

Analýza akciových trhů

Stock Market Analysis

Patrik Odložilík

Bakalářská práce

Vedoucí práce: RNDr. Eliška Ochodková, Ph.D.

Ostrava, 2023



Zadání bakalářské práce

Student: **Patrik Odložilík**

Studijní program: B0613A140014 Informatika

Téma: **Analýza akciových trhů**
Stock Market Analysis

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je analyzovat data z finančních (akciových) trhů. Tato data budou reprezentovaná a analyzovaná jako síťová data. Pro práci s daty, analýzu a reprezentaci výsledků bude vytvořena vhodná webová aplikace.

1. Seznamte se s potřebnými metodami pro analýzu síťových dat.
2. Vyberte vhodná data z finančních trhů.
3. Navrhněte vhodný způsob jejich reprezentace jako dat síťových. Zohledněte časovou složku.
4. Navrhněte a implementujte webovou aplikaci pro analýzu dat a prezentaci výsledků analýzy.

Seznam doporučené odborné literatury:

- [1] Barabási, A.-L. & Pósfai, M. (2016), Network science, Cambridge University Press, Cambridge. <http://networksciencebook.com/>
- [2] Stephen, M., Gu, C. and Yang, H., 2015. Visibility graph based time series analysis. PloS one, 10(11), <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0143015>
- [3] Podle pokynů vedoucí bakalářské práce

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **RNDr. Eliška Ochodková, Ph.D.**

Datum zadání: 01.09.2022

Datum odevzdání: 30.04.2023

Garant studijního programu: doc. Mgr. Miloš Kudělka, Ph.D.

V IS EDISON zadáno: 07.11.2022 12:25:52

Abstrakt

Akciové trhy jsou komplexní systém, kde vzájemná závislost mezi akciemi ovlivňuje jeho chování. Pro analýzu akciových trhů existuje velký počet způsobů, z kterých si investoři mohou vybírat. Tato práce je zaměřená na analýzu akciových trhů za použití síťové analýzy, která je jedna z méně používaných způsobů analýzy akciových trhů. Cílem je webová aplikace, která umožňuje uživateli podle zvolených parametrů zobrazit potřebnou síť a její atributy. Součástí webové aplikace jsou nástroje, jako například tvorba akciového portfolia a jeho porovnání s indexem S&P500.

Klíčová slova

akciové trhy; síťová analýza; webová aplikace

Abstract

Stock market is a complex system where interdependence between stocks affects its behavior. There is a large number of ways that investors can choose from to analyze the stock markets. This work is focused on the analysis of stock markets using network analysis, which is one of the less used methods of stock market analysis. The goal is a web application that allows the user to display the necessary network and its attributes based on the selected parameters. The web application includes tools such as creating a stock portfolio and comparing it to stock index S&P500.

Keywords

stock market; network analysis; web application

Poděkování

Rád bych na tomto místě poděkoval vedoucí své práce RNDr. Elišce Ochodkové Ph.D, která mi poskytla neocenitelné rady, podporu, cenné vstupy při tvorbě práce a za čas, který mi věnovala.

Obsah

Seznam použitých symbolů a zkratek	7
Seznam obrázků	8
Seznam tabulek	9
1 Úvod	10
2 Akciové trhy	12
2.1 Akciový trh	12
2.2 Kapitálové trhy	13
2.3 Druhy tržní analýzy	13
2.4 Finanční portfolio	15
3 Teorie sítí	16
3.1 Základní pojmy	16
3.2 Speciální vlastnosti sítí	18
3.3 Sítě a jejich využití	20
4 Síťová analýza akiových trhů	24
4.1 Výběr dat pro síťovou analýzu	24
4.2 Získání a omezení potřebných dat	26
4.3 Příprava dat	29
4.4 Konstrukce sítí	30
4.5 LRnet algoritmus pro konstrukci sítí	32
4.6 Komunity v síti	34
5 Postup a výsledky síťové analýzy	36
5.1 Definice problému	36
5.2 Zkoumané období	37
5.3 Shrnutí dat	38

5.4	Výsledky	38
6	Webová aplikace	44
6.1	Proč webová aplikace	44
6.2	Základní bloky webových stránek	45
6.3	Webový server	47
6.4	Databáze	49
6.5	Klientská část aplikace	50
6.6	Knihovny a moduly	52
6.7	Funcionalita webové aplikace	53
7	Závěr	59
	Literatura	61

Seznam použitých zkratek a symbolů

AMEX	– American Stock Exchange
API	– Application Programming Interface
CSS	– Cascading Style Sheets
EPS	– Earnings per share
HTML	– Hyper Text Markup Language
HTTP	– Hypertext Transfer Protocol
JSX	– JavaScript XML
NASDAQ	– National Association of Securities Dealers Automated Quotations
NYSE	– The New York Stock Exchange
P/E	– price-earnings ratio
SQL	– Structured Query Language
SPA	– Single-Page applications)
UI	– User Interface

Seznam obrázků

2.1	Ukázka vzestupného trendu na svíčkovém grafu používaném při technické analýze . .	14
2.2	Footprint graf s periodou grafu jeden den	15
3.1	Vizuální reprezentace internetu z roku 2004 jako síť [3].	17
3.2	Neorientovaný graf o 8 vrcholech a 12 hranách [6].	18
3.3	Francouzská 400kV energetická síť [8].	21
3.4	Síť vytvořená pro analyzování k zachycení spammerů [9].	23
4.1	Příklad detekovaných komunit v síti [16]	34
4.2	Postup kroků algoritmu Louvain [17]	35
5.1	Síť konstruována s kosínovou podobností a obarvená podle sektorů	39
5.2	Porovnání sítí obarvených podle sektorů a nalezených komunit	40
5.3	Srovnání sítí konstruovaných s kosínovou podobností a Guaussian Kernelem pro období medvědího trhu 2011	40
5.4	Vývoj ceny v průběhu velké recese	42
5.5	Vizualizace průběhu tržního porpadu pandemie COVID-19	43
6.1	Databázový diagram tabulek použitých v databázi naší aplikace	50
6.2	Úvodní stránka aplikace pro vizualizaci a výběr sítě	54
6.3	Použití layout algoritmu ForceAtlas2 pro lepší vizualizaci sítě	55
6.4	Vizuální část sítě při tvorbě portfolia	56
6.5	Porovnání vytvořeného portfolia s akciovým indexem S&P500	57
6.6	Distribuce mír centralit vizualizované sítě	57

Seznam tabulek

4.1	Příjmy EPS, dividendy a P/E za první fiskální čtvrtletí let 2020 až 2022 firmy Advanced Micro Devices, Inc. (AMD)	25
4.2	Formát stažených dat z padnas_datareaderu	29
4.3	Ukázka transformovaných dat na vektorové	29
5.1	Obarvení vrcholů podle změny v ceně akcií	41

Kapitola 1

Úvod

Již dlouhou dobu jsou akciové trhy nedílnou součástí našich životů, protože se týkají našich finančních možností a rozhodnutí. Mnoho z nás má buď své vlastní akciové portfolio, anebo se účastní investičních fondů. Akcie také odrážejí momentální stav ekonomiky a trhu, kde pokles ceny akcií v důsledku ekonomické recese může negativně ovlivnit hospodářství a pracovní příležitosti. Tohle má vliv i na lidi, kteří s akciemi nikdy do styku nepřišli, anebo jsou s touto problematikou seznámeni, ale se svými financemi v tomto ohledu nijak nezacházejí.

Akcie mají obrovský vliv na finanční stránku nejen velkého množství investorů, ale také na rozhodování firem, které se snaží si dané investory udržet. Zároveň se hledá nespočet způsobů, jak akciové trhy analyzovat. Díky různým rozborům jsou pak investoři a firmy schopni dělat informovanější rozhodnutí a podpořit tak svoje finance, anebo také zabránit nečekaným finančním ztrátám.

Cílem této práce je podívat se na druh analýzy, který je v odvětví akciových, ale také jiných trhů používán velmi zřídka. Jedná se o síťovou analýzu, která je více rozvinutá spíše v jiných oborech. Když už se v oblasti akciových trhů tento typ analýzy používá, tak se jedná o takzvané korelační sítě, které jsou generovány z časových řad cen z vybraných akcií a časového období pro účel dané analýzy. V této práci se zaměříme na jiné metody než ty použité pro korelační sítě. Data pro analýzu však budou podobná, jen se budeme zabývat jiným výběrem akcií a časového období.

Analyzovat se budou data akciových trhů společností obchodovaných na amerických burzách NASDAQ, NYSE a AMEX. Pro analýzu budou zahrnuty společnosti, které mají historii obchodování minimálně od 1.1. 2007 až do 18.10.2022. V tomto časovém období se budeme dívat na tržní propady jako byla například velká recese nebo covidový propad v březnu 2020. Jelikož počet společností, které tyto kritéria splňují je hodně a nás spíše zajímají společnosti s větší tržní kapitalizací, které hýbou trhem, tak se omezíme na společnosti, které mají tržní kapitalizaci přesahující 10 miliard \$. Tato omezení nás dostala na 410 obchodovaných společností, jejichž historické data budou použita pro analýzu.

V aplikaci bude možné pro investory analyzovat jakékoliv období v dříve zmíněném časovém rozsahu. Investor bude mít možnost si zobrazit síť pro jednotlivé dny, měsíce a roky, kde bude mít

i možnost výběru počtu těchto časových prvků jako například 15 denní rozsah.

V naší analýze se podíváme na to, jestli jde vizuálně znázornit vztah, který mají akcie v rámci sektorů ve fyzickém světě, v našich konstruovaných sítích tak, že budou jednotlivé sektory tvořit skupiny v našem případě komunity. Vizualizované komunity sektorů budeme srovnávat s komunitami, které získáme použitím algoritmu Louvain. Další část analýzy se bude zabývat vizualizací průběhu vybraných tržních propadů. Budeme zde sledovat, v jakých sektorech prvotně začaly ztráty a jakým způsobem se tržní propad rozrůstal do jiných sektorů v průběhu času.

Celková struktura práce je následující. Prvotně se seznámíme s akciovými trhy. Podíváme se co to vlastně akciové trhy jsou a do jakého celku spadají. Nastíníme metody, které se používají pro analýzu akciových trhů a povíme si, co je to Následně se seznámíme s problematikou sítí. Řekneme si co to je síť, jaké máme druhy sítí a zadefinujeme si pár pojmů z teorie grafů a některé míry používané v sítích. V další kapitole se zmíníme o tom, jaké data budeme používat pro konstrukci sítí, jakým způsobem budeme data získávat, jaký typ dat použijeme a jejich přípravu do finální formy, kterou použijeme pro konstrukci sítí. Potom si povíme, jakým způsobem se konstruuje síť, co to jsou podobnostní funkce a jaký algoritmus použijeme pro konstrukci sítí pro naši analýzu a aplikaci. Další kapitola bude o analýze konstruovaných sítí, kde si povíme, jaké časové období jsme si vybrali, zadefinujeme se problémy, které chceme zkoumat a povíme si o výsledcích, kterých jsme dosáhli. Poslední kapitolou bude seznámení s pojmem webové aplikace, informace o použitých technologiích v naší aplikaci a informacemi o funkcionalitě naší aplikace.

Kapitola 2

Akciové trhy

V této kapitole se podíváme na to, co je to vlastně akciový trh, na jeho důležitost ve společnosti a také proč investoři vůbec s akciemi obchodují. Následně se zaměříme na to, do jakého celku akciový trh zapadá a na nejčastější druhy analýzy, které se používají při rozboru akciových trhů.

2.1 Akciový trh

Akciový trh je tržiště, na kterém se obchodují anebo vydávají akcie veřejných společností. Akcie jsou druhem podílu v podniku, a když jednotlivec nebo instituce akcie dané společnosti nakoupí, tak se stávají akcionářem neboli vlastníkem podílu veřejné společnosti, jejíž akcie koupili. Podle typu akcie pak daný jedinec nabývá jistých práv, která se liší podle druhu akcie, kterou drží. Mezi práva, kterých vlastník nabývá patří například oprávnění na zisk z dividendy. Velkým důvodem koupě akcií je pro investory potenciál získat významné návratnosti ze svých investic.

Akcie také slouží jako podkladové aktivum. Podkladové aktivum má svůj účel k odvození finančních derivátů, jako jsou opce, futures a swapy. Tyto finanční deriváty nemají vlastní hodnotu a jejich cena klesá nebo roste s cenou daného podkladového aktiva. Většina finančních derivátů funguje tak, že smluvená transakce proběhne v budoucnu. Finanční deriváty, ale mají velký vliv i na samotnou cenu obchodovaných akcií, kdy investoři a instituce používají deriváty, jako pojistku proti možným pádům cen podkladového aktiva, a to prodejem derivátu daného podkladového aktiva. Prodej derivátu ve velkém objemu může ovlivnit i cenu podkladového aktiva. Deriváty jsou tak nedílnou součástí akciových trhů. V aplikaci tyto instrumenty nejsou součástí analýzy, její zaměření je na akcie samotné.

Důležitost akciových trhů v globální ekonomice nejde přeceňovat. Jejich vývoj pečlivě sledují vlády, centrální banky a investoři po celém světě, protože může poskytnout informace o stavu ekonomiky a pomoci identifikovat trendy a potenciaální rizika. Poskytují obchodovaným společnostem přístup k finančním zdrojům, které mohou následně použít k financování nových projektů, rozšíření

provozu, k investování nebo celkové akvizici jiných společností a svým rozvojem tak navýšit cenu akcií [1].

V této práci se zaměříme na analýzu akcií amerických společností, které obchodují na burzách NASDAQ, NYSE a AMEX. Výběr byl dále zúžen na společnosti, jejichž celková tržní kapitalizace přesahuje 10 miliard \$. Jednou z částí analýzy ve webové aplikaci je taky analyzování historických tržních propadů jako je například velká recese, která začala v roce 2007. Je tedy potřeba zahrnout společnosti, které se na burzách obchodují již od 1.1.2007.

2.2 Kapitálové trhy

Krom akciových trhů existují i jiné druhy trhů. Samotný akciový trh se dá definovat jako část jednoho trhu, který nazýváme trh kapitálový. Některé z těch známějších částí kapitálového trhu jsou dluhopisové, komoditní a devizové trhy. Obchodování na těchto trzích funguje stejně jako na trzích akciových, akorát už investoři nezískávají podíl ve veřejných společnostech.

Dluhopisový trh je trh, na kterém se obchoduje s dluhopisy. Jedná se o finanční nástroje, které slouží jako půjčka a jsou splatné v určitém čase. Investoři, kteří nakupují dluhopisy, získávají pravidelné platby od emitenta v podobě úroků. Cena dluhopisů závisí na úrokových sazbách a kreditním hodnocení emitenta.

Komoditní trh je trh, na kterém se obchoduje se zbožím, jako zlato, stříbro, ropa a další suroviny. Investoři mohou nakupovat a prodávat komodity na burze, a to buď ve fyzické podobě, anebo pomocí finančních derivátů [1].

2.3 Druhy tržní analýzy

Za poslední roky, kdy se obchodování na kapitálových trzích přesunulo do elektronické formy, se vyvinulo mnoho způsobů, jak trhy analyzovat. Investoři se snaží vyvinout lepší a efektivnější způsoby analýzy, aby získali na trhu výhodu a dělali lépe informované rozhodnutí. V této kapitole si popíšeme 4 druhy analýzy, které se používají přímo pro rozbor akciových trhů, jsou jimi: technická, fundamentální, orderflow a síťová analýza.

2.3.1 Technická analýza

Technická analýza by se dala definovat jako studie chování trhů, primárně za použití svíčkových grafů, pro předpověď budoucího chování trhu. Termínem chování trhu je myšlena změna ve dvou primárních hodnotách, které tvoří výsledný graf, nad kterým se analýza provádí. Těmito hodnotami jsou cena a objem prodaných aktiv. Cena a objem jsou také hodnoty, které jsou použity při konstrukci sítí pro analýzu v naší aplikaci. Při technické analýze derivátů se analyzuje ještě třetí hodnota, kterou je open interest, určující počet neuzavřených obchodů.

Při technické analýze se investor snaží v grafech identifikovat opakující se vzory, anebo zdali existuje v trhu nějaký trend. Trendy jsou velkou součástí technické analýzy a spousta prvků je založená na předpokladu, že trendy mají tendenci přetrvávat. Příklad grafu, kde je identifikován trend můžeme vidět na obrázku 2.1. Identifikování těchto vzorů a trendů pomáhá investorům dělat informovanější rozhodnutí při koupi a prodeji aktiv. Často se technická analýza používá spíše jako podpůrná analýza, a to k analýze fundamentální [2].



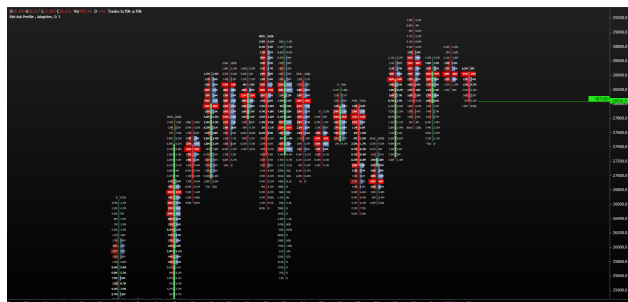
Obrázek 2.1: Ukázka vzestupného trendu na svíčkovém grafu používaném při technické analýze

2.3.2 Fundamentální analýza

Stejně jako u technické analýzy jde o metodu analýzy, která se používá pro předpověď budoucího chování trhu. Místo analyzování grafů se zde investor zaměřuje na možné ekonomické a finanční faktory, které mohou ovlivnit cenu aktiva. Je založena na předpokladu, že skutečná cena aktiva lze odhadnout zkoumáním těchto faktorů, jako jsou například finanční výkazy společnosti, trendy v odvětví, ekonomické ukazatele a geopolitické události. Zkoumáním jsou pak investoři schopní zjistit, zdali je momentální cena aktiva na trhu vyšší nebo nižší než skutečná cena aktiva. Pokud je skutečná cena získaná z fundamentální analýzy menší než cena, za kterou se zkoumané aktivum obchoduje na trhu, jedná se o kupní signál, pokud je cena vyšší, znamená to, že aktivum je předražené a investor by měl aktivum, pokud jej vlastní, prodat [1].

