

Semestrální projekt - Dolování dat z webových stránek s využitím AI nástrojů



Autor: **Bc. ONDŘEJ KOĐOUSEK**

Datum: **26. května 2025**

**Vysoká Škola Baňská - Technická Univerzita
Ostrava**

Fakulta Elektrotechniky a Informatiky

Vedoucí práce: **DOC. ING. GAJDOŠ PETR, PH.D.**

Obsah

1	Úvod	2
2	Současné technologie	3
2.1	Dolování dat	3
2.2	Analýza textu pomocí LLM	3
2.3	Predikce změn na burze pomocí AI	3
2.4	Současné nástroje, technologie, výzkumy	4
2.5	Využití lokální inference	4
2.6	LSTM	5
3	Návrh a implementace řešení	6
3.1	Scraper	7
3.2	Analyzátor LLM	7
3.3	Webová aplikace	8
3.4	LSTM	9
3.5	Správa databáze	10
4	Experimenty a výsledky	11
4.1	Způsob testování	11
4.2	Výsledky modelů	11
4.3	Výsledky zdrojů	13
4.4	Časová degradace predikcí	14
4.5	Výsledky LSTM	15
5	Závěr	16

1 Úvod

Cílem práce je prozkoumat možnosti analýzy sentimentu textu za účelem predikce změn cen akcií na burze pomocí LLM (Large Language Model) [1]. Hlavním záměrem je ověřit, zda je tento přístup prakticky použitelný a do jaké míry predikce odpovídají realitě. Cílem není vytvořit dokonalý predikční nástroj, ale spíše ověřit potenciál této metody a vyhodnotit její úspěšnost, pokud nějaká existuje.

Proces začíná automatizovaným sběrem dat z relevantních finančních zpravodajských serverů, jako je Yahoo Finance [2]. Investoři publikují články obsahující své poznatky a predikce týkající se dění na burze. Záměrem je tyto predikce získat, analyzovat a vyhodnotit jejich možnou přesnost.

Po získání těchto článků je text analyzován pomocí LLM. Pro lepší porovnání možností bylo použito hned 17 modelů, aby bylo možné najít nejvhodnější z nich a porovnat jejich výstupy. Pro inferenci jsou využívána API rozhraní, jako je Google AI Studio [3], OpenRouter [4] nebo Groq [5]. Experimentálně byla také testována lokální inference, aby bylo možné vyhodnotit její potenciální přínos.

LLM poskytne predikce změn pro dané entity na burze pro následujících 12 dní. Jedná se tedy spíše o krátkodobé předpovědi, jelikož i přesné články časem velmi rychle ztrácejí relevanci. Zároveň kvůli povaze práce by jedinou možností pro otestování výsledků bylo používat historická data. To by nejen komplikovalo implementaci, ale kvůli vznikajícím podobným řešením je možné, že současná burza se chová jinak než před několika lety [6].

Pro doplnění a porovnání jsem také využil predikce pomocí LSTM (Long Short-Term Memory) [7], který jsem trénoval na historických datech burzy. Po získání dat se tyto výsledky zobrazí ve webové aplikaci. Výstupem tedy není slepá predikce, do čeho má uživatel investovat, ale jedná se spíše o sadu nástrojů, které mohou zkušení investoři využít pro vytvoření vlastního obrazu o pohybu na burze. Aplikace vizualizuje predikce různých modelů, porovnává je s historickými daty a s predikcemi LSTM a také dokáže zobrazit sumarizované obsahy původních článků.

Tento přístup, který se snaží analyzovat sentiment, je velmi nejistý, jelikož i když se sentiment článku perfektně analyzuje, neznamená to, že tento sentiment je správný. Sentiment autora článku nemá žádnou autoritativní hodnotu. Je nerealistické vytvořit program, který by dokázal predikovat budoucnost. Přesto je cílem vyhodnotit, zda je tento přístup příliš nerealistický, nebo zda má nějaký potenciál – jedná se tedy spíše o proof of concept. Potenciální praktické využití by bylo hlavně v rozhodování ohledně krátkodobého obchodování a následně i možné uplatnění v automatizovaném HFT (High Frequency Trading) [8].

2 Současné technologie

2.1 Dolování dat

Pro získání relevantních finančních zpráv byl využit RSS[9] a následně knihovny Selenium[10] a BeautifulSoup[11] v jazyce Python. Primárním zdrojem byl RSS kanál od Yahoo Finance, který agreguje články z mnoha různých zpravodajských serverů zaměřených na finance. Selenium slouží k navigaci a interakci na dynamických stránkách, zatímco BeautifulSoup byl využit pro parsování HTML dokumentů. Z článků se použije celý text, nadpis a datum zveřejnění.

Scraping (dolování) má ale také etické aspekty. Co se týče legality, scraping se nachází v šedé zóně. Proto jsem nedoloval stránky, které se scrapingu bránily, i když by nebylo obtížné jejich zabezpečení obejít. Například Wall Street Journal detekuje scraping, a proto tento zdroj ignoruji. Cílem je získat veřejně dostupné informace pro akademické účely, nikoli obcházet platební brány nebo přetěžovat servery. Lze také argumentovat, že diskuze o etických i právních aspektech dolování dat je v dnešní době irelevantní, jelikož i světové korporace dolují celý internet pro trénování jejich AI a jejich modely jsou následně zkomerčňovány.

Zároveň s nárůstem popularity AI se postupně ukončovaly programy pro bezplatné akademické API, například na Twitteru, nebo bylo ukončeno CrowdTangle, které umožňovalo jednoduchý přístup k datům na sociálních sítích[12]. Proto jsem se zaměřil čistě na žurnalistické stránky, nikoli na sociální média.

2.2 Analýza textu pomocí LLM

Pro analýzu samotného textu, tedy obsahu zpráv, se využívají různé LLM modely. Bohužel je těžké získat přístup ke speciálním modelům natrénovaným přímo na analýzu finančních dat, každopádně to nemusí být nutně nevýhodou. Tímto lze ověřit, zda i generalizované modely si dokáží poradit s danou problematikou. V retrospektivě by aby bylo lepší využít modely jako FinBERT[13] pro referenční výsledky.

Nejnovější modely jsou přece jen mnohem robustnější a každým měsícem vychází lepší a lepší modely, trénované na stále větších datasetech, takže se neočekávají značné problémy s porozuměním textu, identifikací relevantních entit a celkovým významem zprávy bez nutnosti specializovaného trénování na finančních datech. Mimo jiné je dobré vyhodnotit, jak si obecné modely poradí s více specializovanými tématy.

