

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
ЕКОНОМІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

**РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ МОДЕЛЮВАННЯ ТА
ПРОГНОЗУВАННЯ ЦІН АКЦІЙ НА ФОНДОВІЙ БІРЖІ**

Студента 4 курсу денної форми навчання
спеціальності 051 «Економіка»
освітньо-професійної програми
«Економічна кібернетика»
Атояна Андрія Гарійовича

Науковий керівник
д.е.н., доцент
Ставицький Андрій Володимирович

Засвідчую, що в цій бакалаврській
роботі немає запозичень із праць
інших авторів без відповідних посилань

Студент _____
(підпис)

Робота допущена до захисту в ЕК рішенням кафедри економічної кібернетики від 28 травня 2020 р., протокол № 12

В.о.завідувача кафедри економічної кібернетики,
доктор економічних наук, професор
Ляшенко Олена Ігорівна

(підпис)

Київ - 2020

РЕФЕРАТ

Бакалаврська робота містить: 107 ст., 46 рис., 4 табл., 54 джерела, 2 додатки

Ключові слова: фондовий ринок, фінансові інструменти, програмний комплекс, моделювання волатильності, часові ряди, веб-скрепінг, бот, аналіз.

Об'єкт дослідження: біржові індекси акцій компаній.

Мета дослідження: розробка програмного комплексу для аналізу та прогнозування цін акцій фондових ринків.

Методи дослідження: моделі Arima, Arch, Garch (її різновиди), нейронні мережі ARNN.

Наукова новизна, теоретична значимість дослідження: обумовлена аналізом та порівнянням методів моделювання фінансових часових рядів на різних вибірках, що дозволяє визначати ефективність відповідних моделей.

Практична цінність: полягає в можливості використання даного програмного комплексу для оптимізації процесів аналізу та прогнозування біржових індексів, а також створення більш складного комплексу шляхом удосконалення існуючих модулів

RESUME

Kyiv National Taras Shevchenko University,

Faculty of Economics, Department of Economic Cybernetics

Key words: stock market, financial instruments, modeling volatility, software, time series, web scraping, bot, analysis.

The graduation research of student: «Development of a software for modeling and forecasting stock prices on the stock exchange».

Deals with: automation of the collection, processing and analysis of financial time series by developing software that is made up of two modules: an automated data scraping algorithm and an interactive control panel for data handling.

The practical significance is the possibility of using software by investors, researchers and stock market traders for modeling and forecasting stock market indices.

Pages 107, tables 4, bibliog 54, append 2.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ ФУНКЦІОНУВАННЯ ФОНДОВИХ РИНКІВ.	6
1.1 Сутність фондових ринків.	6
1.2 Цінні папери та їх різновиди	8
1.3 Процедура лістингу та суб'єкти фондової біржі	13
1.4 Види аналізу на фондових біржах.....	20
РОЗДІЛ 2. МЕТОДИ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.....	25
2.1 Загальні положення Arima моделі.....	25
2.2 Модель Авторегресивної умовної гетероскедастичності	29
2.3 Узагальнена модель GARCH та її різновиди.....	31
2.4 Модель нейронних мереж ARNN.....	34
РОЗДІЛ 3. СКЛАДОВІ ПРОГРАМНОГО КОМПЛЕКСУ ТА РЕЗУЛЬТАТИ АНАЛІЗУ.....	36
3.1 Налаштування програмного комплексу.....	36
3.2 Опис роботи алгоритму скрепінгу даних.....	44
3.3 Моделювання та прогнозування цін акцій.....	49
ВИСНОВКИ	63
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	65
ДОДАТКИ.....	70

ВСТУП

Моделювання та прогнозування мінливості інструментів фінансового ринку є невід’ємною частиною роботи на фондовій біржі. Роками інвестори та дослідники були зацікавлені у розробці та тестуванні моделей поведінки цін різних фінансових інструментів. Пороте, притаманні їм динамічність, нелінійність, не стаціонарність та хаотичність зробили цей процес надзвичайно складним.

З виникненням і стрімким розвитком комп’ютерних технологій і обчислювальної техніки в світі відбуваються безперервні зміни в життєдіяльності фінансових ринків. Все більша їх кількість відходять від традиційних підходів та стають автоматизованими. Разом з тим з’являється безліч нових фінансових інструментів торгівлі, що знаходять широке застосування на фондових біржах.

Сьогодні, більшість запропонованих автоматизованих системи, такі як «MetaTrader 4», «eOption» та «Interactive Brokers», надаються в закритому для користувача вигляді, без можливості дослідження алгоритму його роботи.

Саме тому, брокерські компанії та інвестори, що функціонують на ринку цінних паперів, стикаються з необхідністю побудови, тестування і застосування власних автоматизованих систем аналізу.

Метою дипломної роботи є розробка програмного комплексу для аналізу та прогнозування цін акцій фондових ринків. Структура комплексу складається з двох програмних модулів: автоматизованого алгоритму скрепінгу даних та інтерактивної панелі управління для їх аналізу.

Головним ресурсом забезпечення даних є платформа Yahoo! Finance, яка публікує безкоштовну інформацію інструментів фінансового ринку. Технічний перелік необхідних програм та компонентів складається з: R & R Studio, Docker Desktop, ChromeDriver, TightVNC.

Відповідно до мети, **завдання роботи** полягає в:

- дослідженні наукових робіт стосовно методів та підходів аналізу інструментів фінансового ринку.
- виборі ефективної методології прогнозування часових рядів.
- розробці алгоритму збору даних, та його автоматизації.

- створенні інтерактивної панелі управління для попередньої обробки даних та їх аналізу.
- оцінці результатів роботи програмного комплексу, порівняння моделей та виборі найкращої з них.
- розробці прогнозу часового ряду на наступні n періодів.

Об'єктом дослідження є біржові індекси акцій компаній.

Предметом дослідження є автоматизація процесу аналізу та прогнозування акцій з використанням статистичних методів, та методів машинного навчання.