2.3.3 Orderflow analýza

Orderflow analýza, taky nazývaná analýza toku objednávek, umožňuje analyzovat objem a velikost objemu na kupní a prodejní straně. Tyto informace je možné zobrazit na specializovaných grafech zvaných footprint charts. Investor je schopný sledovat rozdíly v kupním a prodejním objemu na úrovni nastavené periody grafu, ale taky na úrovni ceny. Tento pohled umožňuje získat přehled nad tím, zdali se cena hýbe z důvodů snížené likvidity, anebo se na trhu objevil agresivní obchodník. Footprint graf je možné vidět na obrázku 2.2, kde oproti svíčkovým grafům používaným při technické analýze se na levé straně svíčky nachází velikost objemu prodeje a na pravé straně svíčky velikost objemu nákupu.



Obrázek 2.2: Footprint graf s periodou grafu jeden den

Analýza toku prodeje se často přirovnává k technické analýze, jde totiž spíše o rozšíření technické analýzy o již zmíněném rozdělení objemu na kupce a prodejce. Kromě těchto metrik se také analyzuje kniha objednávek. V knize objednávek je zobrazená momentální likvidita trhu, jak na straně kupců, tak na straně prodejců.

2.4 Finanční portfolio

Část naší webové aplikace se zabývá konstrukcí akciové části portfolia, kdy z vytvořených grafů tvoříme podle zadaných parametrů podgrafy. Vzniklý podgraf nám pak reprezentuje výsledné portfolio akcií. Vytvořené portfolio pak můžeme porovnávat proti existujícím akciovým indexům a sledovat tak jejich návratnost.

Portfolio je kolekce aktiv, které si investoři skládají za účelem zisku. Tato kolekce aktiv by se měla skládat z různých typů aktiv a ne jenom jednoho druhu aktiv, jako jsou například akcie. Jeden z důvodů této diverzifikace je to, že aktiva jednoho typu jsou mezi sebou často velmi korelovaná. Díky diverzifikaci je investor schopný lépe ochránit svoje portfolio před možnými ztrátami.

Kapitola 3

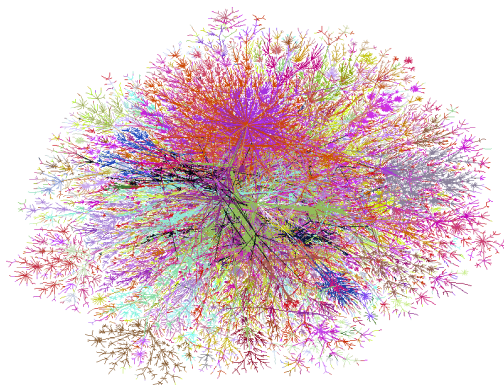
Teorie sítí

V úvodní části této kapitoly je potřeba se seznámit s pojmem síť a co to síť vlastně vůbec je. Zmíníme pár praktických příkladů sítě a co tvoří síť v naší webové aplikaci. Jelikož pojmy síť a graf se často používají zaměnitelně, tak si řekneme jejich rozdíly. Dále se seznámíme s určitými mírami, které se můžou používat například pro vyhodnocení důležitých prvků v síti. Kromě použití sítí pro zkoumání akciových trhů mají sítě nespočet využití v jiných oborech, proto si v jedné části povíme o různých typech komplexních sítí.

3.1 Základní pojmy

Sítě jsou všude kolem nás, dokonce my sami jsme součástí sítí, ať už se jedná o sociální vztahy, vztahy rodinné, tak náš biologický systém je výsledkem komplexní sítě biochemických reakcí. Síť si definujeme jako objekt skládající se z uzlů (vrcholů) a vztahů nebo interakcí mezi nimi (hran). Vrcholy můžou také obsahovat atributy, kde atributem chápeme nějakou vlastnost uzlu. Atributy je pak možné použít například při vizualizaci sítě, kdy podle daného atributu uzel obarvíme. Uzly však nejsou jediná část sítě, která může mít atributy. Atributy mají taky hrany, kde nejčastějším atributem pro hrany je váha dané hrany, kterou většinou získáme při konstrukci naší sítě. Skvělým příkladem sítě jsou například již zmíněné vztahy, kdy uzlem je člověk a jako hrana je definován vztah mezi nimi. Příkladem sítě samozřejmě nejsou jenom mezilidské vztahy, například na obrázku 3.1 můžete vidět vizuální reprezentaci internetu jako sítě, kde vrcholy tvoří zařízení a hrany jsou propojení mezi nimi.

Například uzly v sítích použitých v naší webové aplikaci a analýze budou reprezentovány akciemi. Zatímco uzly lze jednoduše definovat, tak definovat interakce, nebo vytvářet hrany pro akcie už není tak intuitivní, na rozdíl od některých jiných sítí, jako jsou sítě vztahové, kde lze interakci mezi uzly definovat explicitně. Nejtradičnějším způsobem definice hrany rozdílnými uzly akciových trhů je podívat se na vzájemnou korelaci některých definovaných atributů (cena, objem) pro zadanou



Obrázek 3.1: Vizuální reprezentace internetu z roku 2004 jako síť [3].

časovou periodu. V naší analýze se podíváme na jiné podobnostní metody pro konstrukci sítě, o kterých se zmíním v dalších kapitolách.

V matematice může být síť reprezentována jako graf, a proto teorie sítí úzce souvisí s teorií grafů. Teorie grafů je matematické studium grafů, což jsou struktury, které se skládají z množiny vrcholů a množiny hran, které spojují dvojice vrcholů.

Jednoduchý graf G lze definovat jako $G = (V, E)$, kde V je neprázdná množina vrcholů a E je množina hran - množina dvouprvkových podmnožin množiny V . Vrcholy se často označují malými písmeny jako jsou například u a v . [4].

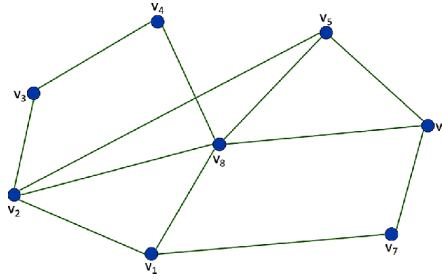
V teorii grafů existuje mnoho různých typů grafů, z nichž každý má svou vlastní sadu charakteristik a vlastností. V rámci naší práce se budeme zabývat právě jednoduchými grafy. V jednoduchém grafu nerozlišujeme hrany (u, v) a (v, u) . Grafy, které takto hrany rozlišují se nazývají orientované a v této práci s nimi pracovat nebudeme.

Nutno také zmínit, že jednoduchý graf neobsahuje smyčky a multihrany. Smyčkou je myšlena hrana, která má oba koncové vrcholy stejné (u, u) . Smyčky v grafech použitých pro analýzu v naší práci neexistují, protože neexistuje žádný vztah (hrana), který by spojoval akcii sám se sebou. Multihrany jsou označení pro hrany, které spojují stejné vrcholy a mezi dvěma vrcholy může tedy existovat více hran. Graf obsahující multihrany se nazývá multigraf [5].

Při vizualizaci grafu na 2D scéně se uzly typicky vykreslují jako puntíky a jednotlivé hrany se kreslí jako čára mezi párem uzlů. Vizualizaci je možné vidět na obrázku 3.2, kde máme vyobrazený neorientovaný graf o 8 vrcholech, které spojuje 12 hran. Obrázek popisuje zmíněnou strukturu $G = (V, E)$, kde množina vrcholů je $V = (v1, v2, v3, v4, v5, v6, v7, v8)$ a množinu hran tvoří $E = ((v1, v2), (v1, v7), (v1, v8), (v2, v3), (v2, v5), (v2, v8), (v3, v4), (v4, v8), (v5, v8), (v5, v6), (v6, v7), (v6, v8))$

Jeden z důležitých pojmů v teorii grafů je stupeň vrcholu. Jde o vlastnost vrcholu, která nám udává počet vrcholů, s kterými má daný vrchol společnou hranu. O vlastnosti se zmiňuji, jelikož je to jedna z mír, která se v naší aplikaci používá pro tvorbu podgrafu akcií portfolia.

Stupeň vrcholu: Pokud je hrana (u, v) v grafu G , tak vrcholy u a v nazýváme koncové vrcholy této hrany a říkáme, že vrcholy u a v jsou s hranou (u, v) incidentní. Stupeň vrcholu v v grafu G je



Obrázek 3.2: Neorientovaný graf o 8 vrcholech a 12 hranách [6].

pak definován jako počet hran, se kterými je vrchol incidentní [4].

Praktickým příkladem stupně může být například počet přátel, které máme na sociální síti. Stupeň se zde zajímá pouze o daného uživatele a nebere v potaz danou sociální síť jako celek.

V grafu na obrázku 3.2 můžeme vidět, že všechny vrcholy v tomto grafu mají stupeň větší než 2 a například vrcholy v_2 a v_8 mají stupeň $k_{v_2} = 4$ a vrchol $k_{v_8} = 5$.

Jelikož v další části kapitoly se zabývám centralitama sítě, tak je potřeba se seznámit s definicemi sledu, tahu a cesty v grafu [4].

Sled: Sled v grafu G je taková posloupnost vrcholů a hran $(v_0, e_1, v_1, e_2, v_2, \dots, e_n, v_n)$, že hrana e_i má koncové vrcholy v_{i-1} a v_i pro všechna $i = 1, 2, \dots, n$. Sled v grafu můžeme chápat jako záznam putování nebo bloudění v grafu.

Tah: Tah je sled, ve kterém se žádná hrana neopakuje.

Cesta: Cesta v grafu je sled, ve kterém se neopakují vrcholy.

3.2 Speciální vlastnosti sítí

Při studiu komplexních sítí se běžně používá několik síťových mír ke kvantifikaci jejich vlastností. Různé míry jsou navrženy tak, aby odpovídaly na konkrétní otázky. Jedna z takových otázek řeší míra zvaná centralita, která nám pomáhá identifikovat důležité uzly v síti. Třemi běžně používanými mírami centrality jsou stupeň, closeness centralita a mezistřednost (betweenness centralita). Další možnou mírou je míra zvaná shlukovací koeficient, která nás může zajímat v případě sociálních sítí, kde například pomáhá řešit otázku toho, jestli se naši přátelé přátelí i mezi sebou.

V jedné z částí webové aplikace je možnost si některé z těchto mír zobrazit a použít je tak k analýze zkoumaných sítí.

3.2.1 Centrality

Jedny ze zkoumaných mír jsou centrality. Centralita je velmi významná míra při analyzování grafů. Používá se pro určení důležitosti uzlů v grafu. Důležitost může ale být definována více způsoby. Máme tedy centrality, kde každá centralita má definovanou důležitost uzlu jiným způsobem. Každý

uzel může tedy být podle kontextu jinak důležitý, záleží na tom, jakou míru centrality zrovna zkoumáme.

V této práci se podíváme na tři míry centrality:

1. **Degree Centrality:** stupňová centralita je míra, kde důležitost uzlu v grafu závisí na stupni uzlu v grafu. Dá se tedy říct, že čím větší stupeň má uzel v grafu, tím je z pohledu této centrality důležitější.
2. **Closeness Centrality:** Closeness centralita řeší důležitost uzlů v grafu podle toho, jak blízko jsou uzly vůči ostatním uzlům. Pro určení této míry je ale potřeba zmínit délku nejkratší cesty. **Délka nejkratší cesty** nám určuje vzdálenost mezi dvěma uzly. Je to vzdálenost, která udává počet hran mezi dvěma uzly na nejkratší cestě mezi nimi. Nejkratší cestou je myšlena cesta s minimálním počtem hran.

Délka nejkratší cesty: je definována jako počet hran mezi uzly u a v na nejkratší cestě z u do v , pokud cesta mezi uzly existuje. Matematicky:

$$d(v, u) = \min_{S_{u \rightarrow v}} D(S_{u \rightarrow v}), \quad (3.1)$$

kde $S_{u \rightarrow v}$ je sled z vrcholu u do v .

Closeness centralitu pro daný uzel získáme součtem všech jeho délek nejkratší cesty. Řekněme, že jsme doručovací služba s balíky a máme síť poboček. Chceme zjistit, kde by bylo nejlepší založit centrální depo, odkud by bylo celkově nejlepší dovážet zboží do zbylých poboček. K tomuto úkolu můžeme použít closeness centralitu a určit tak pobočku, z které budeme převážet zboží do ostatních poboček velmi rychle [5].

Matematická definice closeness centrality $C(i)$ je:

$$C(i) = \sum_j d(i, j) \quad (3.2)$$

3. **Betweenness centrality:** třetí a poslední centralitou použitou v naší aplikaci je betweenness centralita. Betweenness centralita řeší důležitost uzlů v síti z toho pohledu, jak často se daný uzel nachází na nejkratší cestě každého páru uzlů v síti. Řekněme, že máme síť, kde si mezi sebou uzly vyměňují zprávy. Předpokládejme, že každý pár uzlů si v síti vyměňuje zprávy se stejnou pravděpodobností za jednotku času a dále předpokládejme, že tyto zprávy cestují vždy po nejkratší cestě mezi touto dvojicí uzlů, anebo náhodnou z nejkratších cest, pokud jich existuje mezi dvojicí vrcholů několik. Důležitost z pohledu betweenness centrality pro každý uzel získáme, když se zeptáme na následující otázku. Pokud počkáme dostatečně dlouhou dobu, aby síť prošel velký počet zpráv, kolik projde průměrně zpráv daným uzlem při cestě do jejich destinace? Dalo by se tedy říct, že betweenness centralita počítá, na kolika nejkratších

cestách daný uzel v v síti leží. Počtu těchto cest se říká *betweenness index*. Například odstranění vrcholů s vysokou hodnotou *betweenness centrality*, by mělo veliký dopad na komunikaci mezi páry uzlů, jelikož by spousta nejkratších cest v síti byla narušena a komunikace by tak musela procházet jinudy. Mohlo by tak dojít ke zvýšení ceny komunikace v síti. Matematicky, nechť n_{st}^v je 1, pokud vrchol v leží na nejkratší cestě z s do t a 0 pokud neleží nebo žádná taková cesta není. *Betweenness centralita* je tedy dána:

$$B_v = \sum_{s \neq v \in V} \sum_{t \neq v \in V} \frac{n_{st}^v}{n_{st}} \quad (3.3)$$

Betweenness vrcholu v vyhodnocuje zlomek nejkratších cest mezi páry vrcholů s a t , které prochází vrcholem v děleno celkovým počtem nejkratších cest mezi s a t [5].

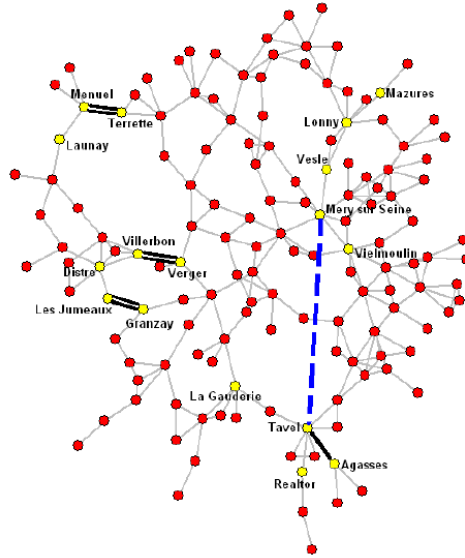
3.3 Síť a jejich využití

Sítě mají nespočet využití ve velkém počtu odvětví. V této sekci se podíváme na různé typy komplexních sítí. Sekci rozdělíme na 4 části, které se budou zabývat nejčastěji studovanými typy sítí, jsou jimi síť technologické, biologické, sociální a informační. Pochopením různých typů složitých sítí, které existují, můžeme získat vhled do základních principů, které řídí chování složitých systémů. Tyto znalosti mají důležité důsledky pro širokou škálu oborů a mají potenciál vést k novým objevům.

3.3.1 Technologické síť

Technologické síť jsou síť fyzické infrastruktury, které tvoří základ moderních technologických společností. Internet je jedním z nejznámějších příkladů takových sítí. Kromě internetu, jehož síťovou reprezentaci jsme mohli vidět na obrázku 3.1, si povíme o jiných důležitých příkladech technologických sítí. Budou jimi energetické a dopravní síť [7].

Energetické síť jsou síť vysokonapěťových přenosových vedení, které přenášejí elektrickou energii na velké vzdálenosti. Uzel v energetické síti reprezentují napájecí stanice nebo rozvodny a jako hrany slouží vedení vysokého napětí, kterým jsou jednotlivé stanice a rozvodny propojeny. Pochopením topologie a chování takovéto síť nám může nabídnout pohled na globální omezení, která řídí jejich tvar a růst. Energetické síť jsou však složité systémy, které jsou více ovlivněny činnostmi operátora a návrhem softwaru, než hrubou topologií síť. V důsledku toho nebyla teorie síť úspěšná ve vysvětlení chování těchto síť. I přesto jsme schopni síťovou analýzou dosáhnout zajímavých výsledků při analýze těchto síť. Příkladem je síť na obrázku 3.3, kde je zkoumána francouzská energetická síť. Výsledkem analýzy bylo zjištění nejzranitelnějších spojů v síti, které jsou na obrázku vyznačeny dvojí čarou a nalezení hrany, která kdyby byla do síť přidána, tak by zvýšila celkovou výkonost této energetické síť (modrá čárkovaná čára) [7].