2.3 Predikce změn na burze pomocí AI

Ve svém jádru se jedná o sentimentální analýzu textu, která je sama o sobě dosti nespolehlivá. Vytvořit správné predikce není jednoduchý úkol. Není nijak zaručeno,

že článek bude mít jakkoli správné a relevantní informace. Následně se tento pravděpodobně nepřesný článek analyzuje pomocí LLM, která se snaží tato potenciálně pochybná data analyzovat.

Lze očekávat, že výsledná data budou poměrně konzervativní, a že pro spolehlivější analýzu bude potřeba více článků, aby se vytvořil trend v predikcích. Hodnoty predikcí budou umělé, pokud je autor článku explicitně nevyčíslil. Jedná se tedy o značné riziko, každopádně si myslím, že tento experiment může mít významný přínos. I v horším případě bude alespoň možné rychleji vyhodnotit obsah článku, proto do výsledné aplikace nevložím pouze vyčíslenou predikci, ale i sumarizovaný text článku. Hlavní výhodou by tedy bylo ušetření času.

Mimo jiné je vhodné poznamenat současnou geopolitickou situaci, i přestože politická témata se do akademických prací nehodí. Bohužel kvůli současné situaci, která hraničí s manipulací burzy (insider trading), je celá burza extrémně nestabilní, a je tedy značně obtížné v této době vytvářet predikce.

2.4 Současné nástroje, technologie, výzkumy

Téma predikce pohybu na burze není nové, již před érou umělé inteligence se využívala výpočetní technika pro krátkodobé predikce. HFT využívá různé algoritmy k vydělávání na velmi malých cenových rozdílech.

S příchodem AI se toto téma začalo více rozvíjet. Vytvořily se specializované modely, jako například FinBERT, které se učily právě na finančních datech a dosahovaly tak lepších výsledků v analýze sentimentu. Většinou se používají buď data ze sociálních sítí jako Twitter, nebo ze zpravodajských serverů. Každopádně problematika je ve všech těchto výzkumech stejná – data se často nedají popsat lépe než jako pouhý šum.

Práce se téměř vždy zaměřují na značně krátkodobé predikce, často necelé 3 dny do budoucnosti[14], ojediněle i celý týden[15]. Tento přístup je logický, jelikož spolehlivost predikce se snižuje s postupem času a již pouhé krátkodobé predikce jsou obtížné.

2.5 Využití lokální inference

Pro projekt byl poskytnut NVIDIA Jetson AGX Orin Developer Kit[16] (dále jen Jetson), který byl původně zamýšlen pro jiné téma. Každopádně i v tomto tématu se přístroj využil. Jako první jsem ověřil výkon jeho inference. I přesto, že menší modely (pod 10B parametrů) byly zprovozněny v pořádku, větší modely nebyly použitelné. Menší modely, jak se dalo očekávat, dávaly nekvalitní výsledky a často měly problém vůbec pochopit svou roli. Zatímco větší modely buď nebylo možné zprovoznit, nebo jejich inference trvala desítky minut na jediný prompt.

Největší model, který jsem zprovoznil, byl distilovaný DeepSeek R1 Dynamic 1.58-bit[17], každopádně se tento model nedal rozumně použít a zpracovával přibližně 1/4 tokenu za vteřinu. Proto by reálně na analýzu jednoho průměrného článku bylo potřeba 20 až 40 minut, a vzhledem k tomu, že můj zdroj zpráv zveřejňoval v průměru 3,9 článku za hodinu, toto řešení by nestíhalo přísun nových zpráv. Samozřejmě jsou zde i další faktory, jako je například spotřeba elektřiny, i přesto, že Jetson je v tomto ohledu efektivní a režim nejvyššího výkonu údajně dosahuje pouhých 60W.

V případě přístupu k výkonnějším zařízením by lokální inference dávala smysl, zejména proto, že by bylo možné využít více specializované modely, jako je FinBERT. Každopádně tato možnost bez využití lokální inference odpadá, jelikož například právě FinBERT nemá mnoho vhodných možností pro spuštění přes API. Proto jsem jako alternativu zvolil využití inference přes API pomocí různých poskytovatelů. Mimo eliminaci potřeby počáteční investice tento přístup umožňuje jednodušeji využívat různé modely a tím mít větší přehled o tom, které modely jsou pro toto téma vhodnější a které méně.

Nakonec jsem Jetson využil také pro výpočet predikcí pomocí LSTM, ale tyto predikce byly zcela nezávislé na zbytku práce. Jetson jsem také využil pro hostování serveru a provádění zbytku procesů. Hlavní výhodou Jetsonu byla značně nízká spotřeba – pomocí wattmetru jsem naměřil hodnoty většinou v rozmezí 12 až 14W i při nejvyšší zátěži (vytváření LSTM predikcí a zároveň dolování dat).

2.6 LSTM

Long Short-Term Memory je typ rekurentní sítě (RNN) [18]. Tento neuronový model funguje na sekvenčních datech, například časových řadách, což je právě případ predikce cen na burze.