Практичне значення полягає в можливості використання даного програмного комплексу для оптимізації процесів аналізу та прогнозування біржових індексів, а також створення більш складного комплексу шляхом удосконалення існуючих модулів. Теоретична цінність обумовлена аналізом та порівнянням методів моделювання фінансових часових рядів на різних вибірках, що дозволяє визначати ефективність відповідних моделей.

Структура дипломної роботи складається з вступу, змісту, трьох розділів та _ підпунктів, списку використаної літератури та додатків. Загальний обсяг роботи становить 107 сторінок.

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ ФУНКЦІОНУВАННЯ ФОНДОВИХ РИНКІВ.

1.1 Сутність фондових ринків.

Фондовий ринок (або ринок цінних паперів) представляє собою сукупність фінансових відносин між емітентами та інвесторами стосовно випуску, купівлі, продажу цінних паперів. Місцем проведення даних операцій є Фондова біржа.

Особливість, яка виділяє фондовий ринок серед інших – є унікальність його фінансових інструментів. Фактично, цінні папери становлять собою певні зобов'язання емітента перед їх власником, тоді як кошти, сплачені інвестором, призначенні для формування фінансового капіталу. Завдяки цьому, вільні грошові кошти спрямовуються у ті галузі економіки де вони є найбільш необхідні.

Як і будь-який економічний об'єкт, фондовий ринок має свою класифікацію. Особливу увагу необхідно приділити саме стадії обігу цінних паперів, відповідно до якої ринок поділяється на первинний та вторинний.

Первинний ринок є місцем розміщення вперше випущених цінних паперів. Саме тут компанії, пройшовши складну процедуру лістингу, отримують можливість розміщення власних фінансових інструментів. Здійснюючи емісію, вони переслідують за мету залучення додаткових фінансових інвестицій. Таким чином, емітенти виступають стороною пропозиції на первинному ринку. Інвестори ж, шукаючи можливість вдалого застосування власних коштів купують ці папери. Вигода, яку отримують інвестори залежить від типу паперів та умов, які там прописані. Гарантом дотримання цих умов, у даному випадку, виступає сама біржа.

Таким чином, можна дійти винову, що саме на первинному ринку відбувається залучення тимчасово вільних коштів та їх інвестиція в економіку.

Набувши нових власників, цінні папери потрапляють на вторинний ринок, характерною рисою якого є наявність спекуляції. Фактично, саме тут встановлюється реальна ціна фінансових інструментів внаслідок постійного перерозподілу власності між суб'єктами ринку. Прибуток у даному випадку досягається за рахунок зміни курсу, адже будь-які зміни одразу ж знаходять своє відображення в ціні фінансового інструменту.

Слід зазначити, що вагомою перевагою фондової ринку над альтернативними інститутами, є фактична можливість у будь-який момент реалізувати фінансовий актив за прийнятною ціною, та навпаки. Іншими словами, забезпечення ліквідності фінансових інструментів є обов'язковою умовою інвестування в них.

Виходячи з інформації зазначеної вище, ми можемо виділити основні функції фондового ринку, які полягають в:

- залученні капіталу та забезпеченні його ефективного використання;
- зборі інформації про суб'єктів і об'єктів ринку і доведення її до учасників;
- формуванні ринкових цін, їх постійний рух;
- гнучкому перерозподілі коштів між галузями, можливість концентрувати їх у найбільш перспективних напрямках;
- залученні іноземних інвестицій;

Дуже довго фондовий ринок був недоступний широкій аудиторії, а йому був притаманний незрозумілий характер. На справді ж робота на фондовому ринку зводиться до купівлі та продажу акцій різних компаній й отриманні вигоди. Правило "купуй дешевше і продай дорожче" є, мабуть, точним описом суті роботи на біржах.

Надалі, коли доступ до торгівлі на світових фондових майданчиках отримали практично всі бажаючі, для деяких це стало однією з важливих статей доходу і заробітку. Сьогодні значний фінансовий оборот, який проходить через світові фондові біржі становить не один трильйон доларів. Такі обсяги здатні впливати не тільки на капітал, який присутній на торгах і ціну на окремі фондові інструменти, а й чинити сильний вплив на економіку різних країн.

У період фінансової нестабільності фондовий ринок відіграє так само не останню роль. Інвестування в цінні папери більш стійких до потрясінь компаній використовується як засіб збереження власних коштів від можливих ризиків і втрат.

На сьогоднішній день більшість країн з економікою, що розвивається і розвиненою економікою мають свої біржові майданчики. Серед найбільших ринків, зі значними фінансовими обсягами операцій є фондовий ринок США, фондовий ринок Європи і Азіатські фондові біржі.

1.2 Цінні папери та їх різновиди

Як ми зазначили раніше, особливістю фондової біржі є те, що об'єктом купівлі-продажу виступають цінні папери. Даний документ містить в собі сукупність майнових прав власника (або фактичних зобов'язань емітента). В залежності від виду фінансових інструментів, інвестор отримує прибуток у виді дивідендів або відсотків.

До основних фінансових інструментів, які набули широкого застосування на фондовій біржі відносяться: акції, облігації, ф'ючерси, опціони, тощо. Кожен з них відіграє особливу роль у біржовій торгівлі.

Акції - різновид цінних паперів, який може розглядатися як безстроковий кредит, який був наданий компанії. Купуючи їх, інвестор стає співвласником компанії, тобто йому належить певна доля у статутному капіталі. Також він має право на участь в управлінні та можливість отримувати прибуток у вигляді дивідендів. Компанії, що випускають акції, набувають статусу акціонерного товариства (АТ).

Акціонерів компанії умовно можна поділити на власників звичайних та привілейованих акцій. Різниця полягає у тому, що виплата дивідендів для власників звичайних акцій не є фіксованою. У періоди, коли компанія має певні фінансові труднощі, дивіденди можуть не виплачуватись. Проте для власників привілейованих акцій, робиться виключення, тому вони завжди отримують частину чистого прибутку компанії. Ще однією особливістю є те, що власники привілейованих акцій, зазвичай, не приймають участь в управлінні. Це правило не стосується випадків, коли мова йде про ліквідацію компанії, або банкрутство.