Obrázek 3.3: Francouzská 400kV energetická síť [8].

Příkladem **dopravních sítí** jsou letecké trasy, silnice a železniční síť. Vytvořit strukturu takových sítí není složité. Letecké síť je možné vytvořit z publikovaných letových řádů leteckých společností, silnice a železniční síť se dají zase zkonstruovat z map. Například silniční síť většinou reprezentujeme uzly jako křižovatky a hrany, kterými jsou spojeny, reprezentují silnice [7].

3.3.2 Biologické síť

Sítě můžeme použít k reprezentování vzorců interakcí mezi biologickými elementy jako jsou chemické reakce v buňkách, spojení mezi mozkovými buňkami a interakce mezi druhy v ekosystémech. Tato část nastiňuje běžné typy biologických sítí [7].

Biochemické síť jsou příkladem biologické sítě, ve kterých jsou biologické molekuly jako jsou enzymy nebo metabolity reprezentovány uzly, které jsou spojeny, když dvě molekuly fyzicky nebo funkčně interagují. Takové modely umožňují provádět komplexní předpovědi o buněčných funkcích, které by nebyly dostupné z jednobuněčných studií [7].

Dalším z odvětví biologických sítí jsou **neurální síť**. Mozek a centrální nervový systém u zvířat se při zpracování informací spoléhá na neurony, specializovaná mozková buňka, která spojuje (obvykle) několik vstupů pro vygenerování jediného výstupu, který pošle dalšímu neuronu nebo neuronům. Typický neuron se skládá z těla buňky, dendritů pro vstupy a axonu pro výstup, přičemž synapse hrají klíčovou roli v regulaci přenosu mezi buňkami. Neurální síť se skládá z více takovýchto buněk propojených dohromady. V neuronové síti mohou být zastoupeny různé typy neuronů a pochopení struktury těchto sítí je zásadní pro vysvětlení vyšších funkcí v mozku zvířete.

Z hlediska sítě lze tedy neuronovou síť reprezentovat jako množinu vrcholů, které jsou reprezentovány neurony, propojené dvěma typy směřových hran.

Mozek je komplexní systém, který se skládá z různých typů těchto neuronů, které mají různé tvary a velikosti. Dokonce i malé oblasti mozku mohou obsahovat četné typy neuronů, které mohou být kódovány v síťové reprezentaci pomocí různých typů vrcholů. Pro vizuální rozlišení těchto typů se pro vrcholy používají různé tvary nebo popisky [7].

3.3.3 Sociální síť

Ze začátku kapitoly jsme se zmínili o tom, že uzel v grafu může reprezentovat člověk a hranu mezi uzly reprezentuje vztah těchto dvou lidí. Takovým to reprezentacím sítě říkáme síť sociální. Uzly ale nemusí být zastoupeny pouze lidmi, ale uzlem může být i skupina lidí.

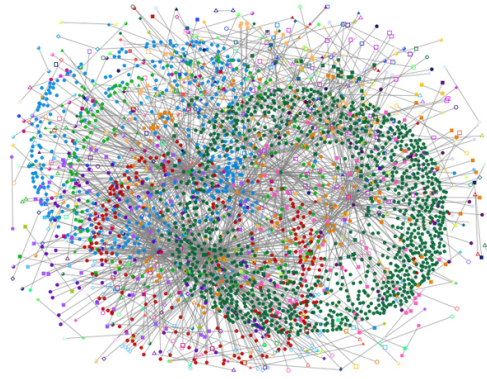
Analýza sociálních sítí není novou oblastí, sociologové byli mezi prvními, kteří kvantitativně studovaly síť. Zakladatel oboru, psychiatr Jacob Moreno, studoval dynamiku sociálních interakcí ve skupinách lidí a provedl možná první skutečné studie sociálních sítí ve 30. letech 20. století. Své diagramy lidských interakcí nazval sociogramy a byly jasně tím, co nyní známe jako síť. Jednou z nejdůležitějších věcí, které je třeba o sociálních sítích pochopit, je, že v takové síti existuje mnoho různých možných definic hrany a konkrétní definici, kterou si člověk vybere bude záležet na tom, na jaké otázky chceme odpovědět. Příklady hran může být přátelství mezi jednotlivci, ale mohly by představovat i profesní vztahy, výměnu zboží popřípadě peněz, romantické nebo sexuální vztahy a mnoho dalších typů propojení. Přímé dotazování pokusných osob a využití archivních záznamů jsou nejčastějšími metodami zjišťování struktury sociálních sítí. Od Morena a dalších byla analýza sociálních sítí aplikována na širokou škálu různých komunit a problémů.

Praktickým využitím tohoto typu sítě může být například zachycení spammerů reprezentovaném na obrázku 3.4 Analyzují se zde přirozené sociální vazby mezi uživateli na webu, které je pro spammery obtížné replikovat. Odhalit spammera není těžké, když se podíváte na jeho nebo její komunikační vzorce. Příkladem takovéto komunikace je to, že klasičtí uživatelé většinou pošlou zprávu jinému uživateli a také od jiného uživatele nějakou zprávu obdrží. Zatímco útočníci většinou pouze posílají škodlivé informace. Ve výsledku bude většina normálních uživatelů propojena do komunit a útočníci budou více izolovaní. Na obrázku 3.4 se spammeři nacházejí na okrajích vizualizované sítě a posílají jednosměrnou komunikaci do centra sítě [7].

3.3.4 Informační síť

Informačních sítí jsou uměle vytvořené sítě skládající se z propojených datových položek. Nejznámějším příkladem takové sítě je World Wide Web, ale patří zde i další sítě, jako jsou citační sítě. Existují také některé sítě s informačními i sociálními aspekty, jako je e-mail, sociální média a online časopisy, které překračují hranice mezi sociálními a informačními sítěmi. [7]

World Wide Web, vytvořený v 80. letech 20. století vědci z laboratoře fyziky vysokých energií CERN v Ženevě, je nejznámější reprezentací informační sítě, ve které jednotlivé webové stránky



Obrázek 3.4: Síť vytvořená pro analyzování k zachycení spammerů [9].

skládající se z textu, obrázků nebo jiných informací tvoří uzly a okraje jsou hypertextové odkazy, které nám umožňují navigaci ze stránky na stránku [7].

Citační sítě jsou sítě tvořené akademickými články, kdy jeden článek cituje jiný. Tyto sítě lze použít ke studiu příbuznosti probíraných témat mezi články. Kvantitativní studie citačních sítí se datují do 60. let 20. století a nyní jsou vytvářeny pomocí ručně spravovaných databází, jako je Science Citation Index, nebo prostřednictvím automatizovaného počítačového indexování citací. Citační sítě jsou podobné World Wide Webu v tom, že mají vrcholy a odkazy, které hrají roli podobnou hypertextovým odkazům. Články s mnoha citacemi jsou často vlivnější a čtenější než články s malým počtem citací [7].

Kapitola 4

Síťová analýza akiových trhů

4.1 Výběr dat pro síťovou analýzu

Tato sekce se zabývá výběrem dat, které budou použity pro generování sítí, jejich následnou analýzu a použití ve webové aplikaci.

Prvotně je potřeba se seznámit s možnými daty, které jsme schopni ohledně obchodovaných akcií získat. Akcie mohou poskytovat různá data, které mohou investoři a analytici využít k informovanému rozhodování o nákupu, prodeji nebo držení akcií. Tady jsou možné údaje, které jsme schopni ohledně akcií získat [1]:

- **Historické ceny:** historické ceny poskytují informaci, jak se pohybovala cena akcií v průběhu času, včetně možných vzorů a trendů. Historická cena se typicky rozděluje na 4 části dat a tyto data určují průběh ceny ve stanovené časové periodě, která může být například 1 den. Jsou to otevírací, nejvyšší, nejnižší a zavírací cena. Často se k těmto údajům přidává ještě údaj o obchodovaném objemu akcií, který nám říká, kolik akcií se v dané periodě prodalo.
- **Tržní kapitalizace:** Tržní kapitalizace je celková hodnota všech akcií společnosti v oběhu. Může poskytnout údaj o celkové velikosti a hodnotě společnosti.
- **Dividendový výnos:** Některé společnosti vyplácejí svým akcionářům dividendy. Dividendový výnos je roční výplata dividendy dělená aktuální cenou akcií, vyjádřená v procentech.
- **Zisk na akcii (EPS):** EPS je čistý příjem společnosti dělený počtem akcií v oběhu. Může poskytnout indikaci ziskovosti společnosti v přepočtu na akcii.
- **Poměr ceny a zisku (P/E):** Poměr P/E je aktuální cena akcií dělená EPS. Může poskytnout údaj o tom, kolik jsou investoři ochotni zaplatit za každý dolar zisku společnosti.

- **Beta:** Beta měří volatilitu akcie vzhledem k celkovému trhu. Hodnota Beta 1 znamená, že akcie je stejně volatilní jako trh, zatímco hodnota Beta vyšší než 1 znamená vyšší volatilitu a hodnota Beta nižší než 1 znamená nižší volatilitu.
- **Zasvěcená obchodní činnost:** Zasvěcená obchodní činnost může poskytnout informace o tom, zda zasvěcenci společnosti jako jsou manažeři nebo členové představenstva kupují, nebo prodávají své akcie ve společnosti.
- **Hodnocení analytiků:** Hodnocení analytiků poskytuje informaci o tom, jak analytici z Wall Street pohlížejí na vyhlídky společnosti.

Celkově lze říci, že analýzou těchto a dalších datových bodů mohou investoři a analytici získat přehled o finančním zdraví společnosti, potenciálu růstu a celkové hodnotě investice.

Je však potřeba z těchto datových bodů vybrat ty, které použijeme pro naši analýzu. Výběr správných akciových dat pro generování síťových grafů zahrnuje několik úvah, jako je časové období dat, frekvence dat, typ dat (např. cena, objem, analytické hodnocení), kvalita dat a jejich přístupnost.

Eliminovat je potřeba ty z datových bodů, které nejsou pro analýzu vhodné. Jelikož chceme analyzovat časové období, jako je například velká recese nebo covidový propad, tak potřebujeme data, která jsou relevantní k těmto časovým obdobím, a zároveň je jich dostatečně velký počet. Tímto eliminujeme datové body jako EPS, P/E, dividendu a tržní kapitalizaci.

Fiskální čtvrtletí	2022	2021	2020
Březen			
Zisk	\$5,887(m)	\$3,445(m)	\$1,786(m)
EPS	\$0.56 (03/26/2022)	\$0.45 (03/27/2021)	\$0.14 (03/28/2020)
Dividendy	N/A	N/A	N/A
P/E	40.65	33.40	105.77

Tabulka 4.1: Příjmy EPS, dividendy a P/E za první fiskální čtvrtletí let 2020 až 2022 firmy Advanced Micro Devices, Inc. (AMD)

V tabulce 4.1 můžeme vidět některé ze zmíněných datových bodů. Jsou to EPS, dividendy a P/E. V tabulce jsou záznamy pro první čtvrtletí let 2022 až 2020. Tyto údaje se počítají čtyřikrát ročně a to vždy pro každé čtvrtletí. Problém u dividend nastává v tom, že ne každá společnost akcionářům dividendy vyplácí, nebo jsou období, kdy dividendy vyplácí a kdy dividendy nevyplácí, a to záleží na rozhodnutí dozorčí rady firmy. Dalším problémem, a to u všech dat v tabulce, je jejich malý počet. Ke každé z mír jsou pouze 4 údaje ročně a to ve chvíli, kdy chceme zkoumat nějaké specifické časové období, které může trvat jenom jeden měsíc v roce, což je málo. Navíc daná matrice, která se počítá pouze každé čtvrtletí, nemusí být ke zkoumanému časovému období vůbec relevantní.

Hodnocení analytiků je zase míra, která není lehce přístupná a nedá se nějak systematicky zařadit. Navíc hodnocením analytiků se většinou myslí dlouhé analytické články, které pouze říkají zda je dobrý nebo špatný čas kupovat danou akcii. My se chceme zaměřit spíše na hodnoty číselné.

Tržní kapitalizace je pouze jedna hodnota, která nám určuje momentální celkovou hodnotu společnosti podle ceny poslední obchodované akcie. Hodnotu jsme však použili pro zúžení počtu společností, které v sítích používáme.

Zbývá nám tedy Beta a historická data cen akcií. Dalo by se říct, že Beta je blízce spojená se změnou ceny akcie. Beta je měřítkem volatility akcie, což znamená, že odráží, jak moc má cena akcie tendenci se pohybovat nahoru nebo dolů ve vztahu k celkovému trhu. Řekněme, že akcii A se změní cena o 10 %, zatímco celkový trh se pohne o 5 %, znamená to, že akcie A je víc volatilní než celkový trh a proto bude mít větší Betu, než například akcie B, které se změnila cena o 4 %. Ačkoliv by se Beta dala spočítat pro každý den, nebyla by to nejlepší míra pro sledování tržního chování. Beta se většinou vypočítává za určité časové období a to v řádech měsíců a roků, čím delší období pro výpočet Bety, tím přesněji změříme citlivost akcie na pohyby trhu.

Pro konstrukci sítí tedy použijeme historické ceny akcií. Velké plus historických cen je to, že jsme schopní s nimi pracovat i pro menší časové období. Historická data také není problém získat, v našem případě pro to použijeme knihovnu `pandas_datareader` v programovacím jazyce python.

4.2 Získání a omezení potřebných dat

Pro analýzu je potřeba si upřesnit, které obchodované akcie do ní zahrneme. V našem případě se zaměříme na analýzu akcií amerických společností obchodovaných na amerických burzách NASDAQ, NYSE a AMEX. Dohromady jde o 5812 akcií, jejichž seznam je možné si zobrazit na NASDAQ Stock Screeneru. Seznam akcií je tedy velmi obsáhlý a obsahuje velký počet akcií, které mají nízkou tržní kapitalizaci a málo se obchodují. Seznam akcií jsme se tedy rozhodli zmenšit na akcie firem, které přesahují tržní kapitalizaci přes 10 miliard \$. Tímto omezením se dostaneme na 539 akcií, což je o dost méně než původní počet. Tyto společnosti už ale jsou z hlediska obchodování a celkového tržního dopadu velmi důležité.

Dále je potřeba si stanovit časové období, pro které budeme potřebovat, aby vybrané akcie měly historická data jejich obchodování. Časové období bylo vybráno dle cíle analýzy, která se zaměřuje na medvědí trhy, které jsou označením pro výrazný pokles cen o více jak 20% od posledního maxima. Jelikož nejdůležitějším obdobím, na které se chceme zaměřit je velká recese, budeme potřebovat data od 1.1. 2007. Součástí tržních propadů bude i pokles cen akcií v období koronaviru a tak jsme se rozhodli sesbírat data až po 18.10.2022. Je tedy potřeba, aby společnosti měly historická data pro období od 1.1.2007 až po 18.10.2022. Toto omezení nás dovede na finální počet společností 410.

Seznam těchto společností je možné získat v souboru csv na již zmíněném NASDAQ Screeneru s tím, že bude obsahovat i společnosti, které nemusí být obchodovány ve stanoveném období. Toto

omezení už se řeší při stahování historických dat, kdy zjistíme, že pro danou společnost historická data pro celé období neexistují.

4.2.1 Python

Veškerá příprava dat od jejich stažení, přípravy a zpracování probíhá v programovacím jazyce Python. V práci budeme používat Python verze 3.11.1. Pro systémy Windows a MacOS je jeho instalační soubor volně ke stažení z oficiálních webových stránek Pythonu [10]. Pro systémy linux, pokud ho nemají nainstalovaný defaultně, je potřeba Python nainstalovat přes terminál, jak je možné vidět na kódu 4.1.