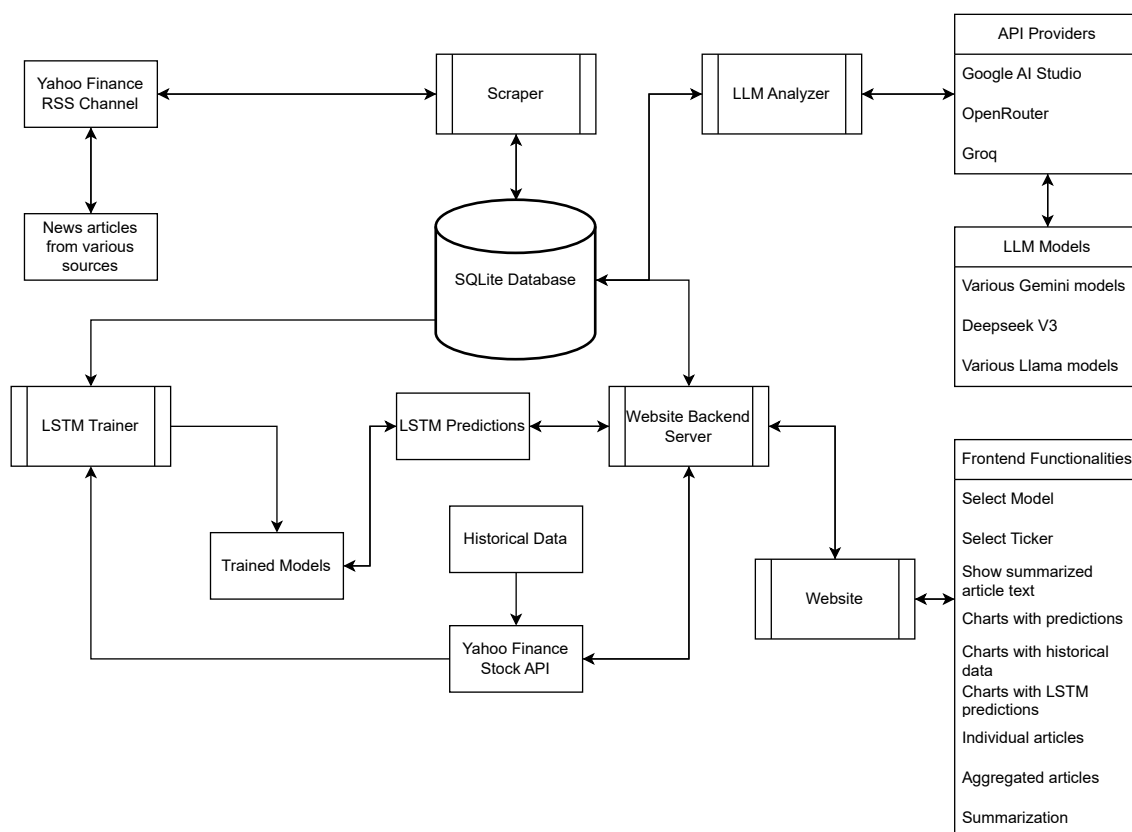
V tomto projektu jsem tuto technologii využil pro vytvoření nezávislých dodatečných predikcí. Existuje také možnost hybridního přístupu kombinujícího LLM analýzu s rekurentními modely [7], každopádně v mé práci jsem tohle použil spíše jako záložní plán, jako referenci a jako další dodatečnou informaci pro uživatele. Hlavní výhodou oproti LLM analýze je, že jazykové modely využívají nespolehlivá data ze zpravodajských serverů, zatímco rekurentní neuronové sítě využívají skutečná historická data, která nelze zpochybnit.

Tato metoda byla dobrá volba pro Jetson, jelikož díky tomu jsem přístroj významně využil, namísto toho, abych ho využíval pouze pro hostování serveru a spouštění jednoduchých skriptů. Analýza pomocí této sítě se provádí každých 24 hodin, aby poskytla co nejpresnější predikce na následující den. Využívá kontext posledních 60 dnů.

3 Návrh a implementace řešení

Implementace byla pro jednoduchost napsaná v Pythonu, webová aplikace v ReactJS a databáze je v SQLite. Proces se skládá z 6 hlavních částí:

- SQLite databáze - Jednoduchá databáze, která zajišťuje sdílený zdroj mezi všemi procesy
- Scraper - Dolování dat pomocí Pythonu a knihoven Selenium a BeautifulSoup. Doluje zprávy a ukládá jejich obsah do databáze.
- Analyzér - Program v Pythonu, který bere texty článků z databáze a posílá je přes API do různých LLM modelů. Po každém článku se aktualizuje sumari-zovaný report pro danou společnost.
- Webová aplikace - Aplikace vytvořená ve frameworku React pro zobrazení a zpracování výsledků.
- Web backend - Pro komunikaci mezi databází a webovou aplikací se používá Flask server
- LSTM - Vypočítává LSTM predikce



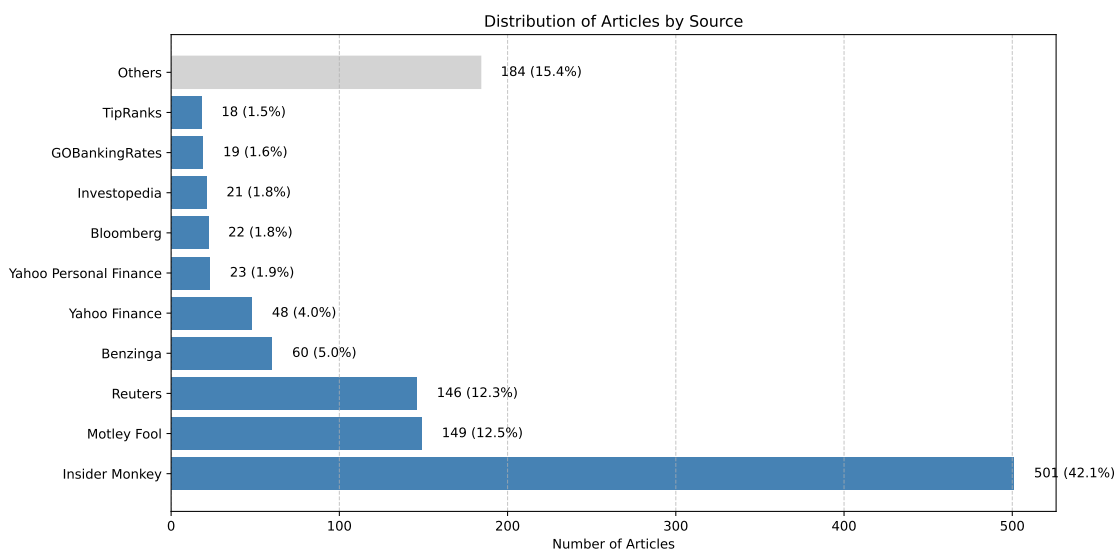
Obrázek 1: Architektura projektu, integrace mezi komponenty. Většina komponentů je nepřímá závislá na ostatních komponentech, např. není potřeba, aby při běhu LLM analyzáru běžel Scraper, pokud Scraper již do databáze uložil dostatek článků.

3.1 Scraper

Pro získání článků se využívá RSS feed z Yahoo Finance. Z tohoto feedu se postupně získávají nové články. Proces běží opakovaně, každých pár minut se spustí, zkontroluje, zda se v RSS feedu nachází nové články, a pokud ano, doluje z nich data.

Yahoo Finance agreguje články, tedy ve výsledku jsou články z přibližně 40 různých zdrojů. Několik zdrojů využívá ochranu proti dolování dat, a ty se tedy přeskakují. Zabezpečení lze obejít, každopádně pokusy o dolování dat ze stránek, které se tomu výslovně brání, jsou v šedé zóně.

Z článků se extrahuje text, titulek, zdroj a datum publikace. Tyto údaje se uloží do databáze a následně se zpracují dalšími procesy. Rozdělení zdrojů článků je vidět v grafu 2.



Obrázek 2: Poměr zdrojů článků. Zdroje, které měli méně než 1.5%, byly spojeny do společné kategorie.