Акціонерне товариство розміщує привілейовані акції різних класів, з метою залучення більшої кількості фінансових інвестицій. Умовою їх розміщення є черговість отримання дивідендів і виплат з майна ліквідованого товариства. Проте їх частка у статутному капіталі компанії не може перевищувати 25%.

На фондовій біржі акція є одним з найпривабливіших фінансових інструментів. Вдало вибравши компанію для інвестування, ми можемо примножити вкладений капітал за рахунок зростання акцій та відсотків від їх вартості – дивідендів.

Серед переваг слід виділити те, що для отримання дивідендів фактичним власником акцій необхідно бути лише 1 день. Ми можемо сьогодні купити акцію,

потрапити під день, так званої "відсічки акціонерам", а потім продати її, отримавши солідний прибуток. Таким чином дивідендна прибутковість має величезну перевагу перед прибутковістю наприклад банківських депозитів, де гроші лежать у банку цілий рік.

Проте, необхідно пам'ятати, що кожній фондовій біржі властивий свій режим торгів. Так, для Нью-Йоркської фондової біржі "NYSE", T + 2 – основний режим для більшості операцій. Цифра «2» означає, що фактичний перехід цінних паперів відбувається через 2 дні після операції. Наприклад, якщо інвестор 01.08.18 купив акції якоїсь компанії, то фактично їх власником він стане тільки 03.08.18. Відповідно, з продажом ситуація аналогічна.

Такий режим був введений для зручності проведення технічних і юридичних процедур, пов'язаних з переходом прав на папери. За аналогією з T + 2 існують режими торгів T + 1 і T0, в яких в основному торгуються облігації.

Облігації - це фактично боргові розписки. Їх випускає емітент - компанія або держава, якій потрібні гроші. Інвестори купують облігації і таким чином дають свої гроші в борг цій компанії, розраховуючи отримати певний дохід.

Такий варіант зазвичай вибирають консервативні інвестори, які хочуть, щоб їх гроші були вкладені в найбільш надійні активи. Коли інвестор купує облігації, він не стає співвласником бізнесу компанії. Він стає співвласником боргу компанії.

Повна сума, терміни і розмір виплат цього доходу (якщо планується кілька платежів), як правило, відомі під час купівлі. Можливість заздалегідь оцінити свою вигоду - це те, чим облігація відрізняється від інших цінних паперів. У встановлений термін відбувається погашення облігацій, тобто емітент виплачує їх власникам номінальну вартість, зазначену на самих облігаціях.

Основний ризик для інвестора - банкрутство компанії-емітента. У цьому випадку він може втратити вкладені гроші. Необхідно зазначити, що державні облігації є більш надійними, але і прибутковість по них буде нижче ніж корпоративним. Інвестування в корпоративні облігації є більш ризиковим, але дохідність їх з буде вище.

За формою виплати доходу облігації поділяються на:

- **процентні (купонні) облигації** - за ним інвестор отримує відсотки від номінальної вартості. За деякими облигаціями буває тільки одна виплата, за іншими - декілька. Виплату відсотків називають погашенням купона. Така назва зберіглася з часів, коли облигації були паперовими: до них кріпилися купони на виплати, які відривали в момент погашення. Найчастіше купонні облигації бувають з фіксованою процентною ставкою, тобто інвестор заздалегідь знає розміри виплат. Але бувають і з плаваючою процентною ставкою - виплати змінюються, через певні ситуації на ринку цінних паперів;

- **дисконтні облигації** - при покупці стоять нижче номінальної вартості, але погашаються за номінальною. Наприклад, ви купили облигацію зі знижкою - за 800 євро, а коли її термін дії закінчився, отримали за нею 1000 євро - всю номінальну вартість. Різниця між ціною розміщення та номінальною вартістю і називається дисконтом.

Облигації як і будь-який інший фінансовий інструмент можуть падати в ціні, а можуть і рости. Інвестор, здійснюючи покупку облигацій може заробити на зростанні їх вартості. Припустимо, якась компанія випустила облигації номіналом 200 г. од. Інвестор купив цю облигацію і через деякий час її вартість зросла до 210 за шт. Таким чином він отримав прибуток в 10 г. од. без урахування відсотків.

Загалом, зміна вартості облигацій та інших фінансових інструментів залежить від великої кількості макроекономічних аспектів, ними можуть бути: напружена геополітична ситуація, економічна криза або просто неспокійний час на фінансових ринках.

Ф'ючерс - це контракт між продавцем і покупцем, який передбачає поставку конкретного товару, акцій або послуг в майбутньому за ціною, що фіксується на момент укладення. Основна мета таких інструментів - зниження ризиків, закріплення прибутку і гарантій поставки «тут і зараз».¹

Ф'ючерсні контракти вперше з'явилися на товарному ринку. Їх суть полягає в тому, що сторони домовляються про відстрочку платежу по товару. Разом з тим, під

¹ Robert J. Shiller. Lecture on the stock market at the Yale University. URL: <https://bit.ly/3akgnjD>

час укладання такого договору, заздалегідь обмовляється ціна. Такий вид контрактів дуже зручний обом сторонам, оскільки дозволяє уникнути ситуацій, коли різкі коливання котирувань в майбутньому спровокують додаткові проблеми у встановленні ціни.

Ф'ючерси, як фінансові інструменти, користуються популярністю не тільки у тих, хто займається торгівлею різними активами, а й у спекулянтів. Вся справа в тому, що одна з різновидів цього контракту не передбачає реального постачання. Тобто договір укладається на товар, але в момент його виконання, цей товар не поставляється покупцю. У цьому ф'ючерси схожі на інші інструменти фінансових ринків, які можуть використовуватися в спекулятивних цілях.

Таким чином, ф'ючерси фактично діляться на два типи:

- Розрахункові – здійснюється лише розрахунок в грошовому еквіваленті;
- Поставні - відбувається реальна поставка;

Ф'ючерс це, по суті, контракт, який укладають дві сторони угоди: покупець і продавець. Через певний період часу, перший зобов'язується купити у другого певну кількість базового продукту, будь то акції або сировину. Таким чином, трейдери самі і є емітентами ф'ючерсів, просто біржа стандартизує укладений ними контракт і жорстко стежить за виконанням обов'язків. Такий контроль дістав назву кліринг.