4.2.2 Python pip

Jelikož pro práci s daty budeme potřebovat specifické balíky tříd a funkcí, je potřeba mít nějaký způsob, jak je nainstalovat. S tímto nám pomůže speciální správce balíčků pip. Pomocí pip si můžeme nainstalovat potřebné knihovny a balíky jednoduchým zadáním názvu balíku do terminálu nebo příkazového řádku. Pip pak automaticky stáhne a nainstaluje balík se všemi jeho potřebnými závislostmi. Pip je předinstalovaný s většinou verzí Pythonu. Na systémech linux je potřeba ho občas nainstalovat, jak je možné vidět ve spodní části kódu 4.1

```
# zkontrolujem, zdali na našem systému neexistuje požadovaná verze pythonu
$ python3 --version
# stáhneme informace ohledně balíčků
$ sudo apt-get update
# nainstalujeme požadovanou verzi pythonu 3.11.1
$ sudo apt-get install python3.11.1
# nainstalujeme pip
$ sudo apt-get install python3-pip
```

Listing 4.1: Instalace Pythonu a Python pip na linuxové distribuci ubuntu

4.2.3 Balíky pro získání a zpracování dat

Historická data akcií jsou volně ke stažení na platformě Yahoo Finance. Nejen, že si uživatel může stáhnout historická data pro specifickou akcii přímo z jejich stránek, tak umožňují přístup k datům přes API. Stahování dat přes csv soubory přímo ze stránek by bylo zdlouhavé a tak k tomu využijeme speciální balík v pythonu `pandas_datareader`, který nám umožní stahovat data ze zmíněné Yahoo Finance API. Jelikož s daty budeme chtít nějakým způsobem pracovat, ať už budeme chtít vybrat nějaké specifické časové období ze stažených dat nebo budeme chtít jenom specifické hodnoty, použijeme k tomu speciální python knihovnu. Nainstalování těchto balíčků je možné vidět na kódu 4.2, kde k instalaci používáme správce balíčků pip.

```
# nasinstalování knihovny pandas
$ pip install pandas
# nainstalování balíku pro stažení dat pandas_datareader
$ pip install pandas-datareader
```

Listing 4.2: Instalace potřebného balíku a knihovny správcem balíků pip

4.2.3.1 Knihovna pandas

Pandas je speciální knihovna v pythonu, která se používá pro manipulaci s daty a jejich analýzu. [11] Knihovnu nebudeme používat pouze pro manipulaci s daty, je totiž potřebnou částí pro používání balíku pandas_datareader, který vrací získaná data o akcích ve speciální datové struktuře pandas.DataFrame. Použití knihovny je možné vidět v části kódu 4.3

4.2.3.2 Balík pandas_datareader

Knihovna sloužící pro vzdálený přístup k velkému počtu finančních API jako jsou Yahoo Finance, Never Finance, MOEX, Stooq a další. [12] Knihovna extrahuje data z těchto API a vrátí je ve speciální datové struktuře pandas DataFrame z knihovny pandas. Stahování dat z Yahoo Finance API je možné vidět v kódu 4.3.

```
import pandas as p
import pandas_datareader.data as web
# načtení seznamu akcií jako pandas.DataFrame ze souboru csv
screener_data = p.read_csv("./screener-all.csv", sep=",")
#funkce, která nám umožní stáhnout historická data akcie ve stanoveném časovém
    rozmezí za pomocí pandas_datareader
def download_ticker_data(ticker, start_date, end_date):
    return web.get_data_yahoo(ticker, start=start_date, end=end_date)
#cyklus pro stažení dat všech symbolů akcií z našeho seznamu
data_list = []
for ticker in screener_data.Symbol.tolist():
    try:
        data_list.append(download_ticker_data(ticker, start="2007-01-01", end="
            2022-10-18"))
    except:
        print("Error downloading ticker: " + ticker)
```

Listing 4.3: Příklad použití knihovny pandas a balíku pandas_datareader

4.3 Příprava dat

Nyní máme stažena historická data 410 vybraných akcií pro časové období od 1.1.2007 až 18.10.2022. Pandas_datareader nám data stáhne z Yahoo Finance ve formátu, který je možné vidět v tabulce 4.2. Volně přístupné jsou pouze historická data pro jednotlivé dny, historické ceny například pro jednotlivé hodiny z daných dnů tedy není možné stáhnout, ale v naší práci si bohatě postačíme s denními záznamy. Kromě sloupce Date nás budou zajímat sloupce Open, Close a občas použijeme sloupec Volume. Open značí cenu akcie, za kterou se začala v daný den obchodovat a Close je poslední cena, za kterou se akcie obchodovala pro daný den. Volume nám zase říká, jaký objem akcií se v průběhu dne pro danou společnost obchodoval.

Date	High	Low	Open	Close	Volume	Adj Close
2019-09-10	9.27	8.90	8.91	9.14	62617200.0	9.062220
2019-09-11	9.36	9.06	9.15	9.36	57094900.0	9.280347

Tabulka 4.2: Formát stažených dat z pandas_datareaderu

Sloupce Open a Close ale nejsou ve formátu, v jakém bychom je chtěli použít. Pro konstrukci našich sítí budeme chtít sledovat podobnost ve změnách cen jednotlivých akcií, kde změnu ceny je možné vypočítat jednoduchým vzorečkem, který používá právě Open a Close cenu:

$$Change = \frac{(Close - Open)}{Open} * 100 \quad (4.1)$$

Takto získáme procentuální změnu ceny akcie pro jeden den. Tento vzoreček použijeme pro přepočítání všech historických cen akcií v našem seznamu.

Pro konstrukci sítí budeme používat vektorová data. Budeme používat definici vektorových dat používaných pro strojové učení, kdy jde o množinu numerických hodnot jdoucích po sobě. Tento formát je potřebný, jelikož ho použijeme pro konstrukci sítí pro naši webovou aplikaci a analýzu. Výsledný formát je možné vidět v tabulce 4.3.

Ticker	Sector	2007-10-12	2007-10-15	2007-10-16	2007-10-17	Volume
A	Industrials	-1,842589917	0,615300507	0,712218	-0,70002	7833134
AAP	Consumer Discretionary	-0,539779651	-0,513699488	-1,48869	-2,32423	2513700

Tabulka 4.3: Ukázka transformovaných dat na vektorové

Je potřeba vysvětlit, co některé sloupce v tabulce 4.3 znamenají. Sloupcem Ticker je myšlen symbol, který se používá pro jedinečnou identifikaci obchodovaných akcií konkrétní společnosti. V analýze a webové aplikaci ho budeme používat pro označení vrcholů. Sloupec volume je použit pouze v některých analyzovaných sítích a součástí aplikace je pouze v sekci historických sítí. Je potřeba se

zmínit, že sloupec volume, představující objem, je normalizován pro sektor, z kterého akcie pochází. Objem pro akcii je součtem všech objemů pro obchodované dny zahrnuté v datasetu, který zrovna chceme použít. V tabulce 4.3 je to tedy součet objemů ze 4 obchodovaných dnů. Matematicky si to můžeme vyjádřit sumu objemů pro akcii A od dne a po den b jako

$$V_A(a, b) = \sum_{i=a}^b V_i \quad (4.2)$$

, kde V_i je objem akcie pro den i . Objem podle sektoru normalizujeme tak, že objem podělíme součtem objemů všech akcií v sektoru. Nechť $V_{i,j}$ je objem akcie i v sektoru j . Tak normalizovaný objem $N_{i,j}$ akcie i v sektoru j je dán vztahem:

$$N_{i,j} = \frac{V_{i,j}}{\sum_k V_{k,j}} \quad (4.3)$$

, kde součet je proveden nad objemy akcií k ze sektoru j . Sektor je vidět ve druhém sloupci tabulky a jde o klasifikaci jednotlivých akcií do skupin a ve výsledných grafech je ho možné použít pro obarvení vrcholů.

4.4 Konstrukce sítí

V předešlé kapitole jsme se zmínili, že máme připravená data ve vektorovém formátu. Víme ale, že vektorová data nám nerepresentují data síťové. Je proto potřeba převést tento formát do síťové podoby a vytvořit tak hrany mezi jednotlivými akciemi. K tomu nám poslouží metody ke konstrukci sítí.

Standardní postup pro konstrukci sítí z vektorových dat zahrnuje několik obecných kroků:

- **Určení podobnosti nebo nepodobnosti:** pro data v našem vektorovém formátu je potřeba pro každou dvojici záznamů vypočítat podobnost nebo nepodobnost. K tomuto slouží podobnostní funkce a funkce nepodobnosti. Výsledkem použití těchto funkcí je matice podobnosti nebo nepodobnosti. Z matice podobnosti nebo nepodobnosti jsme již schopni konstruovat graf, který bude reprezentovat síť.
- **Určení struktury sítě:** Podle matice podobnosti nebo nepodobnosti se rozhodneme pro strukturu naší sítě, kde strukturou je myšleno, které vrcholy jsou spolu propojené a jak silně. Hrany s malým ohodnocením by totiž mohly způsobit špatné výsledky z různých síťových algoritmů. K vyhodnocení struktury sítě se používají metody jako *k-nearest neighbors network*, *ε-radius network* nebo *b-matching network*.
- **Sparsifikace sítě:** Pokud je síť stále moc hustá a je problém síť dobře vizualizovat nebo analyzovat kvůli hranám s nízkou váhou, tak hrany s nízkou váhou odstraníme.

4.4.1 Podobnost a nepodobnost

Podobnost a nepodobnost vyjadřují porovnání mezi dvěma objekty. Jinými slovy podobnost a nepodobnost vyjadřuje míru shody nebo divergence mezi dvojicí záznamů. Zachází se s nimi tedy jako s funkcemi, jelikož na výstupu chceme numerickou hodnotu. Jsou podobnostní a nepodobnostní metody, které vypočtou hodnotu jenom mezi numerickými daty, jako máme například my, ale také existují metody, které vyhodnotí podobnost a nepodobnost mezi kategorickými hodnotami.

Po vypočtení podobnosti nebo nepodobnosti pro všechny dvojice záznamů vznikne podobnostní nebo nepodobnostní matice o velikosti $N \times N$, kde N je počet záznamů v datasetu.

Před vypočtením podobnosti nebo nepodobnosti je často potřeba udělat normalizaci dat v datasetu. Je to převážně z důvodu, že tato data se nachází v jiném rozmezí (intervalu). Normalizací tato data převedeme na stejný interval, nejčastěji jsou to intervaly $\langle -1, 1 \rangle$ a $\langle 0, 1 \rangle$ [5].

Podobnost: Nechť X je neprázdná množina, kde je definovaný vztah rovnosti. Pokud s je podobnostní funkce, tak s je ze shora ohraničená a úplná s oborem hodnot a definičním oborem definovanými jako:

$$s : X \times X \mapsto I_s \subset \mathbb{R} \quad (4.4)$$

kde I_s je ze shora ohraničená a $\max_{\mathbb{R}} I_s = s_{max}$. Dále také podobnostní funkce s splňuje následující vlastnosti:

1. Reflexivita: $s(x, x) = s_{max}$
2. Silná reflexivita: $s(x, y) = s_{max} \iff x = y$
3. Symetrie: $s(x, y) = s(y, x)$
4. Ohraničení: s je zespolu ohraničená když $\exists a \in \mathbb{R} : s(x, y) \geq a, \forall x, y \in X$.
5. Uzavřenost: tato vlastnost zajišťuje existenci ohraničení zespolu. $x, y \in X : s(x, y) = s_{min}$
6. Tranzitivita: pokud τ_s je operátor tranzitivity, tak platí: $s(x, y) \geq \tau_s(s(x, z), s(z, y)), \forall x, y, z \in X$.

Můžeme vidět, že s vezme na vstupu dva záznamy z X a vrátí reálné číslo. Čím víc si jsou dva prvky podobné, tím větší bude mezi nimi výsledná hodnota podobnosti [5].

Nepodobnost: Nechť X je neprázdná množina, kde je definovaný vztah rovnosti. Pokud d je nepodobnostní funkce, tak d je zespolu ohraničená a úplná s oborem hodnot a definičním oborem definovanými jako:

$$s : X \times X \mapsto I_d \subset \mathbb{R} \quad (4.5)$$

kde I_d je zespolu ohraničená, jelikož jsme předpokládali, že d je zespolu ohraničená a $\max_{\mathbb{R}} I_d = s_{min}$. Dále také podobnostní funkce d splňuje následující vlastnosti:

1. Reflexivita: $d(x, x) = d_{min}$
2. Silná reflexivita: $d(x, y) = s_{min} \iff x = y$
3. Symetrie: $d(x, y) = d(y, x)$
4. Ohraničení: d je ze shora ohraničená když $\exists a \in \mathbb{R} : d(x, y) \leq a, \forall x, y \in X$.
5. Uzavřenost: tato vlastnost zajišťuje existenci ohraničení zespodu. $x, y \in X : d(x, y) = d_{max}$
6. Tranzitivita: pokud τ_d je operátor tranzitivity, tak platí: $d(x, y) \leq \tau_d(d(x, z), d(z, y)), \forall x, y, z \in X$.

Můžeme vidět, že d vezme na vstupu dva prvky dat z X a vrátí reálné číslo. Čím víc si jsou dva prvky nepodobné, tím větší mezi nimi bude výsledná hodnota nepodobnosti [5].

4.4.2 Příklady podobnostních funkcí

Pro konstrukci sítí v naší aplikaci a pro část naší analýzy byly použity některé podobnostní funkce.

- **Gaussian Kernel Similarity:** podobnost Gaussian Kernel je dána funkcí:

$$S_{Gaussian}(x_i, x_j) \triangleq a \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \quad (4.6)$$

kde $\sigma > 0$ je rozptyl a a je škálovací konstanta. Výraz $\|x_i - x_j\|^2$ je Eukleidovskou normu mezi x_i a x_j .

- **Cosine Similarity:** cosinová podobnost je dána funkcí:

$$S_{Cosine}(x_i, x_j) \triangleq \frac{\sum_{k=1}^P x_i(k)x_j(k)}{\|x_i\|\|x_j\|} = \frac{\langle x_i(k)x_j(k) \rangle}{\|x_i\|\|x_j\|}, \quad (4.7)$$

kde $\langle \cdot, \cdot \rangle$ je skalární součin a $\|\cdot\|$ je Eukleidovskou norma. [5]

4.5 LRnet algoritmus pro konstrukci sítí

My ke konstrukci sítí použijeme specifický algoritmus LRnet. Metoda LRnet používá míru nejbližšího souseda trochu jinak než ve standardních algoritmech pro konstrukci sítí. Nejbližšího souseda můžeme chápat jako datový objekt, který je k danému datovému objektu nejbližší ve smyslu podobnosti nebo nepodobnosti. V případě LRnetu se předpokládá, že datové objekty mají rozdílnou reprezentativnost. Reprezentativnost je lokální vlastnost postavená na počtu datových objektů, které jsou právě nejbližším sousedem vybraného objektu. Tato reprezentativnost se v LRnet používá ke konstrukci grafu tak, že se vytvoří hrany mezi dvojicemi všech nejbližších sousedů a vytvoří se

také další hrany mezi jednotlivými datovými objekty v úměrném počtu k reprezentativnosti těchto objektů. Reprezentativnost uzlů v sestrojeném grafu pak přibližně odpovídá reprezentativnosti datových objektů. Pro popsání tohoto algoritmu je potřeba definovat si pár pojmů a předpokladů.

Předpokládáme, že $D = (O, s)$ je dataset, kde O je množina objektů a $s : O \times O \rightarrow \mathbb{R}_{\geq 0}$ je podobnostní funkce, která kvantifikuje podobnost mezi dvěma objekty. Sousedy jsou objekty O_j , pro které platí $s(O_i, O_j) > 0$. Sousedství objektu O_i je množina všech jeho sousedů. Nejbližší soused objektu O_i , je objekt O_j , který má největší podobnost s objektem O_i . Když existuje více takových objektů, tak objekt O_i má více nejbližších sousedů [13].

Pro výpočet reprezentativnosti je potřeba zadefinovat si lokální stupeň a důležitost.

Lokální stupeň: Nechť $d(O_i)$ je počet objektů, pro které je objekt O_i sousedem. Tak $d(O_i)$ je lokální stupeň objektu O_i

Lokální důležitost: Počet objektů, pro které je O_i nejbližším sousedem je $g(O_i)$ a $g(O_i)$ je tedy lokální důležitost objektu O_i .

x-reprezentativnost: Pokud máme $x \in \mathbb{R}_{>1}$ a lokální stupeň $d(O_i) > 0$, tak x-reprezentativnost objektu O_i je definována jako:

$$r_x(O_i) = \frac{g(O_i)}{\log_x(1 + d(O_i))} \quad (4.8)$$

základ x-reprezentativnosti: základ x-reprezentativnosti je definovaný jako:

$$b(O_i) = (1 + d(O_i))^{\frac{1}{g(O_i)}}, \quad (4.9)$$

čím nižší je báze x-reprezentativnosti, tím větší má objekt lokální význam. Z výše uvedených vztahů základu x-reprezentativnosti jsme schopní získat lokální reprezentativnost.

Lokální reprezentativnost:

$$lr(O_i) = \frac{1}{b(O_i)} \text{ pro lokálně důležité } O_i, \quad (4.10)$$

$$lr(O_i) = 0 \text{ v ostatních případech.} \quad (4.11)$$

Reprezentativní soused: je definován jako:

$$k = \text{ROUND}(lr(O_i) * d(O_i)) \quad (4.12)$$

a $K(O_i)$ je tedy množina sousedů objektu O_i [13].

4.5.1 Kroky LRnet algoritmu

1. Vytvoří se podobnostní matice S pro dataset D .
2. Vypočte se reprezentativnost všech objektů O_i .

3. Vytvoří se množina uzlů V grafu G tak, že uzel v_i reprezentuje objekt O_i z datasetu D .
4. Vytvoří se množina hran E obsahující hrany e_{ij} mezi párem vrcholů v_i a v_j , pokud je O_j nejbližší nebo reprezentativní soused objektu O_i [13].

Na adrese https://homel.vsb.cz/~kud007/lrnet_files/ je možné stáhnout aplikaci implementující algoritmus LRnet [14].

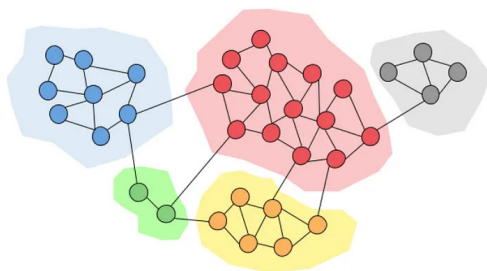
4.6 Komunity v síti

Komunity se používají v sítích pro detekování skupin uzlů, které jsou mezi sebou hustěji propojeny než s uzly mimo skupinu. Tyto skupiny uzlů, také známé jako shluky nebo moduly, lze identifikovat analýzou vzorů spojení mezi uzly v síti.

Komunity mohou být různě interpretovány vzhledem ke kontextu sítě. V sociálních sítích mohou komunity představovat skupiny lidí, kteří sdílejí společné zájmy, národnost a jiné. V biologických sítích mohou komunity představovat skupiny genů nebo proteinů, které spolupracují při provádění specifické biologické funkce.

Identifikace komunit v sítích může poskytnout pohled na strukturu a funkci komplexních systémů. Může nám například pomoci pochopit, jak se šíří informace v rámci sociální sítě nebo jak různé geny nebo proteiny spolupracují v rámci biologického systému.