3.2 Analyzátor LLM

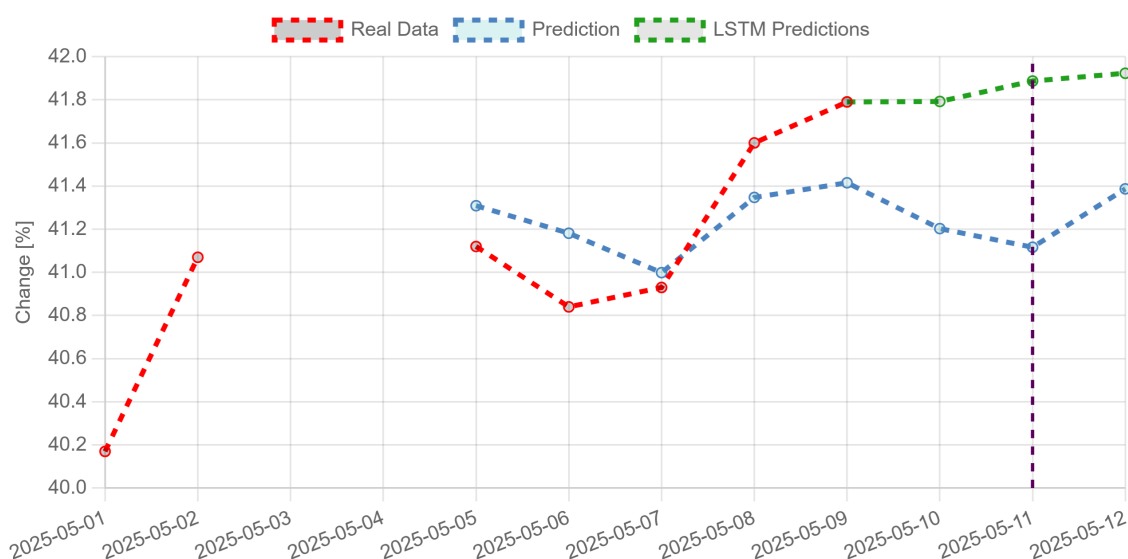
Podobně jako scraper proces, i analyzátor se opakovaně spouští. Vždy se vezmou uložené články z databáze, které ještě nebyly analyzovány, a seznam LLM modelů, kterými se dané články budou analyzovat. Pomocí různých API se vytvářejí požadavky a posílá se text článku. Využil jsem API od Google AI Studio, OpenRouter a Groq. Aby LLM věděly, co s textem mají dělat, využívá se systémová instrukce. Zde se dají odfiltrovat horší modely od těch sofistikovanějších, protože horší modely často ani nedokážou pochopit systémovou instrukci. Tato instrukce mimo jiné vyžaduje výstup v JSON formátu - konkrétně predikce v určitém časovém období a krátké shrnutí článku.

Bylo zapotřebí velkého úsilí jen na to, aby program byl robustní vůči nedeterministickým odpovědím od LLM. I přesto, že systémová instrukce explicitně uvádí, jak má model odpovědět, často LLM odpovídají nespecifikovaným způsobem. Například pokud model neuměl identifikovat, o které entitě se článek dominantně týká, měl vrátit hlášku *ERROR-01*. Každopádně občas vrátil tuhle hlášku s dodatečným textem nebo vrátil JSON formát s toutle chybovou hlášku vloženou do náhodných polí. Musel jsem proto implementovat systém na jejich co nejlepší zachycení. Kvalitní a úplná systémová instrukce měla značný vliv na kvalitu odpovědí.

Tato problematika se zpočátku zdála poměrně jednoduchá, každopádně zajistit spolehlivost a robustnost vůči nedeterministickým odpovědím byla značná překážka. Považuji to za podstatný problém při jakémkoli vývoji, kdy se využívají data z LLM.

3.3 Webová aplikace

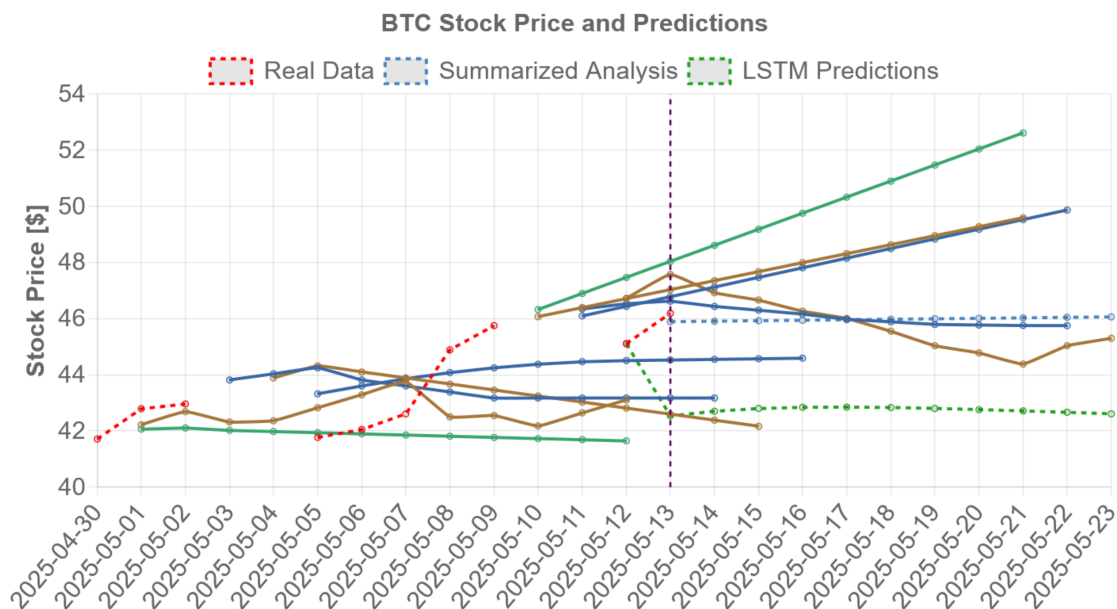
Očekává se, že aplikace bude použita zkušeným investorem. Nejedná se o aplikaci, která doporučí, co investovat, ale spíše o sadu nástrojů, které by potenciální investor mohl využít pro vytvoření vlastního obrazu o pohybu na burze. Mimo jiné se zobrazují historická data, aby bylo možné porovnat trend predikce s reálnými daty.



Obrázek 3: Ukázka grafu jednoho článku vytvořeného ve webové aplikaci. Fialová vertikální čára značí současný den. Červená křivka ukazuje reálná data z minulosti. Neobsahuje data ve všech bodech, jelikož burza je o víkendu uzavřená. Zelená jsou LSTM predikce, modrá jsou predikce od LLM.