На поточний момент по всіх ф'ючерсах розраховується варіаційна маржа, яка представляє собою дохід або збиток трейдера, щодо тієї ціни, за якою укладено угоду. Тобто, якщо після операції продажу ціна стала рости, то трейдер, відкривший коротку позицію, почне зазнавати збитків, а його контрагент, який купив у нього цей ф'ючерс, навпаки буде отримувати прибуткову різницю. Терміновий контракт це фактично спір, предметом якого може бути що завгодно. Так у США, наприклад, торгується ф'ючерс на погоду (рис.1.1).

Доволі схожими за своєю суттю на ф'ючерси є опціони. Фактично, це контракт, який дає право купити або продати будь-який актив за фіксованою ціною, протягом певного проміжку часу.² Головною відмінністю цих фінансових інструментів є те, що

² Lawrence Summers, Robert J. Shiller. Options Markets, Yale University lecture. 2008. URL: <https://bit.ly/39cxZ0u>

ф'ючерс представляє собою певне зобов'язання, коли як опціон – право на купівлю/продаж певного активу.

Опціони цікавлять інвесторів із двох причин: спекуляції та уникнення ризиків. Таким чином, купуючи опціон, інвестор нівелює фактор коливання ціни. А у разі зростання вартості певного виду активів, власники опціонів мають чудову можливість заробити. Це зумовлено тим, що для них вартість даного активу залишається незмінною.

Отже підсумовуючи все вище сказане, можна дійти висновку, що фондова біржа є місцем реалізації власних фінансових інтересів для всіх їх суб'єктів, будь це підприємець який шукає можливість для залучення фінансового капіталу, інвестор, який прагне вдало вкласти кошти, чи спекулянт, який заробляє на різниці купівлі/продажу активів.

Різноманіття цінних паперів не зупиняється лише на акціях, облігаціях, опціонах та ф'ючерсах. На фондовій біржі їх значно більше, адже фінансові інструменти мають безліч аналогів та антиподі. Тому кожний бажаючий має можливість обрати серед широкого спектру необхідний йому фінансовий інструмент та почати заробляти на біржі.

1.3 Процедура лістингу та суб'єкти фондової біржі

У попередньому розділі ми приділили особливу увагу фінансовим інструментам, які набули широкого застосування на фондовій біржі. Проте питання, як цінні папери потрапляють на ринок, хто виступає їх емітентами і які фінансові інститути гарантують належні умови функціонування самої біржі – залишаються відкритим.

Перш ніж цінні папери стануть доступні трейдерам на фондовій біржі, вони повинні пройти складний процес відбору – Лістинг. Фактично це процедура додавання акцій, облігацій підприємств в список інструментів, які котируються на цій платформі.

Процедура лістингу передбачає кілька етапів, які необхідно пройти компанії-емітенту, перш ніж її папери стануть доступні інвесторам:

1) Подання заяви. Зробити це може як сам емітент, так і особа, яка представляє інтереси компанії.

2) Стадія експертиз. Представники біржі аналізують як цінні папери, так і всі доступні дані про компанію. Досліджується рентабельність підприємства, ліквідність його активів. Компанія зобов'язана надати всю фінансову звітність, зібрану за останні кілька років.

3) Спеціальна комісія розглядає результати експертиз і вирішує додавати цінні папери в свій лістинг або відхилити заявку. Якщо рішення позитивне, обидві сторони - компанія і біржа, укладають угоду.

Процедура, в середньому, займає близько 1-2 місяців. Щоб залишатися в списку, емітенту потрібно, як правило, один раз в квартал, подавати необхідні дані біржі.

Важливо зазначити, що правила лістингу, що пред'являються торговими майданчиками до компаній, різняться. Одні можуть допускати до лістингу тільки підприємства, капіталізація яких становить не менше 50 мільйонів доларів, а час присутності на ринку - від 3 років. Інші можуть як підвищувати, так і знижувати планку цих та інших вимог.

Якщо виконані не всі умови, однак папери цікаві біржі, вони потрапляють в попередній список паперів - його ще називають предлістинг. Цими акціями інвестори

також можуть торгувати, проте поза стінами біржі і торгова площадка відповідальності за них не несе.

Переважно на біржах представлено кілька рівнів лістингу. У перший – преміальний, потрапляють найбільш ліквідні папери з високим рівнем надійності. До компаній другого рівня пред'являються вже не такі високі вимоги. І найнижчі для підприємств, що претендують на третій список, цінні папери якого не котируються біржою. Інвесторам, які купують папери компаній цього рівня, варто самостійно ретельно аналізувати їх благонадійність.

Наприклад, на Лондонській фондовій біржі діє Основний і Альтернативний майданчики. Перший ділиться на два рівні: «Стандарт» і «Преміум». Альтернативний ринок призначений для дрібних і середніх підприємств, що розвиваються та допускаються сюди за спрощеною процедурою (рис. 1.2).

Періодично майданчики можуть підвищувати або навпаки знижувати рівні лістингу облігацій, акцій, а то і зовсім проводити їх делістинг.

Компанії не дарма докладають зусиль, щоб потрапити в біржові лістинги, оскільки це надає їм ряд вигод. По-перше, підвищується їх інвестиційна привабливість, а в результаті легше залучати кошти для подальшого розвитку. До плюсів також віднесемо:

- зростання капіталізації підприємства зі збільшенням попиту на випущені папери;
- підвищення популярності і довіри в ділових колах;
- збільшення ліквідності акцій, облігацій;
- можливість додавання в списки зарубіжних бірж;
- в окремих випадках можна претендувати на податкові пільги.

Однак не варто забувати, що відтепер до діяльності компанії буде пред'являтися підвищену увагу, а будь-які дії будуть моментально позначатися на вартості акцій. Крім того, процедура лістингу не безкоштовна. Усі проведені експертизи коштують грошей.

Для інвесторів, процедура лістингу також відіграє важливу роль. Інвестори, вкладаючи гроші в акції компанії, знають, що її благонадійність перевірена професіоналами. Присутність в біржовому лістингу позбавляє їх від самостійних перевірок благонадійності підприємства.