Algoritmus LRnet používá metodu Louvain pro detekci komunit. Ve webové aplikaci pak používáme detekované komunity metodu Louvain pro obarvení vrcholů podle komunity, do které patří. Síť, kde jsou nalezené komunity, je možné vidět na obrázku 4.1. Je zde nalezených 5 komunit a každá z nich je obarvená svojí barvou. [15]



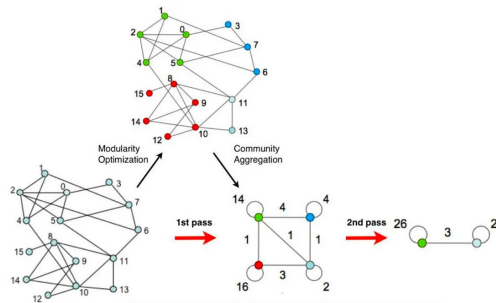
Obrázek 4.1: Příklad detekovaných komunit v síti [16]

4.6.1 Louvain detekce komunit

Algoritmus Louvain pro detekci komunit je efektivní metoda pro detekování komunit ve velkých sítích založená na optimalizaci modularity. Algoritmus se skládá z dvou fází: optimalizace modularity a agregace komunit. V první fázi algoritmus přiřadí uzlům komunity a snaží se zjistit, jestli není lepší

dát jisté uzly do jejich sousedních komunit. První fáze algoritmu přestane ve chvíli, když se v síti najde lokální maximum modularity. V druhé fázi se vytvoří nová síť, kde nalezené komunity v první fázi algoritmu reprezentují uzly v této síti. První fázi algoritmu aplikujeme na nově vytvořenou síť a proces opakujeme až do chvíle, kdy se v síti neprovedou žádné změny a dosáhne se maximální modularity.

Tento iterativní proces je možné vidět na obrázku 4.2, kdy v prvním kroku algoritmus přiřadí uzlům jejich komunity. Pro tyto vytvořené komunity provede druhý krok, kterým je agregace komunit. Vznikne tak graf o 4 uzlech, nad kterým se provede proces znovu a zjistí se, že síť je možné optimalizovat na 2 komunity [16].



Obrázek 4.2: Postup kroků algoritmu Louvain [17]

Kapitola 5

Postup a výsledky síťové analýzy

Cílem této práce je analyzovat komunity na akciových trzích prostřednictvím síťové analýzy. V předchozí sekci jsme definovali data, která jsme použili pro naši analýzu, a nyní se budeme soustředit na to, zda jsou komunity zkonstruovaných grafů v souladu se sektory akcií, které reprezentují vrcholy v této síti. Tento přístup nám umožní lépe pochopit strukturu a vzájemné vztahy mezi různými sektory na trhu a může být užitečný pro investory a další subjekty, kteří se zajímají o finanční trhy.

Analyzovali jsme komunity na akciových trzích z období velké recese, pandemie COVID-19 a medvědích trhů v srpnu 2011. Tato období byla vybrána, protože představují klíčové momenty v historii finančních trhů a byla doprovázena významnými změnami v cenách akcií a volatilitě trhů. Síťovou analýzu jsme provedli s cílem identifikovat klíčové komunity a sektory, které se výrazně liší v chování během těchto období.

5.1 Definice problému

Síťová vizualizace: detekce komunit: Vrcholy ve vybraných sítích obarvíme podle sektorů, do kterých patří. Účelem je zobrazení toho, jestli síť konstruované algoritmem LRnet s použitím vybraných podobnostních funkcí jsou schopné zobrazit vztah v sektorech z fyzického světa jako komunitu v pozorovaných sítích. Dále nad sítěmi použijeme algoritmus Louvain pro detekci komunit a porovnáme, jestli jsou vytvořené komunity tímto algoritmem shodné nebo podobné komunitám sektorů v síti.

Síťová vizualizace: průběh tržní krize: Na zkoumané síťové data z vybraných kritických období budeme aplikovat 5 různých obarvení pro vrcholy, které budou představovat procentuální posun v ceně za specifický úsek ve vybraném období. Počet úseků určíme podle délky zkoumaného období, čím delší bude zkoumané období, tím ho rozdělíme na více úseků. Cílem bude vizualizovat průběh vybraných období a sledovat, v kterých sektorech začaly poklesy cen, a do kterých sektorů se to šířilo, ale i které sektory nebo akcie nebyly tržní krizí ve zkoumaném období vůbec postíženy.

5.2 Zkoumané období

Budeme analyzovat síťová data vytvořená pro 3 časové období, kdy akciové trhy zažily výrazné cenové poklesy. Jsou to velká recese, medvědí trh v srpnu 2011, a tržní propad během pandemie COVID-19 v roce 2020. Pro velkou recesi je vybrané časové rozmezí od 11.10.2007 až 4.6.2009. Medvědí trhy v srpnu 2011 mají rozsah zkoumaného období od 1.8.2011 až 14.10.2011 a pro tržní propad během pandemie COVID-19 máme stanovený rozsah od 12.2.2020 až 8.6.2020.

Nejvíce se zaměříme na velkou recesi, zejména jelikož je to období, které mělo z vybraných období největší vliv na ekonomiku po celém světě. Podíváme se, jestli banky, finanční instituce a realitní firmy tvoří společné komunity v síti.

Velká recese začala jako finanční krize, která se postupně rozšířila do celé ekonomiky. Začátkem krize byla bublina na trhu s nemovitostmi v USA, která vedla k růstu hypotečního trhu, spekulativnímu chování a přeúvěřování. Během velké recese v letech 2007-2009 si některé sektory na trhu vedly hůře než jiné. Nejvíce postiženými sektory byly:

1. Finanční sektor: Finanční krize byla jednou z hlavních příčin velké recese a finanční sektor byl během tohoto období nejvíce postižený. Společnosti v oblasti bankovníctví, pojišťovnictví a investičních fondů se potýkaly s vysokými ztrátami a snižováním hodnoty svých aktiv, což vedlo k celkové nestabilitě na trhu.
2. Realitní sektor: Snížila se aktivita na trhu s nemovitostmi, což mělo dopad na realitní agentury, developery a další společnosti působící v oblasti nemovitostí. Tyto společnosti musely čelit vysokým ztrátám a snižování hodnoty svých aktiv.

Medvědí trh v srpnu 2011 začal po zveřejnění zpráv o zhoršující se hospodářské situaci v USA a v Evropě. Konkrétně, během léta roku 2011 se objevily obavy o dluhovou krizi v Evropě, zejména v Řecku, což vyvolalo nervozitu na finančních trzích. Současně se v USA snížila důvěra spotřebitelů a firem, což vedlo k poklesu akciových trhů. Nejvíce postiženými sektory v tomto období byly:

1. Finanční sektor: Finanční společnosti, včetně bank, pojišťoven a investičních společností, byly velmi citlivé na vývoj ekonomiky a hospodářské stability. Během medvědího trhu v srpnu 2011 došlo k poklesu akcií těchto společností kvůli obavám z recese a poklesu zisků.
2. Energetický sektor: Společnosti v oblasti těžby, výroby a distribuce energie byly také zasaženy poklesem cen komodit, zejména ropy. Tyto společnosti závisí na cenách ropy a dalších surovin, což může mít vliv na jejich ziskovost.
3. Průmyslový sektor: Společnosti v průmyslovém sektoru, zejména v oblasti výroby strojů a zařízení, byly citlivé na pokles poptávky po jejich výrobcích, který souvisel s poklesem ekonomické aktivity a obavami z recese.

Pandemie COVID-19 v roce 2020 měla obrovský dopad na celosvětovou ekonomiku, a to včetně trhů s akciemi a komoditami. V období únor až březen 2020 došlo k výraznému propadu téměř všech hlavních tržních indexů a sektorů, přičemž se jednalo o jeden z největších propadů za posledních několik desetiletí. Velmi zajímavé bylo velmi rychlé zotavení trhu, kdy se ceny většiny akcií v časovém rozmezí od 24.3.2020 do 8.6.2020 zotavily na únorové ceny. Nejvíce postiženými sektory byly:

1. Cestovní ruch a letecký průmysl: Kvůli pandemii bylo cestování značně omezeno a mnoho leteckých společností muselo zrušit nebo omezit lety, což vedlo k významnému propadu trhu.
2. Energetický sektor: Kvůli poklesu poptávky po ropě a zemním plynu během pandemie došlo k výraznému poklesu cen energií, což mělo negativní dopad na energetický sektor.

5.3 Shrnutí dat

Přehled použitých akcií a formátu dat pro konstrukci sítí je zmíněný v předešlé kapitole i s přesným formátem v tabulce 4.3. Formáty budeme používat dva, jeden bude obsahovat poslední sloupec vektorových dat s normalizovaným objemem a druhý formát bude bez objemu. Vybrané akcie mají každá svůj sektor pod který spadají a podle něj budeme vrcholy obarvovat. Je potřeba zmínit, že informace o sektorech vybraných akciových společností máme podle americké burzy NASDAQ. Celkem máme 13 označení pro sektory pod které společnosti spadají, kde jedno označení zahrnuje sektory, které podle burzy NASDAQ pod žádný sektor nespádají. Postižený sektor cestovní ruch a letecký průmysl zmíněný v předešlé sekci spadá podle označení burzou NASDAQ do jiného označení, než jejich název napovídá a spadá tak do sektoru Consumer Discretionary (spotřebitelská volba).

Pro výsledné sítě byly použity dvě podobnostní funkce Gaussian Kernel a kosínová podobnost. Výsledky se porovnají a k prezentování výsledků bude použita funkce, se kterou jsme dosáhli lepších výsledků.

5.4 Výsledky

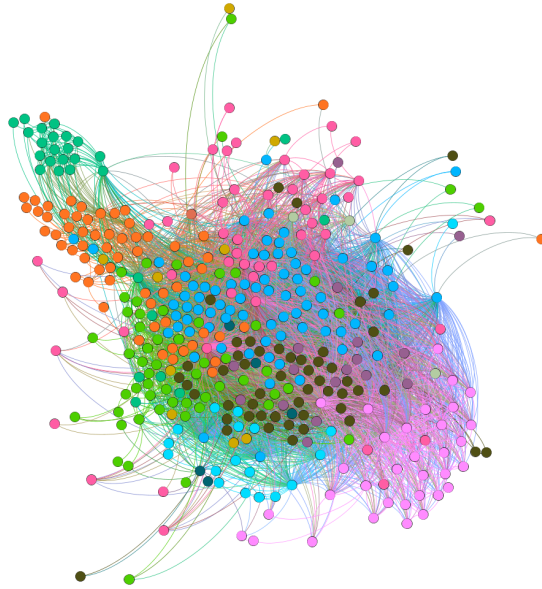
V této sekci se podíváme na výsledky našeho definovaného problému, podíváme se, která metoda a formát byla pro jeho cíl přesnější.

5.4.1 Síťová vizualizace: detekce komunit

Detekci komunit jsme aplikovali pro všechna 3 zmíněná období, kde největší důraz klademe na velkou recesi, jelikož šlo o období s největším historickým významem z vybraných období, ale taky bylo jeho trvání delší než zbylé dvě vybraná období. Zbylá dvě období byla vybrána nejen protože to jsou zajímavé období tržních propadů, ale také kvůli jejich délce, aby jsme byli schopni porovnat, jestli se nám podaří dobře detekovat komunity sektorů i pro sítě konstruované nad menším

časovým obdobím. Ke konstrukci sítí jsme použili dva formáty vektorových dat, kdy jeden formát obsahuje normalizovaný objem jak v tabulce 4.3. Druhý formát je stejný, akorát neobsahuje sloupec s normalizovaným objemem. Předpokládáme, že formát s normalizovaným objemem podle sektoru nám detekuje komunity jako sektory lépe.

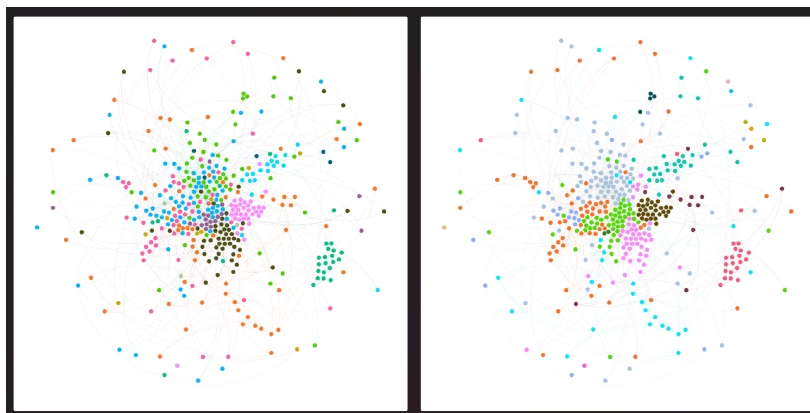
Pozorováním rozdílů mezi sítěmi konstruovanými z vektorového formátu pro období velké recese, obsahujícího normalizovaný objem a sítěmi konstruovanými bez objemu, jsme vypořizovali, že sítě s objemem byly patrně lépe rozdělené podle sektorů. Šlo tak lépe v síti pozorovat jednotlivé rozdělení na sektory, Žádný ze sektorů ale nebyl jednoznačně ohraničený, resp. separovaný a většina sektorů měla společnosti rozesté na různých místech v síti. Nejlépe se povedlo zachytit komunity realitního, finančního a sektoru veřejných služeb na síti konstruované s kosínovou podobností. Tuto síť vidíme na obrázku 5.1, kde v levém horním rohu můžeme najít právě finanční sektor označený oranžovou barvou a realitní sektor označený světle zelenou barvou. Vpravo dole se nachází zmíněný sektor veřejných služeb vyznačený růžovou barvou.



Obrázek 5.1: Síť konstruována s kosínovou podobností a obarvená podle sektorů

Rozdíly mezi použitím podobnostní funkce Gaussian Kernel a kosínové podobnosti byly patrné. Sítě konstruované s použitím Gaussian Kernelu obsahovaly značně menší počet hran než konstruované sítě s použitím kosínové podobnosti. Sítě konstruované s kosínovou podobností tak byly velmi husté, naopak sítě s Gaussianem byly velmi řídké a nebyly v nich zřejmé vztahy mezi sektory. Co bylo na druhou stranu výhodou řídké sítě, bylo porovnání obarvené sítě podle sektorů se sítí obarvenou nalezenými komunitami algoritmem Louvain, kdy nalezené komunity algoritmem Louvain přibližně shodovaly na nalezené komunity sektorů v síti. Toto porovnání je možné vidět na obrázku

5.2, kde realitní, zdravotní, energetický a sektor veřejných služeb na levé síti, se přibližně rovnají nalezeným komunitám na pravé síti.



Obrázek 5.2: Porovnání sítí obarvených podle sektorů a nalezených komunit

Pro období pandemie COVID-19 a medvědího trhu v srpnu roku 2011 jsme očekávali, že jednotlivé sektory nebudou tvořit jednoznačné komunity, kvůli délce vybraných období. Tento problém jsme nejvíc očekávali u medvědího trhu 2011, pro nějž byla zkoumána data od 1.8.2011 až po 14.10.2011 a je tak z vybraných období nejkratší. Výsledek ale nebyl podle očekávání a komunity sektorů jsou vidět při použití obou podobnostních funkcí a dostali jsme dobré výsledky, i při pozorování sítí konstruovaných bez sloupce s objemem. Dokonce nebyly patrné ani rozdíly v hustotě sítě konstruované s jinou podobnostní funkcí. Sítě obarvené dle sektorů se ani v jednom případě nepodobají sítím obarvených dle nalezených komunit. Na obrázku 5.3 je možné vidět srovnání výsledných sítí konstruovaných s kosínovou podobností vlevo a Gaussianem vpravo. Obě dvě sítě jsou konstruovány z formátu s objemem. Na obrázku u obou sítí je možné vypořozovat velmi viditelné komunity realitního, finančního, průmyslového a sektoru veřejných služeb.



Obrázek 5.3: Srovnání sítí konstruovaných s kosínovou podobností a Gaussian Kernelem pro období medvědího trhu 2011

Nejméně byly patrné sektory u sítí konstruovaných pro období tržního propadu při pandemii COVID-19, kde nejvýraznější komunity šly vyzorovat na síti z formátu s objemem konstruované s podobnostní funkcí Gaussian Kernel.

Výsledky bychom shrnuli velmi pozitivně, u většiny sítí se podařilo detekovat některé sektory, kde toho mají společnosti do nich patřící hodně společného. Převážně to byly finanční, realitní a sektor veřejných služeb. Například sektor spotřebitelské volby netvořil skoro v žádné ze zkoumaných sítí pořádné uskupení. Důvodem může být to, že spousta společností spadající pod tento sektor je ve fyzickém světě velmi odlišná ve svých oborech. Nejlepší výsledky jsme dosahovali s kosínovou podobností, ale po použití algoritmu pro detekci komunit Louvain se vytvořené komunity se sektory neshodovaly a tvořily je spíš skupiny sektorů. Tady se naopak vedlo lépe při použití funkce Gaussian Kernel, kdy se vytvořené komunity docela shodovaly se sektory.