Uživatel si jako první vybere model, kterým byly analýzy provedeny. Podle toho se v dalším sloupci zobrazí dostupné entity na burze. Tento sloupec také ovlivňuje filtr podle minimálního počtu analyzovaných zpráv. Na pravé straně se zobrazují

všechny články, s grafy které jsou vidět na obrázcích 3 a 4. Nadpis obsahuje odkaz na originál, pod ním se nachází sumarizovaný text článku a následně výsledná predikce a dostupná historická data. Pokud se tedy díváte na týden starý článek, uvidíte i historická data z tohoto týdne a můžete porovnat, jak moc byla nebo nebyla predikce pravdivá.



Obrázek 4: Ukázka agregovaného grafu. Přerušovaná červená čára jsou reálná historická data (mezery jsou víkendy), přerušovaná modrá je predikce sumarizace všech predikcí, zelená přerušovaná čára jsou LSTM predikce. Ostatní plné čáry jsou predikce vytvořené z jednotlivých článků. Stejně barvy znamenají stejný zdroj článku.

3.4 LSTM

Trénování probíhá maximálně v 25 epochách, přičemž se využívá early stop. Průměrný čas na natrénování je přibližně 192.8 vteřin. Se značnou rezervou lze tedy každý den natrénovat více než 400 modelů.

Trénování probíhá každý den a spouští se automaticky pomocí *Cron*. Je snaha vytvořit model pro každou společnost z databáze. Po natrénování se vygenerují predikce na dalších 12 dnů, které se následně uloží do databáze. Data se pak zobrazí jako další křivka v grafech na webové stránce. Skript vždy vezme 10 společností na burze, které mají nejvíce článků a zároveň ještě nemají vytvořené LSTM predikce pro daný den. Po dokončení se vezme dalších 10, dokud nejsou vytvořeny všechny predikce, nebo dokud nepříjde další den, a v tom případě proces začne od začátku.

3.5 Správa databáze

Projekt vyžadoval uchování určitých dat, ale nepotřeboval nic komplexního. Proto jsem využil jednoduchou databázi pomocí SQLite. Nebylo potřeba databázi nijak komplikovat, ve výsledku má pouze 5 tabulek, které lze funkčně rozdělit na 3 celky.

Tabulka *articles* uchovává samotné články a jejich dodatečné informace. Tabulka *analysis* ukládá všechny permutace analýz článků. Na jeden článek připadá N analýz, kde N je počet využívaných LLM modelů. Dále tabulka *predictions* uchovává jednotlivé hodnoty predikcí. Můžete si všimnout položky *confidence* – tento atribut se ve výsledku nevyužil. Původní plán byl, že LLM vrátí predikci a hodnotu, jak moc si je danou predikcí jistý. Každopádně to nikam nevedlo, ve výsledku to byla opět arbitrární hodnota, která by pouze vytvořila další vrstvu nejistoty výsledku.

Dále zde jsou tabulky *summarized_predictions* a *summarized_analysis*. Jak jejich názvy napovídají, uchovávají data o agregované analýze (analýze dané společnosti s využitím všech dostupných článků a jejich analýz). Opět se zde nachází nevyužitá položka *confidence*.

Jako poslední je tabulka *lstm_predictions*, která jednoduše uchovává data o LSTM predikcích. Predikují se následujících 12 dnů, přičemž data se aktualizují každý den.

Cron spouští každý den skript, který maže položky starší než 14 dní. Tím není databáze zbytečně plněna redundantními daty a není potřeba procházet databázi s velkým množstvím položek pokaždé, když uživatel aktivuje nějakou funkci webové stránky. Před implementací tohoto čištění databáze obsahovala více než 7 000 článků, více než 30 000 různých analýz a téměř 400 000 položek v tabulce *predictions*. Většina těchto dat byla starší než 2 týdny, a tedy obsah článků byl prakticky irelevantní.

4 Experimenty a výsledky

4.1 Způsob testování

Pro vyhodnocení jsem potřeboval nasbírat a správně zformátovat všechna data. K tomuto účelu jsem vytvořil funkcionalitu webové stránky, která postupně stahovala všechna dostupná data. Doba stahování všech dat se pohybovala mezi 30 až 60 minutami. Pro každou kombinaci modelu a entitě na burze, která měla alespoň jeden aktuální článek, se vytvořil a stáhl jeden JSON soubor obsahující reálná data ve všech relevantních dnech spolu s predikovanými hodnotami. Celkem to bylo v 6017 souborů v době, kdy jsem tyhle výsledky analyzoval.

Jako vzorek sloužil jeden pár reálné a predikované hodnoty ve stejný den pro danou entitu na burze a model. Ukládaly se predikce pro dny 1 až 12, proto z jedné analýzy článku mohlo vzniknout maximálně 10 vzorků (kvůli víkendům). Počet vzorků je dynamický. Pokud byl článek ze dne, kdy se data sbírala, poskytl 0 vzorků, protože vytvořené predikce byly pro budoucí dny, pro které neexistovala reálná data. Průměrné stáří článku odpovídalo polovině sledovaného období, tedy průměrný článek poskytoval 6 vzorků.

Jako hlavní metriku pro vyhodnocení úspěšnosti predikcí jsem použil MAPE (Mean Absolute Percentage Error), tedy střední absolutní procentní chybu. Tato metrika je nejvhodnější, protože nezávisí na absolutní hodnotě akcie a vždy porovnává rozdíl mezi reálnými a predikovanými hodnotami. Cílem je tuto hodnotu minimalizovat.