Якщо цінні папери і потрапили до котирувального листа біржі, немає гарантії, що вони там залишаться назавжди. При певних умовах майданчик може викреслити їх з нього. Відбувається це в разі, наприклад, якщо емітент перестає відповідати умовам біржі, не виконав взяті зобов'язання, оголошує себе банкрутом, тощо.

Також привести до делістингу може і вартість акцій, яка не росте з часом, що свідчить про низький попит. Трапляється, що ініціатором делістингу виступає сама компанія, наприклад, при злитті декількох підприємств в одне.

Таким чином, пройшовши складний процес лістингу, компанія набуває права розміщення цінних паперів на ринку.

Важливим є той факт, що жодна операція на фондовій біржі неможлива без професійних учасників ринку - фондових посередників. Вони забезпечують зв'язок між емітентами та інвесторами і мають державні ліцензії на відповідні посередницькі види діяльності (брокерські і дилерські послуги).

Брокери - це фондові посередники, які здійснюють операції з цінними паперами за рахунок коштів клієнта. Вони отримують дохід у вигляді комісійної винагороди.

Доступ до торгів здійснюється через спеціальні сервери, і як правило, при укладанні договору з брокером, клієнтові надають торговий термінал з усіма необхідними налаштуваннями (IP-адреса, номер порту, назву сервера і т.д.). Дані налаштування дозволяють підключитися до сервера і дати можливість торгувати на біржі.

Загалом комісійні, що сплачуються брокеру є зовсім незначною сумою. Проте цей фактор стає відчутним, якщо операції купівлі/продажу відбуваються дуже часто, на незначні суми. Також, щомісячно брокеру сплачується фіксована сума за саму можливість мати доступ до торгів.

Основна роль брокерських компаній - виконувати доручення клієнта і відправляти заявки на біржу. Клієнт звертається до брокеру на основі торгових

наказів, які він дає йому на покупку або продаж цінних паперів. Такі накази іноді називаються ордерами (від англ. order).

Найпоширенішими біржовими наказами є:

- Market order - (Q);
- Limit order - (Q, P);
- Stop order - (Q, P);

Ринковий наказ (MA) - це наказ купити або продати за поточною ціною. Це означає, що брокер може на власний розсуд купувати або продавати цінні папери в залежності від того, яка ціна сьогодні складається на ринку.

Фактично, ми не називаємо ціну. Ми дізнаємося її тільки після виконання замовлення. Проте, може скластися ситуація коли ціна нас не влаштує. У такому разі нам залишається тільки прийняти це як факт, оскільки ми самі не вказали граничну ціну покупки.

Щоб уникнути таких випадків використовують Лімітний приказ (LO). З ним ми повідомляємо брокеру як кількість, так і бажану ціну. Як правило, такий цей вид наказу захищає клієнта від надлишкових втрат, якщо ціна покупки недостатньо низька, або від втрат при продажу цінних паперів. Природно, лімітний наказ обмежує можливості брокера по здійсненню угоди, але в кінцевому рахунку саме клієнт вирішує, як йому чинити в даному випадку.

Також існує так званий Стоп наказ (SO). У цьому випадку, клієнт також зазначає ціну і кількість. Але тут є певна відмінність від лімітного наказу. Наприклад, якщо ми говоримо про лімітний наказ на продаж, брокер продасть ваші акції за обумовленою ціною або вище. Але у разі стоп наказу брокер реалізує акції за вашою ціною або нижче.

Стоп наказ також називається «Stop loss» або наказом по захисту від збитків. Для прикладу розглянемо ситуацію, коли вартість наших акції становить 100\$. Ми можемо віддати наказ на «Stop lost» 80\$. У такому разі, якщо акції почнуть різко падати у ціні і досягнуть відмітки у 80\$, брокер одразу їх продасть. У такий спосіб, ми страхуємо себе від великих збитків.

Таким чином, брокер відіграє роль посередника між клієнтом і біржою. Вони значно полегшують процес торгівлі, оскільки вони повністю беруть на себе технічну сторону питання доступу на біржу.

Брокерські компанії ведуть не тільки облік операцій клієнта, що значно економить час, а й в більшості випадків є податковими агентами. Тобто самостійно сплачують податок з доходу клієнта від торгівлі на біржі, що безперечно дуже зручно. З отриманого прибутку сплачується ПДФО. У випадку, якщо рік не був прибутковий, податок не сплачується. Податок сплачується тільки з чистого прибутку.

Ще одним посередником на ринку цінних паперів виступають дилери - посередники, які працюють на біржі від свого імені і за свій рахунок. Вони можуть бути об'єднані в дилерську фірму або існувати окремо, як фізичні особи.

Дохід дилера складається з різниці цін продажу і покупки. Тому дилер повинен постійно контролювати і враховувати мінливу кон'юнктуру ринку. Зазвичай він спеціалізується на певних видах цінних паперів, але великі організації можуть обслуговувати ринок цінних паперів в цілому.

Розглянемо ці процеси дещо детальніше на конкретному прикладі. У табл. 1.1 ми бачимо гіпотетичний журнал лімітних наказів для акцій Microsoft.

Таблиця 1.1 Журнал лімітних наказів для акцій Microsoft

Shares	MPID	Bid	Shares	MPID	Ask
100	arcx	25.23	2400	NSDQ	25.24
9430	NASDQ	25.22	8200	cinn	25.24
1000	FLOW	25.22	100	HDSN	25.34
400	W	25.19	800	HDSN	25.35
350	MSCO	25.15	100	GSCO	25.36
800	UBSS	25.14	600	iseg	25.42
100	SBSH	25.12	100	HILL	25.44

Джерело: Robert J. Shiller. Lecture on the stock market at the Yale University. URL: <https://bit.ly/2w09Tr8>

Зараз це статична картинка, але на екрані цифри будуть постійно змінюватися. По суті, ми просто зафіксували один зі станів ринку.