5.4.2 Síťová vizualizace: průběh tržní krize

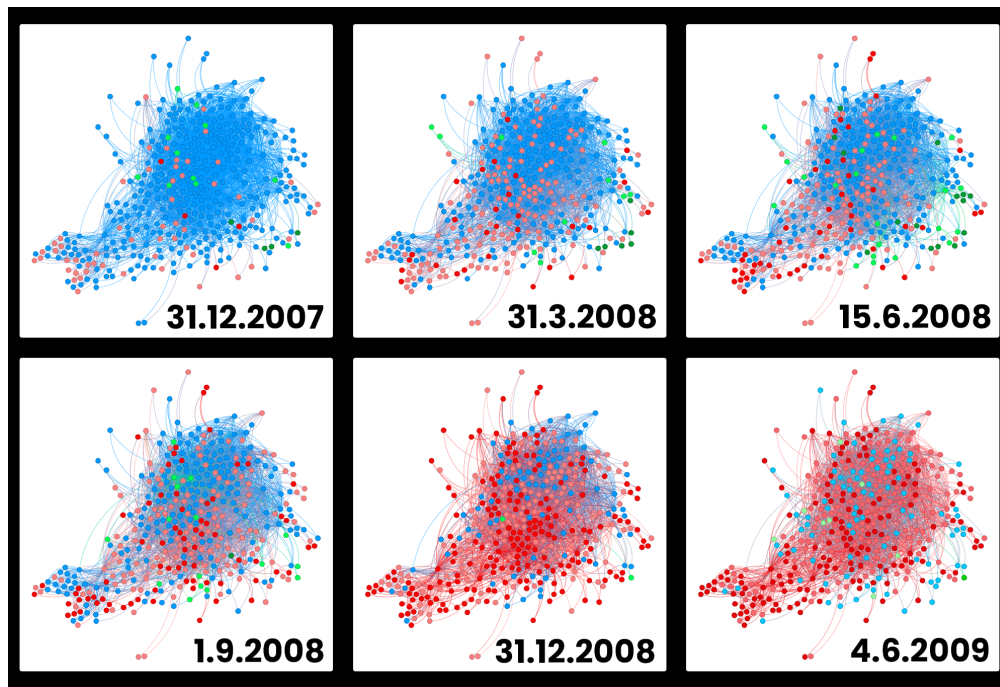
Vizualizovali jsme průběh cen akcií pro období velké recese a pandemie COVID-19. Podíváme se, jestli se nejvíce postižené sektory v našich sítích shodují s postiženými sektory, které jsou zmíněny v sekci o zkoumaných obdobích 5.2. Pro obarvení vrcholů bylo použito 5 barev, které jsou popsány v tabulce 5.1. Vizualizaci provádíme nad sítěmi konstruovanými algoritmem LRnet a podobnostní funkce, kterou jsme při tom použili byla kosínová podobnost.

Procentuální změna	Obarvení vrcholů	HEX barevné označení
méně než -40%	Tmavě červená	#ff0000
-40% až -20%	světle červená	#ff8080
-20% až 20%	světle modrá	#0099ff
20% až 40%	světle zelená	#00ff55
více než 40%	tmavě zelená	#009933

Tabulka 5.1: Obarvení vrcholů podle změny v ceně akcií

Vizualizace průběhu velké recese: Pro vizualizaci průběhu recese bylo použito 6 datových úseků, které je možné vidět na obrázku 5.4. Všechny úseky začínají dne 11.10.2007. Při vizualizaci poklesů cen do 31.12.2007, bylo pozorováno, že ve finančním, realitním a sektoru spotřebitelská volba byly společnosti, které dosahovaly ztrát mezi -20% až -40%. Tato ztrátovost se do 31.3 rozšířila i na technologický sektor a velmi se rozrostla v sektoru spotřebitelské volby, což bylo nejspíše následkem snížené kupní síly spotřebitelů z důsledku probíhající úvěrové krize. Pokračujeme do poloviny roku 2008, kde společnosti ve finančním sektoru, jako například AIG, což je pojišťovací společnost, která byla v polovině roku 2008 na pokraji krachu a musela být zachráněna vládou USA, začaly dosahovat cenových ztrát akcie pod -40%. Ke konci roku 2008 se cenový propad rozšířil do všech sektorů a zůstalo to tak až do konce našeho vybraného období. Nejméně postiženým sektorem v průběhu celého období bylo zdravotnictví. Celkově tímto pozorováním můžeme usoudit, že vytvořená síť

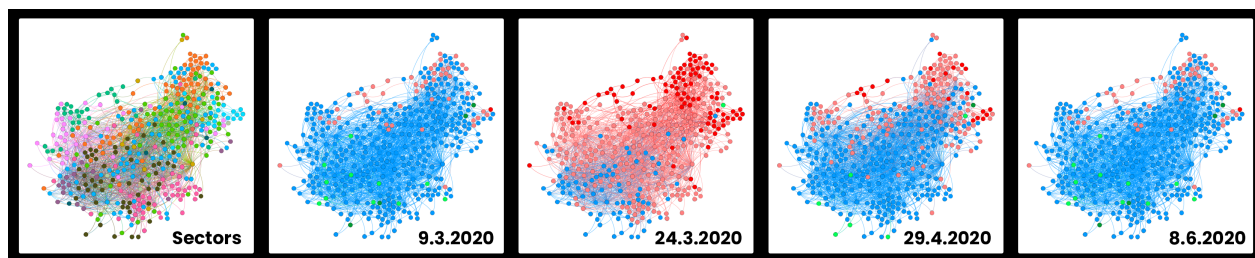
vizualizovala průběh vývoje ceny kritických sektorů a rozšíření do celé ekonomiky podle předpokladů s malou výjimkou. Touto výjimkou je průběh propadů ceny v realitním sektoru, kdy si sice podle očekávání spolu s finančním sektorem na začátku období vedly špatně, ale jejich propad se skoro nezměnil až do konce roku 2008.



Obrázek 5.4: Vývoj ceny v průběhu velké recese

Vizualizace pandemie COVID-19: Pro vizualizaci propadů cen akcií během pandemie COVID-19 byly zvoleny 4 úseky. Zkoumané období tržního propadu pandemie COVID-19 zahrne pozorování nejen propadu cen, ale také jeho zotavení. Podíváme se také, jestli jsou některé sektory, kterým se během zkoumaného období nepodařilo zotavit. První časový úsek je od 12.2.2020 až do 9.3.2020. Zde pozorujeme, že energetický, finanční a industriální sektor jsou nejvíce postiženou oblastí a majorita společností v energetickém sektoru dosahuje ztrát větších než 40%. V dalším pozorovaném úseku do 24.3.2020 pozorujeme, že se cenový propad šíří i do zbylých sektorů a většina společností ve finančním sektoru čítá ztráty přes 40%. Spotřební sektor má naopak většinu svých zástupců v rozmezí od -20% do 20%, důvodem může být vyšší poptávka po některém spotřebním zboží. V následujícím úseku do data 29.4.2020 došlo k zotavení většiny postižených sektorů do stavu při prvním pozorování. Stále zde pozorujeme postižený energetický, finanční a industriální sektor. Zajímavé pozorování je u realitních společností, kterým se nepodařilo zotavit a ceny akcií zůstávají na podobné úrovni jako při druhém úseku. Ve finálním období je většina společností zotavena do únorových cen a jsou zde i společnosti, jejichž akcie dosahují cenového růstu vyššího než 20%. Jediný viditelně postižený sektor je finanční a také je postižených pár společností z energetického sektoru, kterým se nepodařilo zotavit. Zmíněné sektory letectví a cestovního ruchu patří taktéž mezi postižené sektory,

ale jelikož spadají podle označení sektorů NASDAQ pod větší množinu sektorů, cestovní ruch do spotřebitelské volby a letecké společnosti pod průmysl, tak jsme nebyly schopni pozorovat jejich změny z pohledu sektorů, ale museli jsme chování jejich ceny pozorovat jednotlivě.



Obrázek 5.5: Vizualizace průběhu tržního porpadu pandemie COVID-19

Kapitola 6

Webová aplikace

V této kapitole se podíváme na technologie použité k vývoji webové aplikace a důvody k jejich volbě. Shrneme si, proč jsme se rozhodli pro webovou aplikaci a popíšeme si jednotlivé funkcionality naší aplikace.

6.1 Proč webová aplikace

V této části si shrneme, proč je poslední dobou lepší vyvíjet aplikace ve webovém prostředí a podíváme se na hlavní důvody, proč jsme se rozhodli pro webovou aplikaci z pohledu sítí. Webové aplikace jsou stále populárnější pro vývoj softwaru z několika důvodů:

1. **Přístupnost:** K webovým aplikacím lze přistupovat z jakéhokoli zařízení, které má připojení k internetu a webový prohlížeč.
2. **Kompatibilita napříč platformami:** Vzhledem k tomu, že webové aplikace běží ve webovém prohlížeči, lze je spouštět na jakémkoli operačním systému, ať už jde o Windows, Mac, Linux nebo mobilní platformy jako Android a iOS. Není tak potřeba vyvíjet aplikaci pro více platforem.
3. **Snadné nasazení:** S webovými aplikacemi není potřeba do zařízení uživatele instalovat žádný software. Díky tomu je nasazení mnohem jednodušší, protože aplikaci lze aktualizovat na straně serveru bez potřeby jakékoli interakce uživatele.

Hlavními body pro naše rozhodnutí jsou přístupnost a kompatibilita napříč platformami. Přístupnost umožňuje přístup do aplikace odkudkoliv, stačí aby používané zařízení, obsahovalo prohlížeč a mělo přístup k internetu. Kompatibilita na druhou stranu ulehčuje vývoj, snižuje potřebné náklady a není potřeba řešit specifické platformní problémy.

Dalším důvodem, který pomohl při rozhodování je obrovský počet knihoven, které pomáhají a usnadňují vizualizaci síťových dat ve webovém prostředí.

Potřebujeme si určit, co všechno potřebujeme pro to, aby nám běžela webová aplikace, která splňuje naše požadavky. Pro běh naší webové aplikace budeme potřebovat tyto části:

1. **Webový server:** Webový server je zodpovědný za přijímání a odpovídání na požadavky ze strany webového prohlížeče klienta. Tato část je potřebná pro odesílání dynamických dat uživateli naší aplikace.
2. **Programovací jazyk na straně serveru:** K vývoji backendu webové aplikace je potřeba programovací jazyk, jako je PHP, Python, Ruby nebo JavaScript.
3. **Databáze:** K ukládání a získávání dat používaných webovou aplikací je potřebná databáze.
4. **Webový framework:** Webový framework poskytuje strukturu pro vývoj webových aplikací.
5. **Knihovny a moduly:** Pro specifickou funkcionalitu je potřeba použít knihovny nebo balíky, které nám potřebnou funkcionalitu přidají. Příkladem je knihovna pro vizualizaci síťových dat.
6. **Základní bloky webových stránek:** Webové stránky mají 3 základní bloky, kterými jsou HTML, CSS a JavaScript. HTML se používá pro strukturování obsahu, CSS se používá pro stylování obsahu a JavaScript se používá pro přidání interaktivity a funkčnosti webové stránky.

6.2 Základní bloky webových stránek

V předešlé části jsme se zmínili o 3 základních blocích, které tvoří webovou stránku. Jsou to HTML, CSS a JavaScript. Ačkoliv jsou všechny 3 technologie spolu úzce spojené, každá z nich na webové stránce řeší úplně jiný problém. Všechny 3 části pak zkombinuje webový prohlížeč a rozhodne se, jak nám webovou stránku zobrazí.

6.2.1 HTML

HTML neboli HyperTextLanguage je standardní značkovací jazyk. Používá se k popisování toho, jakým způsobem chceme, aby se zobrazil obsah webové stránky. K popisu obsahu se používají takzvané tagy, které říkají prohlížeči, zda se jedná o paragraf, obrázek, video nebo nadpis, ale taky jestli chceme, aby daný text byl kurzívou nebo silně vyznačený. Jednou z důležitých funkcí HTML je možnost vkládat CSS a JavaScript soubory pro rozšíření funkcionality webové stránky. Příklad HTML je možné vidět na kódu 6.1, kde je definovaná základní struktura HTML dokumentu a jsou zde použity tagy pro nadpis a blok textu [18].

Je potřeba se zmínit ohledně HTML DOM (Document Object Model). Je to objektový model samotného HTML a definuje tagy v našem HTML jako objekty. V těchto objektech pak uchovává informace o vlastnostech, metodách a událostech všech HTML tagů v souboru. DOM je vytvořen

prohlížečem při načtení webové stránky a je reprezentovaný jako stromová struktura. Příkladem přístupu k HTML DOM je hned v prvním řádku kódu 6.3, kde přistupujeme k DOM za použití JavaScriptové API.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Nadpis HTML dokumentu</h1>
  <span>Blok textu</span>
</body>
</html>
```

Listing 6.1: Základní struktura HTML dokumentu

6.2.2 CSS

HTML sice slouží k popsání obsahu, to nám ale nestačí, jelikož výsledný dokument je neuspořádaný a nemá žádnou ucelenou a smysl dávající strukturu. Zde přichází na řadu kaskádové stylování CSS (Cascading Style Sheets), které nám umožní nejen rozložit obsah HTML dokumentu přesně jak potřebujeme, zároveň jsme schopní díky němu nastavit vzhledy elementů, fonty, velikost textu a další estetické prvky webové stránky. V CSS tvoříme skupiny pravidel, které následně přiřadíme tagu nebo skupinám tagů, které chceme nastylovat. V kódu 6.2 je možné vidět pravidla přiřazená tagu body, která mu přidávají barvené pozadí a velikost textu, který bude uvnitř tagu [19].

```
body{
  background-color: #FFFFFF;
  font-size: 16px;
}
```

Listing 6.2: CSS stylovací pravidla pro tag body

Pro CSS existují knihovny, které už mají připravené skupiny pravidel pro rychlé použití a urychlení vývoje. Při specifických požadavcích na vzhled aplikace je ale třeba použít vlastní stylování a tak to bude i v naší aplikaci.

6.2.3 JavaScript

Kdybychom přirovnali tyto 3 základní bloky k lidskému tělu, tak HTML by byla lidská kostra, CSS by se dalo přirovnat k vnějšímu vzhledu a JavaScript můžeme považovat za mozek, který hýbe s jednotlivými částmi lidského těla. JavaScript umožňuje na webové stránce dynamicky tvořit a upravovat obsah. S jeho použitím je možné časovaně aktualizovat obsah, spravovat multimédia a spoustu dalších funkcionalit. Na klientské části JavaScriptu je postavena speciální funkcionalita takzvaných API. Díky těmto API je možné jednoduše manipulovat a pracovat s HTML DOM, animovat elementy, ale taky pracovat s 2D a 3D grafikou. Manipulaci s HTML dokumentem za použití JavaScriptu je možné vidět na ukázce kódu 6.3. V kódu přiřadíme jednomu paragrafu funkci, která se zavolá při kliknutí na něj. Funkce otevře dotazové okno, kde vložíme text, který se následně zobrazí v našem paragrafu [20].

```
//vybereme paragraf v dokumentu
const paragraph = document.getElementsByTagName("p")[0];
//přiřadíme mu funkci, která se spustí při kliknutí na paragraf
paragraph.addEventListener("click", insertText);

//funkce která zavolá vstup do kterého vložíme text, který přiřadíme paragrafu
function insertText() {
  const text = prompt("Vložte text:");
  paragraph.innerHTML = `Váš text: ${text}`;
}
```

Listing 6.3: JavaScript code

JavaScript budeme používat v naší aplikaci jak na klientské straně ve zvoleném frameworku, ale taky ho budeme používat na programování naší serverové části.

6.3 Webový server

Jelikož data, které budeme zobrazovat na klientské části aplikace, budeme mít uloženy v databázi, tak je potřeba použít webový server. Na serveru bude implementováno REST API, které umožní přijímat požadavky a posílat na ně odpovědi použitím HTTP metod jako GET a POST.

6.3.1 REST API

API neboli Application Program Interface je dohodnutý způsob odesílání a přijímání dat mezi počítači. Příkladem takového API může být již dříve zmíněné Yahoo Finance API. Řekněme, že potřebujeme zobrazit informace o ceně nějaké konkrétní akcie, které nám právě Yahoo Finance API je schopné poskytnout. Způsob jak se zeptat Yahoo Finance na to, aby nám tuto informaci poslala,

je právě přes API dodané od Yahoo Finance, které nám říká na jaké adresy a v jakém formátu bychom měli náš požadavek pro získání dat poslat [21].

REST je API, které definuje funkce, které vývojáři můžou použít pro poslání požadavků a přijetí požadavků za použití HTTP protokolových metod. Jelikož požadavky jsou postavené na univerzálním HTTP protokolu a vrácené informace se typicky vrací v JSON formátu, tak může REST API být použito jakoukoliv webovou stránkou nebo aplikací [21].

6.3.2 Express.js

Pro implemetaci webového serveru a REST API použijeme framework Express, který běží na JavaScriptovém runtimovém prostředí Node.js. Express.js poskytuje jednoduchý a flexibilní systém směrování pro zpracování příchozích HTTP požadavků a jejich mapování na příslušné funkce. Poskytuje také middlewarový systém pro zpracování požadavků a odpovědí, což umožňuje vývojářům přidávat do aplikací funkce, jako je ověřování požadavků, logování a zpracování chyb [22].

Pro zprovoznění Express.js je nejprve potřeba nainstalovat Node.js, na kterém Express běží. Node.js je volně dostupný na jeho oficiálních stránkách a to jako instalátor, anebo soubor již připravený ke spuštění. Node.js je na stránkách dostupný pro všechny standardní operační systémy. Po nainstalování Node.js je potřeba použít správce balíčků npm. Npm funguje podobně jako správce balíčků pip. Je ho potřeba k instalaci a správě JavaScriptových balíčků a knihoven.

V kódu 6.4 můžeme vidět instalaci frameworku Express.js za pomoci správce balíčků npm. Je potřeba zmínit, že instalovat Express.js chceme v adresáři, kde budeme spravovat webový server. Npm nám nainstaluje naše závislosti do adresáře `node_modules`, který vytvoří v adresáři, kde příkaz pro instalaci spouštíme. Zároveň vytvoří soubor `package.json`, ve kterém ukládá informace o všech závislostech, které jsme v adresáři nainstalovali a pokud by jsme složku `node_modules` odstranili, tak je možné díky tomuto souboru a příkazu `npm install`, tyto závislosti znovu nainstalovat.