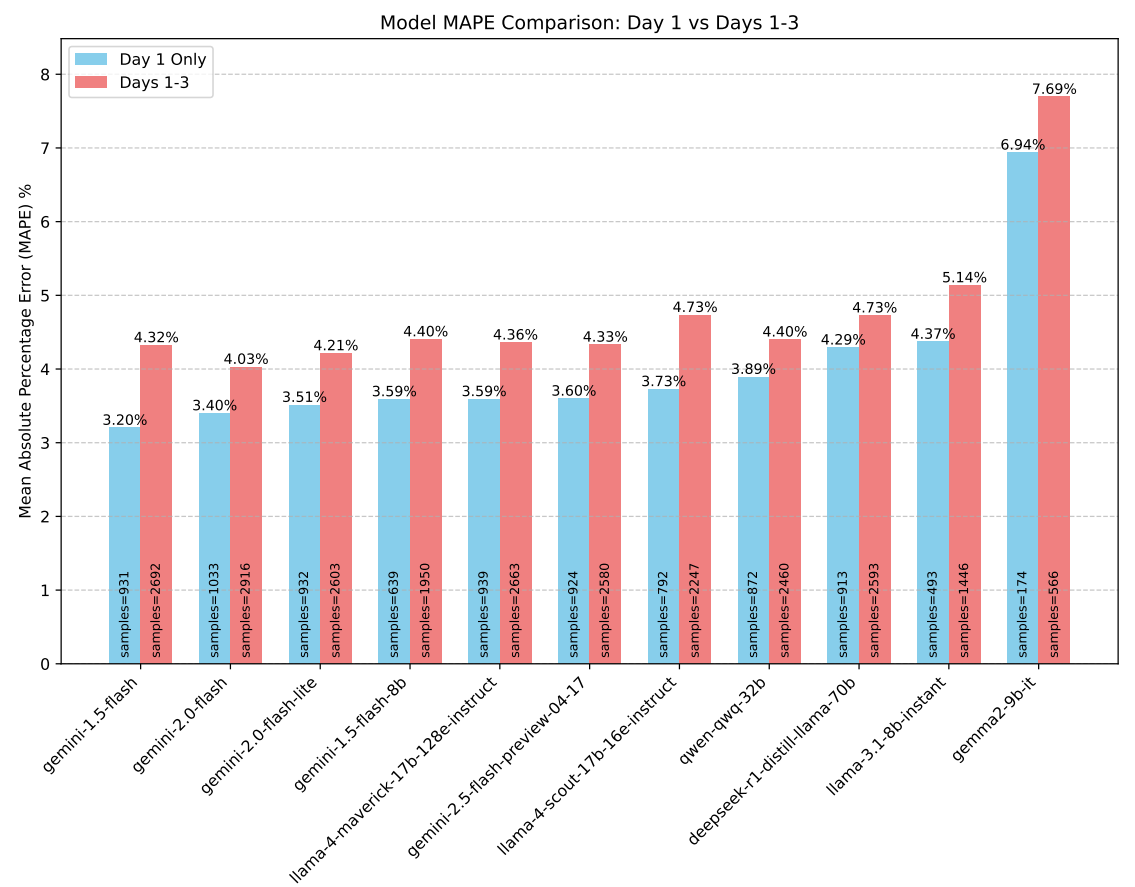
Cílem bylo vyhodnotit úspěšnost LLM modelů, zdrojů zpráv a časovou degradaci predikcí. Pro vyhodnocení LLM modelů a zdrojů zpráv jsem se zaměřil na první 3 dny predikce, ale pro vyhodnocení degradace predikcí v čase využil všech 12 dnů.

Výsledky v této kapitole jsou velmi obecné a představují zprůměrované hodnoty všech výsledků, tedy nebere se ohled na konkrétní entity na burze

4.2 Výsledky modelů

V grafu 5 je vidět, že kromě *gemma2-9b-it* zde není žádný značný rozdíl mezi modely. Graf ukazuje výsledky predikcí jeden den po zveřejnění jednotlivých článků. V případě zaměření se na první 3 dny se výsledky dle očekávání zhoršovaly, u všech modelů zhruba ve stejné míře. I širší záběr neměl značně odlišné úspěšnosti.

Nejúspěšnější model s dostatkem dat byl *gemini-1.5-flash* s 3.20% MAPE v první den, nejmenší úspěšnost, s podobným počtem dat, byl *deepseek-r1-distill-llama-70b* s hodnotou 4.29%. Podobný počet dat znamená, že oba modely porozuměly systémové instrukci i textu článku, ale přesnost hodnot jejich predikcí se lišily. Procentuálně je tento rozdíl 34%, což se dá pokládat za značný. Tedy modely, které správně



Obrázek 5: Graf porovnávající přesnost predikcí jednotlivých modelů. Modrá je výsledek pouze v první den, červená je výsledek ve dnech 1 až 3.

zanalyzují článek, jsou schopny vytvořit jinak přesné predikce, které nejsou pouze šum, ale opravdu mají jiné přesnosti predikce.

Tabulka 1: Rozdělení LLM modelů do skupin dle počtu vytvořených dat.	
Kategorie	Modely
1 - Dostatek dat	gemini-1.5-flash, gemini-2.0-flash, gemini-2.0-flash-lite, llama-4-maverick-17b-128e-instruct, gemini-2.5-flash-preview-04-17, qwen-qwq-32b, deepseek-r1-distill-llama-70b
2 - Méně dat	gemini-1.5-flash-8b, llama-4-scout-17b-16e-instruct, llama-3.1-8b-instant, gemma2-9b-it
3 - Kvótové limity	gemini-2.5-pro-preview-03-25, gemini-1.5-pro, deepseek-chat:free, llama-3.3-70b-versatile

Tabulka 4.2 má rozdělení všech použitých modelů dle počtu dat. Jak lze vidět, tak modely v kategorie 2 mělo značně méně dat než v kategorii 1. To indikuje velmi časté neúspěchy při analýze článků. Jedná se buď o nepochopení systémové instrukce a vracení nevalidního výstupu, a nebo neporozumění samotnému článku, například

když model nebyl schopen identifikovat, o jakou akcii se v článku jedná.

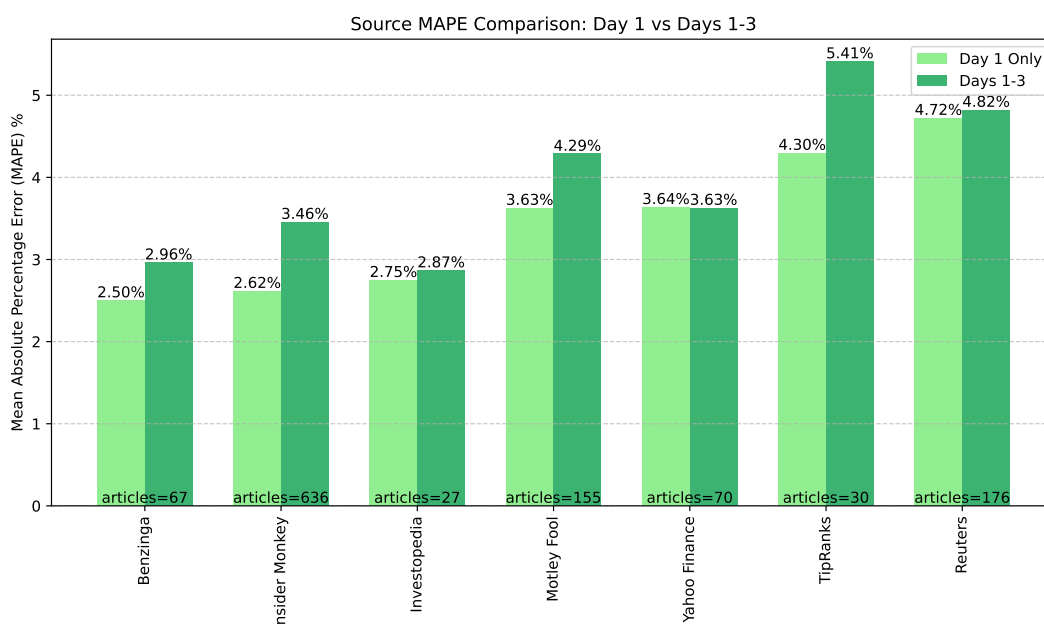
Neúspěchy v kategorii 2 se u všech modelů dají jednoduše připsat k tomu, že dané modelu jsou dost malé, největší z nich, *llama-4-scout-17b-16e-instruct*, má pouze 17B parametrů, zatímco zbytek nemá ani 10B. *gemma2-9b-it* dopadl nejhůře, povedlo se mu zanalyzovat $5\times$ méně článků než ty nejlepší modely, a i predikce z těch úspěšných analýz byly drasticky horší než jakýkoliv jiný model.