Перші 3 колонки відносяться до запитів на покупку:

- Shares - кількість акцій, які хтось хоче купити;
- Bid - це ціна, по якій покупці хочуть купити вказану кількість акцій;
- MPID - це ідентифікатор біржі, де була подана заявка;

Наприклад, перший рядок можна прочитати так: «Купити 100 акцій Microsoft за ціною 25.23 \$, заявка залишена з біржі arcx».

Важливо зауважити, що ціни заявок представлені в порядку убутання, чим ближче до нижньої частини таблиці, тим вони менше. Система автоматично відсортувала всі невиконані накази.

На іншій стороні таблиці - пропозиції з продажу. Тут все теж саме, тільки ціни за якими хочуть продати акції відсортовані в зворотному порядку - від найдешевших до найдорожчих.

Наприклад, хтось захоче продати 2400 акцій по 25.24 \$ за шт. У нього є конкурент, який віддає свої активи за тією ж ціною, але залишив свій наказ пізніше. У таких випадках на сортування впливає не ціна, а час віддачі наказу.

Слід звернути увагу, що ціна на покупку нижче, ніж ціна на продаж. Як це розуміти? Чи означає це, що угода не відбудеться? Адже хтось хоче продати свої акції по 25.24 \$, в той час як купити хочуть тільки по 25.23 \$.

В такому випадку, акції не змінять власника, поки одна зі сторін не змінить свою ціну. Той факт, що ми бачимо ці заявки не є дивним. Як тільки ціни зійдуться, наказ буде миттєво виконаний.

Ми можемо помітити, що ціна навряд чи стане нижчою за 25.22 \$, оскільки за цією ціною є великий наказ на покупку і він не дасть ринку провалитися.

Якби ми бачили цей екран в реальному часі, всі цифри змінювалися б перед нами з нереальною швидкістю. Накази, які були актуальні навіть 20 секунд тому вже були б виконані. Тому необхідно діяти максимально швидко, щоб реалізувати свої акції за необхідною ціною.

У повністю автоматизованій системі накази виконуються автоматично. І така форма торгів зараз активно поширюється по всьому світу. Накази приймаються і закриваються комп'ютерами за частки секунди. Через це багато ордерів не встигають з'явитися на екрані.

Зараз отримала розвиток так звана «Високочастотна торгівля» (High frequency trading), яка здійснюється комп'ютерами. Коли ми торгуємо через брокерів, в залі NYSE все відбувається в спокійному ритмі. Щоб укласти угоду необхідно зателефонувати брокеру, який зробить ще один дзвінок своєму представнику в торговому залі. Він дійде до групи трейдерів, щоб поцікавитись як проходять торги і через якийсь час він закриє наш наказ.

Однак, коли в справу вступають комп'ютери, угоди укладаються миттєво. Тому виникає потреба в автоматизованих стратегіях. Ціни змінюються швидше, ніж наше око здатне сприйняти цю інформацію.

Це також називають «Алгоритмічною торгівлею» (Algorithmic trading). На сьогоднішній день ці програми стали дуже важливою частиною торгів на ринку. Вони відповідають за величезну частку укладених угод. Брокери все частіше звертаються до таких мілісекундних стратегій. На деяких біржах можливо здійснювати операції, які тривають 0.001 частку секунди.

Слід зауважити, що високочастотна торгівля ставить автоматизовані біржі в більш вигідне становище. Люди починають розуміти ці алгоритми все краще і краще, що штовхає їх до використання найсучасніших бірж. Тому традиційні біржі з торговими залами поступово вмирають по всьому світу. До речі, NYSE дуже повільно адаптувалася до нових технологій, незважаючи на їх популярність серед учасників торгів.

1.4 Види аналізу на фондових біржах

Аналіз біржової кон'юнктури – невід'ємна частина роботи на фондовій біржі. Вміння передбачати рух ціни краще, ніж інші гравці ринку, надає трейдеру величезну перевагу. Роками інвестори та дослідники були зацікавлені у розробці та тестуванні моделей поведінки цін різних фінансових інструментів. Однак, притаманні їм динамічність, нелінійність, нестационарність та хаотичність зробили цей процес надзвичайно складним – Ю. Фама, 1995.³

Згідно досліджень Д. Енке та С. Чжун⁴, на фондові ринки впливають безліч взаємопов'язаних факторів, які мають економічні, політичні, психологічні аспекти. До основних видів аналізу, які набули широкого застосування на фондових біржах, відносять технічний та фундаментальний.

Відповідно з працями Л. Су та співавторів⁵, фундаментальний аналіз передбачає перш за все комплексний підхід, сутність якого полягає у визначенні факторів впливу на цінні папери на рівні компанії, галузі та економіки в цілому. За мету ставиться прогнозування майбутніх тенденцій та отримання прибутку внаслідок передбачених коливань ціни.

Таким чином, провівши необхідні дослідження, інвестор може впевнено визначити коло емітентів, цінні папери яких представляють для нього найбільший інтерес. Не менш важливим є встановлення справедливої ціни цих фінансових інструментів. Слід зазначити, що основною передумовою цього виду аналізу є те, що в короткостроковій перспективі ціна акцій не відповідає їх дійсній вартості. Отож, маючи уявлення про справжню вартість фінансових активів, можна отримати прибуток, купуючи їх за ціною, нижчою за дійсну, або продаючи, якщо вона є дещо завищеною.

Потенційному інвестору необхідно розглянути всі частини фінансової звітності компанії, проаналізувати можливості та перспективи розвитку, зробити порівняльний

³ Fama, Eugene F. 1995. Random walks in stock market prices. *Financial Analysts Journal* 51: 75–80.

⁴ Zhong, Xiao, and David Enke. 2017. Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications* 67: 126–39

⁵ Hu, Yong, Kang Liu, Xiangzhou Zhang, Lijun Su, E. W. T. Ngai, and Mei Liu. 2015. Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Applied Soft Computing* 36: 534–51.

аналіз по рокам, а також дослідити зальну інформацію щодо сектору та стану економіки.