```
$ npm install express --save
}
```

Listing 6.4: Nainstalování Express.js přes správce balíčků npm

Nejjednodušší příklad Express.js REST API je znázorněn v kódu 6.5. Server poslouchá na portu 8000 a je schopný zpracovávat požadavek pro URI (Uniform Resource Identifier) `/ahoj`, který se přidá na konec URL adresy na které server poslouchá.

```
//index.js
//inicializovat Express.js
const express = require('express');
const app = express();
const port = 8000;
//zpracování GET požadavku na URI /ahoj
```

```
app.get('/ahoj', (req, res) => res.send('Příklad Použití GET Requestu'));
//nastavíme server pro naši ip adresu a port 8000
app.listen(port, () => {
  console.log('Listening on port ' + port);
});
}
```

Listing 6.5: Ukázka REST API v Express.js

6.4 Databáze

V aplikaci je potřeba zobrazovat vytvořené sítě, informace o akcích a mít přístup k jejich historickým cenám. K ukládání těchto dat nám slouží databáze. Databázi můžeme chápat jako kolekci strukturovaných dat. Je to jednoduše místo, kde jsme schopní ukládat a organizovat data [23]. K ukládání našich dat používáme databázový systém MySQL. Jde o relační databázový systém, to znamená, že data jsou uloženy v tabulkách a tabulky mají mezi sebou nějaký vztah. Pro vytvoření tabulek, vkládání dat a jejich následné získání, se používá v MySQL jazyk zvaný SQL (Structured Query Language). Jeho použití je možné vidět v části kódu 6.6, kde se vytváří jedna z tabulek použitých v aplikaci. Použitá verze MySQL je MySQL community, která je volně ke stažení z oficiálních stránek MySQL [24]. MySQL je zde dostupná pro operační systém Windows ve formě instalátoru a pro některé distribuce linux je možné stáhnout Repository Setup Package.

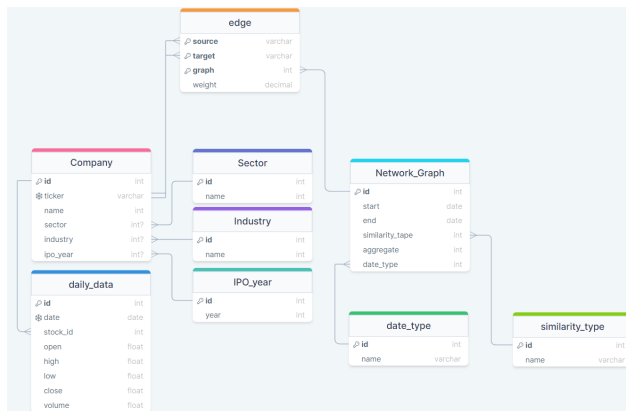
```
CREATE TABLE 'Sector' (
  'id' INT UNSIGNED NOT NULL AUTO_INCREMENT,
  'name' INT NOT NULL
);
ALTER TABLE 'Sector' ADD PRIMARY KEY('id');
```

Listing 6.6: Vytvoření tabulky Sector v SQL

6.4.1 Data v databázi

Pro uložení potřebných dat do databáze je nejprve potřeba určit která data potřebujeme ukládat a definovat strukturu tabulek a jejich vztahů ve formě databázového diagramu. Náš diagram bude velmi jednoduchý a bude obsahovat pouze 9 tabulek. Na obrázku 6.1 můžeme vidět vytvořený databázový diagram, který obsahuje všechny potřebné tabulky pro uložení dat. Tabulka Company spolu se Sector, IPO_year a Industry představují informace o akciových společnostech. Do tabulky daily_data jsou uložena denní historická data obchodovaných společností. Trochu složitější vztah nám tvoří v databázi reprezentace sítě. Pro každou síť v aplikaci existuje záznam v tabulce Network_Graph, který obsahuje cizí klíče do tabulek date_type a similarity_type. Tabulka date_type

nám říká, do jakého časového rozmezí síť patří, jestli se jedná o denní, měsíční nebo roční rozmezí. Similarity_type nám zase říká, jaká podobnostní metoda byla použita pro konstrukci sítě. Hranu sítě reprezentuje tabulka Edge, která obsahuje dva cizí klíče do tabulky Company, která v tomto případě slouží jako vrchol.



Obrázek 6.1: Databázový diagram tabulek použitých v databázi naší aplikace

6.5 Klientská část aplikace

Při vývoji aplikace jsme se zaměřili na moderní způsob tvorby aplikací zvaný SPA. Jde o způsob implementace obsahu webové aplikace do jedné stránky. Oproti klasickému přístupu několika stránkových aplikací, kdy je potřeba ze serveru načíst celou statickou stránku, načítáme dynamicky pouze danou část, kterou zrovna potřebujeme. To nám umožní v aplikaci snížit odezvu a zároveň tak zvýšit uživatelský prožitek. Pro vývoj SPA webových aplikací se používají JavaScriptové webové frameworky a knihovny. Knihovnu, kterou pro vývoj použijeme, se jmenuje React [25].

6.5.1 React

React je JavaScriptová knihovna pro vytváření uživatelských rozhraní. React umožňuje tvořit znovu použitelné UI (User Interface) komponenty a efektivně spravovat jejich stavy. Používá model deklarativního programování, kde vývojářům umožňuje definovat to, jak chtějí, aby UI komponenta vypadala podle momentálního stavu aplikace.

Jedna z výhod Reactu je ta, že používá virtuální DOM, který umožňuje upravovat UI prvky bez toho, aniž by manipuloval s opravdovým DOM. To zvyšuje výkon aplikace a lepší uživatelský prožitek.

React ale není kompletní framework pro vývoj webových aplikací, který definuje všechny potřebné prvky jako například Vue.js a Angular. Veškerou potřebnou funkcionalitu je potřeba si vybrat sám. Tento přístup na druhou stranu umožňuje vybrat si moduly, které můžou být lepší pro specifický případ použití.

Webové aplikace psané v Reactu tvoří již zmíněné komponenty. Komponenty jsou tedy jádrem celého Reactu. Komponenta je kus UI, který má svůj vlastní vzhled a programovou logiku. Je to tím, že React kombinuje HTML, css a JavaScript dohromady.

React komponenty ale nepoužívají klasický HTML, ale speciálně upravený markup jazyk, který je stejný jako HTML, akorát má programovatelnou funkcionalitu navíc. Jedná se o markup jazyk JSX.

React komponenta je JavaScriptovská funkce, která vrací JSX. Příklad komponenty můžeme vidět na ukázce kódu 6.7, kde jsou vytvořené dvě komponenty, které vrací markup JSX. V druhé komponentě jde vidět, že vytvořené komponenty můžeme vkládat do jiných komponent.

```
function Article({text}) {
  return (
    <article>{text}</article>
  );
}
function App(){
  return (
    <div>
      <Article text="Nějaký text"/>
    </div>
  );
}
```

Listing 6.7: React Komponenta a její použití v další komponentě

6.5.2 Create-react-app

Už dříve jsme se zmínili, že React sám o sobě není plnohodnotný framework a funkcionality jako například routování, spravování stavu a načítání dat je potřeba si vybrat a doinstalovat sám. Create-react-app nám umožňuje tuto část přeskočit a umožňuje nám nainstalovat React aplikaci obohacenou o všechny potřebné základní funkcionality. Je to jednoduchý nástroj, který vytvoří strukturu a nainstaluje potřebné knihovny zadáním jediného příkazu příkazové řádky. Příkaz používá npm balíček npx, který slouží k exekuci stažených balíčků správcem npm. Funkcionalita, kterou create-react-app přináší je pro potřeby naší aplikace naprosto dostačující. Instalaci Reactu za použití create-react-app je možné vidět na kódu 6.8, kde na prvním řádku dojde k instalaci za použití npx do adresáře my-app a na třetím adresáři je spuštění demo React aplikace, kterou nám create-react-app nainstaloval spolu se všemi balíky [26].

```
npx create-react-app my-app
cd my-app
```

Listing 6.8: Použití create-react-app a spuštění demo aplikace na localhost:3000

6.6 Knihovny a moduly

Veliká část funkcionalit v aplikaci je možná díky existujícím knihovnám a modulům. V aplikaci jsou použity knihovny různého typu a ty nejvýznamnější z nich jsou knihovny umožňující a usnadňující práci s grafy a knihovny pro jejich vizualizaci ve webovém prostředí. Všechny knihovny jsou nainstalovány přes správce JavaScriptových balíků npm.

6.6.1 Graphology

Jelikož význačná část naší aplikace pracuje s grafy, tak v aplikaci používáme knihovnu Graphology, která nabízí robustní a víceúčelovou reprezentaci grafu jako objektu. Kromě objektové reprezentace grafu nabízí knihovna Graphology také řadu funkcí, které implementují algoritmy a běžné utility jako jsou algoritmy pro rozložení, průchod a vypočítání specifických mír grafu a tak umožňují jednoduchý způsob pro manipulaci s grafem. V našem případě spoustu z těchto funkcí použijeme například pro výpočet centralit a algoritmus rozložení pro lepší 2D vizuální reprezentaci grafu v naší webové aplikaci [27]. Graphology použijeme jako součást knihovny sigma.js, která s knihovnou Graphology úzce spolupracuje.

Na ukázce kódu 6.9 je příklad vytvoření instance neorientovaného grafu v Graphology a následné přidání dvou vrcholů a jedné hrany mezi nimi. Instanci grafu je možné vytvořit i z více generické třídy Graph, která se bez jakékoliv bližší specifikace bude chovat jako graf neorientovaný.

```
import {UndirectedGraph} from 'graphology';

//vytvoření instance grafu
const graph = new UndirectedGraph();

// Přidání vrcholů
graph.addNode('APPL');
graph.addNode('BRK-A');