Data z kategorie 3 nelze využít, kvůli malému počtu dat. Narozdíl od kategorie 2 to není způsobeno modely, ale kvůli kvótám jejich API poskytovatelů. Jsou to více vyspělejší modely, takže se dá očekávat, že by měly podobnou či lepší úspěšnost než kategorie 1, každopádně bez dostatku dat tohle nelze potvrdit ani vyvrátit.

4.3 Výsledky zdrojů

Při vyhodnocení zdrojů byl zásadní problém v nevyrovnanosti, jak bylo vidět v grafu 2. V průběhu vývoje byl tenhle poměr značně odlišný, *Insider Monkey* zastupoval necelých 20% ze všech článků, ale nejspíš začal zveřejňovat více článků, a vyšplhal se až na 40% ze všech článků. Kvůli tohoto faktu, nelze dělat žádné závěry. Následující zhodnocení je tedy pouze návrh.

Jsou zde pouze 3 zdroje článků, u kterých se dá říct, že mají dostatek článků, a to *Insider Monkey*, *Motley Fool* a *Reuters*. U těchto zdrojů je poměrně značný rozdíl v úspěšnosti, *Insider Monkey* má 2.62% a nejvíce dat, zatímco *Reuters* má 4.72%. Zároveň u tohoto zdroje je velmi malý rozdíl mezi predikcemi v prvním dnu a ve dnech 1 až 3, což může značit, že predikce byly až moc konzervativní.

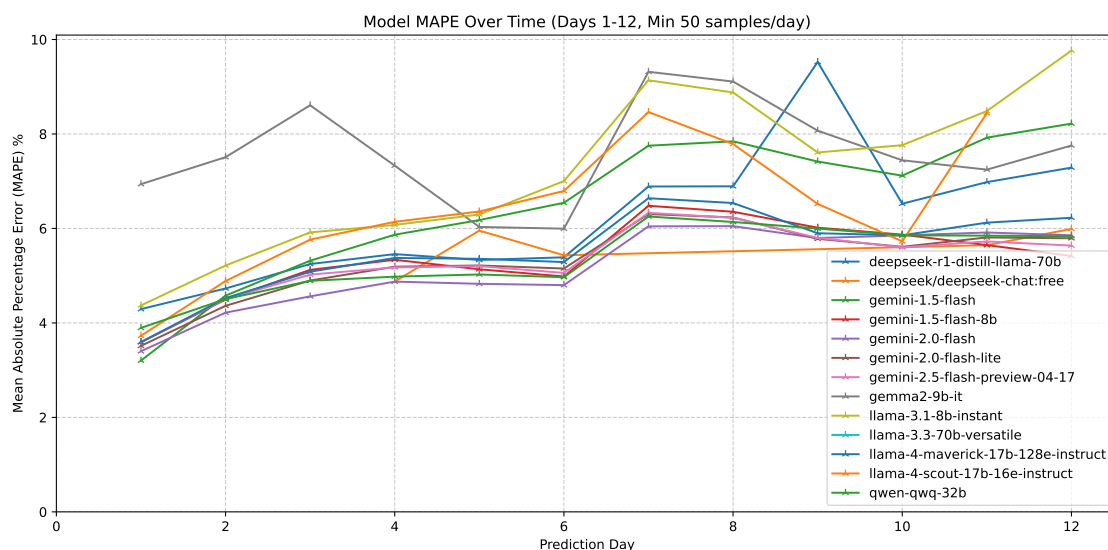


Obrázek 6: Graf ukazující úspěšnost predikcí podle zdroje zpráv s minimem 15 článků.

Rozdíly lze přičíst několika faktorům:

- Odlišné zaměření zdrojů zpráv – zdroje zaměřující se na stabilnější akcie mají obvykle lepší predikce
- Různé časové horizonty predikcí u jednotlivých zdrojů
- Jiný formát článků – některé obsahují konkrétní predikce, jiné jsou obecnějšího charakteru
- Rozdílná míra opatrnosti – některé zdroje vytvářejí články pouze při vysoké míře jistoty, jiné se zaměřují na množství

4.4 Časová degradace predikcí



Obrázek 7: Graf znázorňující pokles přesnosti predikcí v čase podle modelu. Minimum 100 vzorků na den. Dny 11 a 12 byly kvůli tomuto odfiltrovány. Modely v den 10 začínají divergovat, což je způsobeno menším počtem dat.

V grafu 7 je podle očekávání vidět, jak se přesnost predikcí s rostoucím časem zhoršuje. Výsledek byl ve dvou předešlých pozorováních mnohem méně chaotický, a téměř až lineární. Ale v tomto pozorování lze vidět značný výkyv mezi dny 6 a 7, a pomalé ustálení po dnu 9. Je zde poměrně nízký limit minimálního počtu vzorků, a při limitu 500 vzorků se výsledky chovají mnohem blíže k očekávání.

Je zde také vidět model s nejhorší úspěšností, a to je *gemma2-9b-it*. Tento model se i v předešlých pozorování choval velmi odlišně.

Každopádně lze stále počítat s logicky odvozeným předpokladem, že predikce se časem degradují. Zejméně jelikož LLM předpovídají ve většině případech velmi lineárně – tedy nedělají různé skoky, ale téměř vždy se jedná o lineární křivku, bez značných výkyvů v jejím směru.

4.5 Výsledky LSTM

LSTM predikce byly vytvořeny spíše jako záloha či dodatečné data pro uživatele. Jelikož LSTM predikce se aktualizují každý den, a vytvářejí predikce do budoucna, bylo by potřeba projekt předělat, aby bylo možné tyto predikce porovnat s opravdovými daty. Proto jsem místo toho vytvořil nový skript, které vytvoří predikce stejným způsobem, ale v minulosti, a tím je možné je porovnávat s existujícími historickými hodnotami. Výsledky jsou hodnoty ze 100 nejpopulárnější entit na burze (dle počtu článku v mé databázi). Tyto statistiky jsou v tabulce 2.