Ефективність фундаментального аналізу у своїх роботах дослідили Дутта А. та Гаутан Б.⁶ шляхом використання фінансових коефіцієнтів для відокремлення перспективних акцій від поганих. Автори порівняли свою річну прибутковість з Nifty (50 найбільших акцій з великою капіталізацією), який дає точність 74,6%. Це одна з небагатьох статей, в яких основна увага приділяється використанню основних характеристик (коефіцієнтів, характерних для компанії) для визначення акцій для інвестування.

Технічний аналіз ґрунтується на передумові, що всі важливі фундаментальні фактори вже відображені в ціні акцій. Відповідно, увага зосереджується лише на ціні активів та обсязі торгів, які вимірюються та подаються в таблицях і графіках. Аналізуючи ці графіки, інвестор може отримати інформацію про певні тенденції та закономірності руху цін, обсяг торгів та їх взаємозалежність.

Очікування можуть призвести до короткострокових коливань на ринку, що, в свою чергу, спричиняє розриви між ціною та справжньою вартістю акцій компанії, протягом певного періоду – Р. Шиллер 1980.⁷ Проте, незважаючи на це, моделювання поведінки ціни фінансових інструментів залишається можливим.

Прогнозування цін на фондовому ринку - доволі складний процес. Кілька теорій щодо фондових ринків були концептуалізовані за ці роки. Вони або намагаються пояснити природу фондових ринків, або чи можна їх обіграти. Однією з найбільш обговорюваних теорій, є гіпотеза ефективного ринку Ю. Фама, 1970.⁸ Згідно цієї теорії, зміна цін на цінні папери є непередбачуваною, а прогнозування фінансового ринку - безнадійна справа.

⁶ Dutta Avijan, Gautam Bandopadhyay, and Suchismita Sengupta. 2012. Prediction of Stock Performance in Indian Stock Market Using Logistic Regression. *International Journal of Business and Information* 7: 105–36.

⁷ Shiller, Robert J. 1980. Do Stock Prices Move Too Much to Be Justified by Subsequent Changes in Dividends? Cambridge: National Bureau of Economic Research.

⁸ Fama, Eugene F. 1970. Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance* 25: 383–417. doi:10.2307/2325486

Однак С. Барррас та К. Несбіт, 2004⁹ стверджували, що цінових трендів на фінансових ринках і наявність кореляції між фундаментальними подіями і економічними показниками, що впливають на ринки, є двома з багатьох доказів проти ЕМН. Дослідники і інвестори не згодні з ЕМН як в емпіричному, так і в теоретичному плані, тим самим переводячи фокус обговорення з ЕМН на поведінкові і психологічні аспекти учасників ринку.

Критика ЕМН привела до збільшення числа досліджень, які ставлять під сумнів обґрунтованість ЕМН і вводять нові і успішні підходи, які об'єднують індикатори технічного аналізу та шаблони діаграм з методологіями з економетрики, статистики, інтелектуального аналізу даних і штучного інтелекту.¹⁰

Останні досягнення в області аналізу даних та їх прогнозування поділяються на чотири категорії: статистичні, розпізнавання образів, машинне навчання (ML) і аналіз настроїв. Ці категорії в основному підпадають під більш широку категорію технічного аналізу, однак існують деякі методи машинного навчання, які також об'єднують ширші категорії технічного аналізу з підходами фундаментального аналізу для прогнозування фондових ринків.

До появи методів машинного навчання статистичні методи, які часто припускають лінійність, стаціонарність і нормальність, дозволяли аналізувати і прогнозувати запаси. Часові ряди в аналізі фондового ринку являють собою хронологічну сукупність спостережень, таких як щоденні підсумки продажів і ціни акцій - Р. Люк та ін., 2005.¹¹

Згідно Д. Енке та С. Чжун, однієї з груп статистичних підходів, що підпадають під категорію одновимірного аналізу, завдяки використанню часових рядів в якості вхідних змінних, є модель авторегресійної ковзаючої середньої (ARMA), авторегресивної інтегрованої ковзаючої середньої (ARIMA), узагальненої

⁹ Nesbitt, Keith V., and Stephen Barrass. 2004. Finding trading patterns in stock market data. *IEEE Computer Graphics and Applications* 24: 45–55

¹⁰ Arévalo, Rubén, Jorge García, Francisco Guijarro, and Alfred Peris. 2017. A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting. *Expert Systems with Applications* 81: 177–92

¹¹ Fu, Tak-chung, Fu-lai Chung, Robert Luk, and Chak-man Ng. 2005. Preventing Meaningless Stock Time Series Pattern Discovery by Changing Perceptually Important Point Detection. Paper presented at the International Conference on Fuzzy Systems and Knowledge Discovery, Changsha, China, August 27–29. Berlin/Heidelberg: Springer.

авторегресивної умовної гетероскедастичності (GARCH) і модель авторегресії плавного переходу (STAR).

Модель ARIMA широко використовується для аналізу фондового ринку. ARMA об'єднує моделі авторегресії (AR), які намагаються пояснити ефекти імпульсу і середньої реверсії, часто спостерігаються на торгових ринках, і моделі Moving Average (MA), які намагаються вловити шоківі ефекти, які спостерігаються у тимчасових рядах. Ключове обмеження моделі ARMA полягає в тому, що вона не враховує кластеризацію волатильності, ключове емпіричне явище в багатьох фінансових часових рядах. ARIMA є логічним продовженням класу моделей ARMA і може зводити нестационарні ряди до стаціонарних.

Машинне навчання широко вивчалось з точки зору його потенціалу в прогнозуванні фінансових ринків - Т. Чжан та ін., 2012.¹² Завдання машинного навчання в цілому поділяються на контрольоване і неконтрольоване навчання. У контрольованому навчанні доступний набір помічених вхідних даних для навчання алгоритму і спостережуваних вихідних даних. Метою контрольованого навчання є навчання алгоритму для автоматичного зіставлення вхідних даних з заданими вихідними даними. Після навчання машина навчилася б бачити точку вхідних даних і прогнозувати очікуваний результат. Мета неконтрольованого навчання полягає в тому, щоб навчити алгоритм знаходженню шаблонів, кореляції або кластера в даному наборі даних.¹³

Кілька алгоритмів були використані в прогнозуванні напрямку ціни акцій. Простіші методи, такі як єдине дерево рішень, дискримінантний аналіз і байесовський аналіз, були замінені більш ефективними алгоритмами, такими як випадковий ліс, логістична регресія і нейронні мережі - Н. Хеспілс та ін., 2015.¹⁴ Володіючи нелінійними, керованими даними і простими для узагальнення характеристиками, багатовимірний аналіз з використанням глибоких штучних нейронних мереж (ANN)

¹² Shen, Shunrong, Haomiao Jiang, and Tongda Zhang. 2012. Stock Market Forecasting Using Machine Learning Algorithms. Stanford: Department of Electrical Engineering, Stanford University, pp. 1–5.