// Přidání hrany
graph.addEdge('APPL', 'BRK-A');
```

Listing 6.9: Vytvoření instance neorientovaného grafu z knihovny Graphology

6.6.2 React Sigma

React Sigma je jedním ze zásadních rozhodnutí, které vedly pro použití Reactu k vývoji naší webové aplikace. React sigma umožňuje použití populární knihovny sigma.js, jako vizuálních UI komponent právě v Reactu. Sigma.js slouží k 2D vizualizaci grafů ve webovém prostředí. K vizualizaci používá právě objektovou reprezentaci grafu z knihovny Graphology. Knihova JavaScriptové API WebGL pro renderování 2D scény a díky WebGL integraci je jednou z nejrychlejších vizualizačních grafových knihoven pro webové aplikace.

6.6.3 Recharts

V částech aplikace se používají klasické informativní grafy jako čárový graf a bar graf. Pro zobrazení grafů v naší aplikaci používáme knihovnu Recharts, která nabízí velkou přizpůsobitelnost grafů pro specifické účely.

6.7 Funkcionalita webové aplikace

V této části se zaměříme na webovou aplikaci, která byla vyvinuta za použití technologií zmíněných v předešlých sekcích této kapitoly. Webová aplikace je určena pro analýzu a převážně vizualizaci síťových dat. Použití knihovny React umožnilo vytvořit aplikaci, kde spousta UI komponent je znovupoužitelná do jiných projektů a je jednoduchá i jejich škálovatelnost.

Jedna z největších funkcionalit aplikace je množství sítí, které je možné zobrazit a analyzovat. tato funkcionalita je velmi inspirována způsoby analýzy akciových trhů jako je technická a orderflow analýza, kde je možné analyzovat data z pohledů různých časových období jako jsou dny, měsíce a roky. Jednotlivé sítě je pak možné si zobrazit v různém rozložení a porovnávat tak informace v síti. Další funkcionality jsou zobrazení časového průběhu sítě, kdy zobrazujeme změny v síti, jak se například mění měsíc po měsíci. Také je možné si zobrazit informace o centralitních mírách v síti a sledovat, zdali je mezi nimi nějaký vztah. Jedna sekce aplikace je věnována tvorbě portfolia ze sítě a jeho porovnání s akciovým indexem S&P500.

Aplikaci je možné jednoduše rozšířit a škálovat díky využití moderních technologií a nástrojů.

6.7.1 Úvodní zobrazení

Základní zobrazení při prvotním spuštění aplikace se skládá ze 4 částí, kdy každá z nich má svoji vlastní funkcionalitu, ale navzájem spolu spolupracují. Na obrázku 6.2 je možné vidět úvodní zobrazení aplikace. Vidíme, že úvodní zobrazení umožňuje vizualizaci sítí a nějaké jednoduché zobrazení informací o uzlech reprezentujících akcie. Vrchní panel slouží k výběru sítě, stylu zobrazení a přepínání módů aplikace.

Pro výběr sítě slouží na vrchním panelu 3 komponenty, jednotlivě Agregace, Range a Similarity Type. Agregace nám určí pro jaké časové rozmezí chceme síť zobrazit. Je tedy možné si zobrazit síť

pro dny, měsíce a roky, kde je pak možné zvolit si i délku tohoto časového rozmezí. Pro dny je to například 1 - 30 dnů a pro měsíce zase 1 - 6 měsíců. Range se nám aktualizuje pro zvolení časového rozmezí v komponentě agregace a je možné zde vybrat síť se zvolenou časovou délkou pro specifické datum. Poslední důležitá komponenta je Similarity Type, kde si volíme, podle které podobnostní funkce byla síť konstruována.

Na vrchním panelu se také nachází komponenta Mode, která slouží pro přepínání mezi módy aplikace, jako je portfolio sekce nebo zobrazení časového průběhu sítí.

Další důležitou částí úvodního zobrazení je levý panel, který slouží pro přepnutí mezi zobrazením momentální sítě a zobrazením informací o mírách momentálně zvolené sítě v sekci zobrazení.



Obrázek 6.2: Úvodní stránka aplikace pro vizualizaci a výběr sítě

6.7.2 Vizualizace sítě

V sekci 6.6.2 jsem se zmínil o knihovně React Sigma. Jeho použití velmi usnadnilo vývoj síťových UI komponent, použitých k vizualizaci síťových dat. Vizualizaci sítě je možné vidět na úvodním zobrazení aplikace. Provedení vizualizace sítě je možné vidět na obrázku 6.2, kde je zobrazená síť ukázána na dvou bílých komponentách. Tyto komponenty umožňují síť zobrazovat s různým obarvením vrcholů, můžeme si tak například zobrazit síť obarvený podle komunit nalezených algoritmem Louvain a vedle něj si obarvit jednotlivé vrcholy podle toho, do jakého sektoru daná akcie spadá. Obarvit je uzly možné také podle jejich centralit a můžeme si tak zobrazit, které uzly jsou pro z pohledu dané centrality nejdůležitější. Změnit obarvení vrcholů sítě je možné přes ozubené kolečko v komponentě, kde je síť vykreslená. Síť nemusíme zkoumat pouze na dvou komponentách, styl rozložení je možné si zvolit na vrchním panelu pod názvem Layout Option a je možné zkoumat až 4 instance sítě najednou. Komponenty také umožňují pohyb ve vykreslené scéně za použití myši a možnost

přibližovat a oddalovat scénu použitím kolečka myši. Na jednotlivé vrcholy je možné kliknout a v pravém panelu, viz obrázek 6.2, se nám zobrazí informace o akci, kterou uzel reprezentuje.

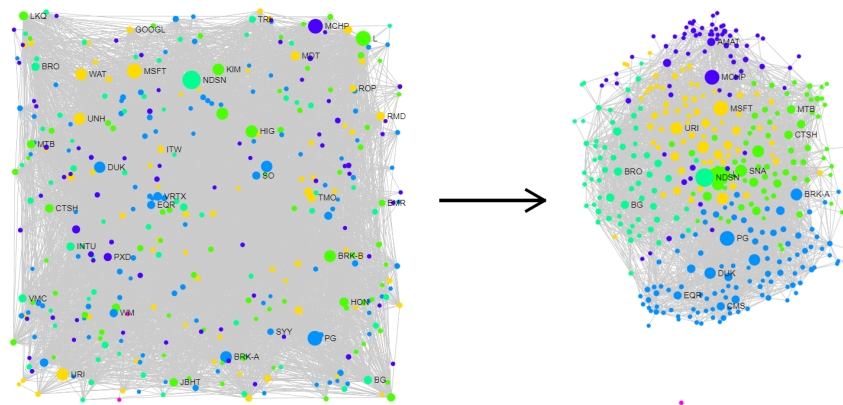
6.7.3 Rozložení sítě

Rozložení sítě na obrázku 6.2 je docíleno použitím algoritmu ForceAtlas2 [28], který je implementovaný knihovnou Graphology. ForceAtlas2 je silově řízený algoritmus rozvržení grafu používaný k vizualizaci grafů většího měřítka. Jedná se o iterační algoritmus, který simuluje chování fyzikálních sil, působících na systém částic, k umístění uzlů v grafu. Algoritmus je navržen tak, aby vyrovnával přitažlivé síly mezi spojenými uzly s odpudivými silami mezi uzly, které nejsou spojeny.

Algoritmus považuje každý uzel v grafu za částici a hrany za pružiny. Při každé iteraci algoritmus aplikuje fyzikální síly na částice (uzly) na základě jejich aktuální polohy a polohy uzlů, které jsou s nimi propojeny. Na základě působících sil pak algoritmus upraví jejich polohy.

ForceAtlas2 bere v úvahu několik parametrů, které je možné upravit, aby bylo dosaženo požadovaného rozložení grafu.

Na obrázku 6.3 je možné vidět porovnání sítě, na kterou nebyl použit žádný algoritmus rozložení a sítě, na kterou byl aplikován algoritmus ForceAtlas2. V lépe rozložené síti je pak možné vidět, že uzly, které jsou obarveny podle komunit vypočtených algoritmem Louvain, patří do stejných komunit se přitahují.



Obrázek 6.3: Použití layout algoritmu ForceAtlas2 pro lepší vizualizaci sítě

6.7.4 Časová síť

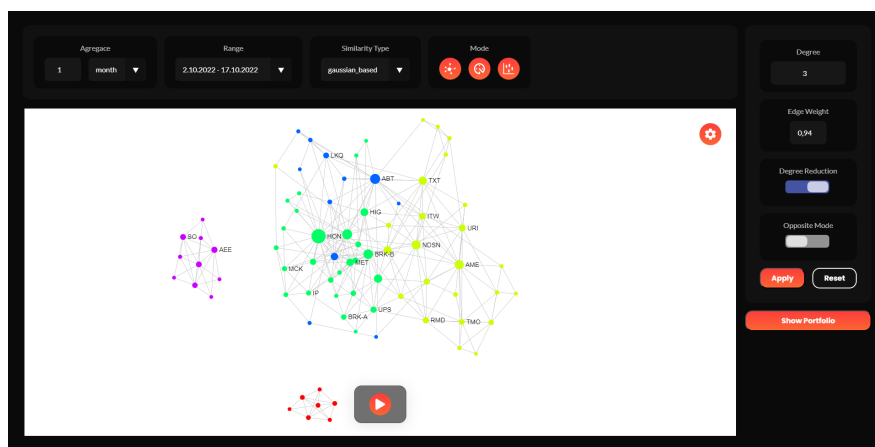
V úvodním zobrazení si na horním panelu, kde si vybíráme, kterou síť chceme vizualizovat, máme možnost se přepnout do sekce časové sítě přes první ikonu v záložce mode. V této sekci se setkáme s podobným UI jako v úvodním zobrazení. Na vrchním panelu jsme schopni si stejným způsobem jako v úvodní sekci vybrat síť. Je tu menší rozdíl, když vybereme například síť pro 10 dnů, tak se nám nezobrazí 10 denní síť z vybraného období, ale zobrazí se nám 1 denní síť z prvního dne vybraného

období. Tuto síť je možné následně animovat a vidět změnu v komunitách a celkovém rozložení sítě den po dni, až narazíme na poslední den z vybraného časového rozmezí. Tato funkce nám umožňuje sledovat, jak se mění komunity v tomto časovém období a sledovat tak, jestli komunitu tvoří stále stejné akcie například ze stejného sektoru, anebo se komunity neustále mění.

6.7.5 Porovnání portfolia

Do sekce aplikace pro tvoření portfolia se dostaneme z komponenty Mode vrchního panelu úvodního zobrazení. Tato sekce slouží pro vytvoření portfolia ze síťových dat. Skládá se ze dvou důležitých částí, a to část pro výběr a vizualizaci sítě, ze které chceme portfolio vytvořit a z části, kde portfolio porovnáme proti akciovému indexu S&P500.

První část je možné vidět na obrázku 6.4, kde si na horním panelu jsme schopni vybrat síť, z které chceme portfolio tvořit. Jelikož dny jsou velmi krátká časová složka pro porovnávání a tvorbu portfolia, tak se zde omezíme pouze na měsíce a roky. Pravý panel slouží pro omezení vlastností zobrazované sítě. Tento panel obsahuje 4 důležité části, jsou to stupeň uzlu, váha hrany, redukce stupně a opačný mód.

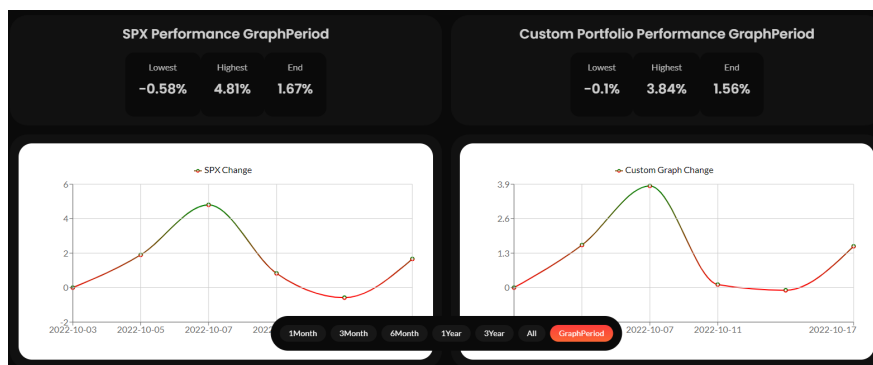


Obrázek 6.4: Vizuální část sítě při tvorbě portfolia

Stupněm určíme, které uzly nám zůstanou v nové síti. Pokud zvolíme například stupeň 3, tak v nové síti budou uzly, pro které v předešlé síti platilo, že měly stupeň větší nebo roven 3. Je možné zapnout funkci redukce stupně, kdy naopak říkáme, že chceme aby v nové síti měly všechny vrcholy minimální stupeň 3. Další omezení je váha hrany, a to nám odstraní ze sítě hrany, které jsou menší než zadaná váha. Poslední funkcí je opačný mód, ten nám umožňuje vytvořit podsít s přesně opačnými parametry. Je tak možné se podívat na portfolio, kde si uzly v síti nebyly dostatečně podobné, výsledkem tak může být lépe diverzifikované portfolio, které nemusí mít větší výnosy než zvolený index, ale může být bezpečnější z pohledu možných ztrát.

Druhou částí této sekce je porovnání portfolia z vytvořené podsítě s akciovým indexem S&P500. Portfolio je možné porovnat pro 7 časových období, kde jedním z nich je i časové období vybrané sítě.

Všech 7 časových období pro porovnání a celou část je možné vidět na obrázku 6.5. Pro vybranou periodu pak vidíme výnosnost indexu a našeho portfolia, jak na čárovém grafu, tak v komponentách nad čárovými grafy.

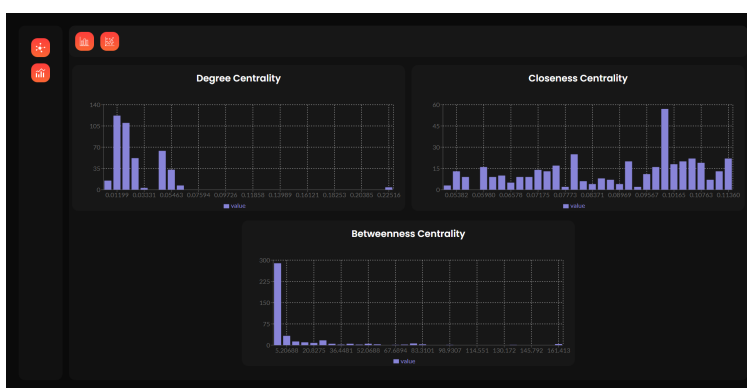


Obrázek 6.5: Porovnání vytvořeného portfolia s akciovým indexem S&P500

Každá akcie má v portfoliu svůj podíl, ale ne všechny akcie jsou si zde rovny. Vrcholy akcie s vyšším stupněm mají v portfoliu vyšší podíl, než vrcholy s nižším stupněm. Tento přístup byl vybrán, jelikož akcie tvořící portfolio normálně nemají stejný podíl v portfoliu. Je naprosto normální a zároveň smysluplnější mít jednu akcii tvořící například 10% portfolia a jinou jenom 5%.

6.7.6 Míry centrality

Jak jsem se v části 6.7.1 zmínil, jednotlivé centrality je možné použít pro obarvení sítě a sledovat tak, které uzly jsou z pohledu dané centrality nejdůležitější. Zkoumané centrality jsou degree, closeness a betweenness. Obarvení je vedeno přes tříbarevný gradient, kdy modrá barva nám značí nejnižší, zelená středové a červená nejvyšší hodnoty.



Obrázek 6.6: Distribuce mír centralit vizualizované sítě

O centralitách si můžeme také zobrazit více informací a to přepnutím do této sekce přes ikonku sloupcového grafu v levé komponentě úvodního zobrazení. Zde se nám zobrazí distribuce pro všechny

3 centrality. Tyto distribuce je možné vidět na obrázku 6.6. Z této sekce se můžeme dostat ještě do jedné části s informacemi o centralitách, kde jsou zobrazeny bodové grafy a je možné zde zkoumat vztah centralit mezi sebou. Jsme tak schopní sledovat, jestli vrcholy, které mají vysoké hodnoty jedné centrality, vysoké hodnoty u centrality jiné, anebo mají naopak hodnotu jiné centrality velmi nízkou.

Kapitola 7

Závěr

Cílem bakalářské práce byla analýza akciových trhů. Podívali jsme se na dva problémy a to vizualizace detekce komunit a vizualizace průběhu vybraných tržních propadů. Při vizualizaci komunit jsme zjistili, že jsme schopni zobrazit vztah, který mají společnosti v rámci sektoru v reálném světě v námi konstruovaných sítí. Při vizualizaci průběhu vybraných tržních propadů jsme zjistili, že jde pomocí sítí zobrazit šíření propadů cen akcií ze sektoru na sektor. Povedlo se nám potvrdit, které sektory byly u vzniku vybraných tržních propadů a sledovat, jak se změna v cenách akcií šíří do zbytku sítě a vyzorovat tak, které sektory byly nejpostiženější.

Jednou z částí bylo vybrat vhodná data z akciových trhů, které následně použijeme při konstrukci sítí. Tyto data bylo následně potřeba vhodným způsobem analyzovat a vytvořit webovou aplikaci pro jejich reprezentaci a možnou analýzu. Jako vhodná data jsme vybrali denní informace o cenách a obchodovaném objemu akcií. Bylo potřeba si stanovit, které akcie zahrneme a z jakého časového období. Nakonec bylo vybráno 410 akcií obchodovaných na amerických burzách NASDAQ a NYSE. Informace o cenách byly následně staženy pro období od 1.1.2007 až po 18.10.2022. Část období byla zahrnuta v analýze, kde jsme se podívali na velkou recesi, medvědí trh v srpnu 2011 a covidový tržní propad. Tato období nezahrnují všechna stažená data, celý rozsah období je však přístupný ve webové aplikaci. Dále jsme si uvedli způsob, jakým budeme konstruovat sítě pro naši analýzu a aplikaci. Vybraným algoritmem pro konstrukci sítí je LRnet, který používáme s dvěma podobnostními funkcemi.

Webová aplikace je programovaná v JavaScriptové knihovně React a serverová část je programována v JavaScriptovém Node.js. Jsme schopni si v ní zobrazit sítě jak ze zkoumaných období analýzy, tak je možné je vybírat podle časové periody. Po vybrání časové periody si pak můžeme vybrat z datumů o délce této časové periody. Sítě je možné obarvit podle nalezených komunit algoritmem Louvain, sektorů a metrik centrality K vybrané sítí jsme si schopni zobrazit distribuci míry centrality a na bodovém grafu sledovat vztah, který mají centrality mezi sebou. V jedné sekci animujeme průběh sítě v čase a sledujeme tak změny, které nastávají v síti se změnou v čase. Podle sítě, kterou si vybereme jsme v jedné části aplikace schopni vytvořit ze sítě podsít, jejíž vrcholy

slouží k tvorbě portfolia, které je možné porovnat s výnosností tržního indexu S&P500.

Celkově se sítě projevily velmi užitečné. Zejména při vizualizaci vztahů, které mají jednotlivé akcie nejen mezi sebou v rámci sektoru, ale taky vztahy které mají sektory mezi sebou. Při vizualizaci průběhu tržních propadů se sítě projevily jako výborný nástroj, který umožňuje jednoduchý způsob pro sledování šíření cenových propadů mezi akciemi a sektory.

Aplikaci by bylo možné dále rozšířit o nové funkcionality. Jedna z možných funkcionalit je vizualizace průběhu zobrazované sítě v aplikaci, která by byla omezená pouze na sítě, kde je délka časového období minimálně jeden měsíc. Aplikace obsahuje velké množství sítí a toho by se dalo využít pro zobrazení informací jako jsou míry centrality v průběhu času. Bylo by tak možné sledovat, jestli mají některé akcie vysokou některou z mír centralit po delší časové období.

Literatura

1. INVESTOPEDIA. *Investopedia* [online]. [cit. 2023-04-02]. Dostupné z: <https://www.investopedia.com/>.
2. MURPHY, John. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. USA: New York Institute of Finance, 1999. ISBN 978-0735200661.
3. JURVETSON, Steve. *Internet Splat Map* [online]. [cit. 2023-04-10]. Dostupné z: <https://www.flickr.com/photos/jurvetson/916142/>.
4. PETR, Kovář. *Teoriegrafů* [online]. [cit. 2023-04-27]. Dostupné z: https://homel.vsb.cz/~kov16/files/skriptum_teorie_grafu.pdf.
5. SILVA, Thiago Christiano; ZHAO, Liang. *Machine Learning in Complex Networks*. 1st. Springer Publishing Company, Incorporated, 2016. ISBN 3319172891.
6. NGUYEN, Quan. *Models and Methods for Big Graph Visualization*. 2013-06. Dis. pr.
7. NEWMAN, Mark. *Networks: An Introduction*. USA: Oxford University Press, Inc., 2010. ISBN 0199206651.
8. RESEARCHGATE. *The french 400kV power grid* [online]. [cit. 2023-04-10]. Dostupné z: https://www.researchgate.net/figure/The-french-400kV-power-grid-with-the-most-vulnerable-edge-black-thick-line-the-most_fig2_267719003.
9. DONOHUE, Brian. *Researchers Graph Social Networks to spot Spammers* [online]. [cit. 2023-04-10]. Dostupné z: <https://threatpost.com/researchers-graph-social-networks-spot-spammers-061711/75346/>.
10. PYTHON. *Python Download* [online]. [cit. 2023-04-10]. Dostupné z: <https://www.python.org/downloads/>.
11. PYDATA. *pandas* [online]. [cit. 2023-04-10]. Dostupné z: <https://pandas.pydata.org/>.
12. PYDATA. *pandas-datareader* [online]. [cit. 2023-04-10]. Dostupné z: <https://pydata.github.io/pandas-datareader/index.html>.

13. OCHODKOVA, Eliska; ZEHNALOVA, Sarka; KUDELKA, Milos. Graph Construction Based on Local Representativeness. In: CAO, Yixin; CHEN, Jianer (ed.). *Computing and Combinatorics*. Cham: Springer International Publishing, 2017, s. 654–665. ISBN 978-3-319-62389-4.
14. OCHODKOVA, Eliska; ZEHNALOVA, Sarka; KUDELKA, Milos. *Graph Construction Based on Local Representativeness* [online]. [cit. 2023-04-27]. Dostupné z: https://home1.vsb.cz/%20kud007/lrnet_files/.
15. BARABÁSI, Albert-László. *Network Science*. Cambridge University Press, 2016. ISBN 9781107076266. Dostupné také z: <http://networksciencebook.com/>.
16. JAYAWICKRAMA, Thamindu Dilshan. *Community Detection Algorithms* [online]. [cit. 2023-04-15]. Dostupné z: <https://towardsdatascience.com/community-detection-algorithms-9bd8951e7dae>.
17. RITA, Luís. *Louvain Algorithm* [online]. [cit. 2023-04-15]. Dostupné z: <https://towardsdatascience.com/louvain-algorithm-93fde589f58c>.
18. O'GRADY, Brian. *What is HTML? An Introduction* [online]. [cit. 2023-04-10]. Dostupné z: <https://codeinstitute.net/global/blog/what-is-html-and-why-should-i-learn-it/>.
19. DOCS, MDN Web. *What is CSS?* [online]. [cit. 2023-04-10]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS.
20. DOCS, MDN Web. *What is JavaScript?* [online]. [cit. 2023-04-10]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript.
21. BENHAROSH, Joseph. *What is REST API? in plain English* [online]. [cit. 2023-04-10]. Dostupné z: <https://phpenthusiast.com/blog/what-is-rest-api>.
22. KINSTA. *What Is Express.js? Everything You Should Know* [online]. [cit. 2023-04-10]. Dostupné z: <https://kinsta.com/knowledgebase/what-is-express-js/>.
23. HOSTINGER. *What is MySQL: MySQL Explained For Beginners* [online]. [cit. 2023-04-11]. Dostupné z: <https://www.hostinger.com/tutorials/what-is-mysql>.
24. ORACLE. *MySQL Community Downloads* [online]. [cit. 2023-04-11]. Dostupné z: <https://dev.mysql.com/downloads/>.
25. WELT | RASCASONE, Experten in der digitalen. *Co je jednostránková webová aplikace (SPA) a kdy ji využít?* [online]. [cit. 2023-04-11]. Dostupné z: <https://www.rascasone.com/cs/blog/jednostrankova-webova-aplikace-spa>.
26. STARTED, Getting. *Create React App* [online]. [cit. 2023-04-12]. Dostupné z: <https://create-react-app.dev/docs/getting-started>.

27. (2021), Guillaume Plique. *Graphology, a robust and multipurpose Graph object for JavaScript*. *Zenodo*. [online]. [cit. 2023-04-12]. Dostupné z: <https://doi.org/10.5281/zenodo.5681257>.
28. JACOMY, Mathieu; VENTURINI, Tommaso; HEYMANN, Sebastien; BASTIAN, Mathieu. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLOS ONE*. 2014-06, roč. 9, č. 6, s. 1–12. Dostupné z DOI: [10.1371/journal.pone.0098679](https://doi.org/10.1371/journal.pone.0098679).