Průměrná úspěšnost v první den predikce byla 5.36%, zatímco nejlepší LLM predikce dosáhly 3.2%, podobně i u ostatních dnů. Výsledky LSTM byly zedy značně horší než výsledky LLM predikcí, každopádně testované data jsou starší, což by do jisté míry mohlo ovlivnit přesnost.

Tabulka 2: MAPE hodnoty testů LSTM predikcí

Day	Mean	Median	Minimum	Maximum
1	5.36	4.25	0.10	27.47
2	5.32	4.04	0.01	25.54
3	5.95	4.79	0.02	25.09
4	6.16	4.71	0.11	28.42
5	7.28	5.85	0.05	31.35
6	7.64	6.43	0.03	41.48
7	7.43	6.24	0.08	43.49
8	7.73	6.44	0.06	47.47
9	8.02	6.46	0.04	37.33
10	8.43	7.40	0.01	40.87
11	9.37	8.19	0.01	44.65
12	11.03	9.67	0.18	50.39

5 Závěr

I přes naivní premisu práce lze do jisté míry využít současné modely pro predikci pohybu na burze. V případě opravdového nasazení by hlavní výhodou bylo množství dat - i kdyby predikce měly horší úspěšnost než analýza článku zkušeným člověkem, rychlost a objem zpracovaných dat by mohly tento rozdíl kompenzovat.

Výsledky ukazují, že ačkoliv predikce mohou připomínat spíše nic neříkající šum, stále se zde objevují odchylky od průměrné úspěšnosti. Například při predikci na následující den, model *gemini-1.5-flash* dosáhl MAPE kolem 3,2%, zatímco nejhorší výsledek se stejným počtem dat (což znamená, že oba modely pochopily instrukce i článek, ale jinak vytvořili rozdílné hodnoty predikcí) je 4.29% od *deepseek-r1-distill-llama-70b*. Některé modely jsou tedy skutečně lepší v analýze článků a tvorbě predikcí, i když méně, než bylo očekáváno. Každopádně je potřeba větších modelů. Menší modely měli značné problémy s porozuměním systémové instrukce i článku textu.

Podobný trend platí i pro zdroje článků. Po odfiltrování zdrojů s malým počtem článků je patrný rozdíl mezi dvěma populárními zdroji. Například nejčastější zdroj *Insider Monkey* dosáhl MAPE 2,62% v první den, zatímco *Reuters* vykazoval výrazně horší výsledek 4,72%, takže analýza článků z *Insider Monkey* vede k lepším výsledkům. Tento jev samozřejmě může být způsoben dalšími faktory, jako je odlišné zaměření článků, jejich kvalita apod. Každopádně články byly silně nerovnoměrně rozděleny, a 40% ze všech článků bylo z *Insider Monkey*, takže tento závěr nelze s jistotou potvrdit ani zamítnout.

Experiment také potvrdil, že predikce do vzdálenější budoucnosti mají výrazně nižší úspěšnost než predikce na nejbližší dny. Tento závěr dává smysl i bez kontextu LLM, nicméně v této práci byl navíc kvantifikován.

Výsledky LSTM predikcí byly značně horší než LLM predikce nejlepšího modelu, dalo by se tedy říct, že tato metoda je přesnější než LSTM, každopádně má LSTM implementace nebyla dostatečně optimalizovaná a prozkoumána, a také se tyto dva způsoby testovaly na jiných datech.

V retrospektivě by bylo mnohem přínosnější místo vytváření LSTM predikcí využít Jetson na lokální inferenci FinBERT či podobného modelu, přímo zaměřeného na finanční analýzu, aby se získal referenční bod. V případném navázání by bylo prospěšné vytvořit tuhle referenci, a více se zaměřit na možnosti opravdového automatizovaného High Frequency Trading, či jiného přístupu, který by se dal lépe využít v praxi.

Reference

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Lukasz Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.
- [2] Yahoo. Yahoo finance - stock market live, quotes, business & finance news. <https://finance.yahoo.com/>, 2025. Accessed: 2025-03-15.
- [3] Google DeepMind. Google AI studio: Build with Gemini models. <https://ai.google.dev/>, 2025.
- [4] OpenRouter. Openrouter: Unified API for LLMs. <https://openrouter.ai>, 2025.
- [5] Groq. Groq: Fastest AI inference platform. <https://groq.com/>, 2025.
- [6] Sowmya Kotha. The transformative impact of artificial intelligence and machine learning on marketing operations. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 10:176–182, 11 2024.
- [7] Wen jun Gu, Yi hao Zhong, Shi zun Li, Chang song Wei, Li ting Dong, Zhuo yue Wang, and Chao Yan. Predicting stock prices with finbert-lstm: Integrating news sentiment analysis. In *Proceedings of the 2024 8th International Conference on Cloud and Big Data Computing, ICCBDC 2024*, page 67–72. ACM, August 2024.
- [8] Joel Hasbrouck and Gideon Saar. High-frequency quoting: Short-term volatility in bids and offers. *Journal of Financial and Quantitative Analysis*, 53(2):613–641, 2018.
- [9] RSS Advisory Board. Rss 2.0 specification. <https://www.rssboard.org/rss-specification>, 2023. Accessed: 2025-05-19.
- [10] SeleniumHQ. Selenium documentation. <https://www.selenium.dev/documentation/>, 2023. Accessed: 2025-05-19.
- [11] Leonard Richardson. Beautiful soup documentation. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>, 2023. Accessed: 2025-05-19.
- [12] Megan A. Brown, Andrew Gruen, Gabe Maldoff, Solomon Messing, Zeve Sanderson, and Michael Zimmer. Web scraping for research: Legal, ethical, institutional, and scientific considerations, 2024.

- [13] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models, 2019.
- [14] Kemal Kirtac and Guido Germano. Sentiment trading with large language models. *Finance Research Letters*, 62:105227, April 2024.
- [15] Yixuan Liang, Yuncong Liu, Boyu Zhang, Christina Dan Wang, and Hongyang Yang. Fingpt: Enhancing sentiment-based stock movement prediction with dissemination-aware and context-enriched llms, 2024.
- [16] NVIDIA. Nvidia jetson agx orin developer kit. <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/>, 2023. Accessed: 2025-05-19.
- [17] Unsloth. Deepseek r1 dynamic 1.58-bit. <https://unsloth.ai/blog/deepseekr1-dynamic>, 2023. Accessed: 2025-05-19.
- [18] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.