Výsledný systém má potenciál pro široké uplatnění v oblasti automatizovaného obchodování, nejen pro jednotlivce, ale i pro malé a střední investory, kteří chtějí využít pokročilé algoritmy bez potřeby hluboké znalosti trhů. Automatizovaní obchodníci a analytici mohou systém integrovat do svých obchodních platforem pro zjednodušení procesu obchodování.

Budoucí vylepšení budou zaměřena na zajištění vyšší efektivity, ziskovosti a stabilnosti obchodních strategií. Tímto směrem se projekt vyvíjí, s cílem poskytnout silný nástroj pro obchodníky, kteří chtějí využívat moderní technologie pro zlepšení svých obchodních strategií a výsledků.

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- [1] *MetaAPI Dokumentace*. [online]. Dostupné na: <https://metaapi.cloud/docs/>
- [2] *TensorFlow Dokumentace*. [online]. Dostupné na: <https://www.tensorflow.org/api>
- [3] *Django Dokumentace*. [online]. Dostupné na: <https://docs.djangoproject.com/en/5.1/>