¹³ Bhardwaj, Aditya, Yogendra Narayan, and Maitreyee Dutta. 2015. Sentiment analysis for Indian stock market prediction using Sensex and nifty. *Procedia Computer Science* 70: 85–91.

¹⁴ Ballings, Michel, Dirk Van den Poel, Nathalie Hespeels, and Ruben Gryp. 2015. Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications* 42: 7046–56.

став домінуючим і популярним інструментом аналізу фінансового ринку. Останнім часом глибокі нелінійні топології нейронних мереж починають привертати увагу дослідників в прогнозуванні часових рядів - Бао та ін., 2017.¹⁵

Отже, проаналізувавши теоретико-методологічні основи фондових ринків, ми можемо дійти висновку, що аналіз поведінки його інструментів є складним та комплексним процесом. Через особливості впливу внутрішніх та зовнішніх факторів, неможливо обрати універсальну методологію їх оцінки та прогнозування. Саме цим, обумовлений широкий спектр методів та підходів, кожен з яких покликаний вирішити проблему певного змісту.

¹⁵Bao, Wei, Jun Yue, and Yulei Rao. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE 12: e0180944.

РОЗДІЛ 2. МЕТОДИ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

2.1 Загальні положення Arima моделі

В аналізі та прогнозуванні часових рядів широкого застосування набула модель Бокса-Дженкінса - ARIMA(p, d, q). Фактично, це удосконалена версія ARMA моделі, основним недоліком якої є робота виключно зі стаціонарними рядами.

Модель ARIMA являє собою комбінацію моделей авторегресії (AR) та ковзної середньої (MA), значення яких позначаються як p та q відповідно. Відмінність від попередника полягає у наявності частки «I» (integrated), яка відповідає за порядок диференціювання¹⁶. Метою даного процесу є зведення ряду до стаціонарного, шляхом переходу від ряду Y_t до ряду різниць $Y_t - Y_{t-1}$. Перетворення відбуваються до моменту отримання бажаного результату, проте на практиці їх кількість не перевищує двох разів. Число ітерацій процесу диференціювання відображає змінна d.

Загальне рівняння ARIMA моделі можна представити як:

$$Y_t = c + \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \dots + \alpha_p Y_{t-p} + \varepsilon_t + \beta_0 \varepsilon_t + \beta_1 \varepsilon_{t-1} + \dots + \beta_q \varepsilon_{t-q} \quad (2.1.1)$$

де c – константа, ε_t – білий шум, $\alpha_i, i = \overline{1, p}$ та $\beta_j, j = \overline{1, q}$ – авторегресійні коефіцієнти і коефіцієнти змінного середнього відповідно.

У своїй роботі Гуджараті 2003¹⁷ описав методологію Боксу-Дженкінса в чотири етапи: ідентифікація, оцінка параметрів моделі, діагностика результатів дослідження та прогнозування.

Перший етап – ідентифікація, покликаний визначити, чи є спостережувані дані стаціонарними. Виконання даної умови є важливим з ряду причин, адже багато статистичних і економетричних методів, як AR та MA засновані на цьому припущенні¹⁸. Нестационарні часові ряди є помилковими і непередбачуваними, в той час як стаціонарність передбачає коливання навколо постійного середнього та сталу дисперсію. Слід зазначити, що ряд може бути нестационарним з різних причин:

¹⁶ E. Holmes, M. Scheuerell, and J. Ward. Applied Time Series Analysis

¹⁷ Gujarati. (2003), Econometría, México: McGraw Hill.

¹⁸ R. Shumwa, S. Stoffer. Time Series Regression and ARIMA Models. URL: <https://bit.ly/2zhKg75>

наявності тренду, сезонного компоненту, або обох одночасно. В свою чергу ряд є стаціонарним лише в одному випадку, коли виконуються умови (2.1.2):

$$\begin{aligned} 1) E(X_t^2) &< \infty \quad \forall t \in \mathbb{Z} \\ 2) EX_t &= \mu \quad \forall t \in \mathbb{Z} \\ 3) \gamma_x(s, t) &= \gamma_x(s+h, t+h) \quad \forall s, t, h \in \mathbb{Z} \end{aligned} \quad (2.1.2)$$

Іншими словами, стаціонарний часовий ряд $\{X_t, t \in \mathbb{Z}\}$ задовольняє трьом вимогам: має постійну середню та дисперсію, а функція автоковаріації залежить тільки від $(t - s)$, а не від s або t . Зверніть увагу, що у даному випадку мова йде про слабку стаціонарність.

Серед альтернативних підходів покликаних звести ряд до стаціонарного, слід також зазначити декомпозицію даних¹⁹. Дана математична процедура передбачає розбиття початкового часового ряду на три складові: тренд, сезонність та залишковий компонент. Як правило до складу тренду входить також циклічна складова, оскільки у часовому розрізі тривалість коливань циклу складає декілька років. Тому виділяти її як окрему компоненту на практиці не прийнято.

Для досягнення успішного розкладання важливо вибрати між адитивною та мультиплікативною моделями, що потребує аналізу серії (2.1.3). Цей аспект є вкрай важливим адже він характеризує взаємозв'язок між її складовими. Виконавши декомпозицію даних, стаціонарність досягається шляхом вилучення однієї або декількох компонент.

$$\begin{aligned} 1) Y_t &= S_t + T_t + R_t \\ 2) Y_t &= S_t \cdot T_t \cdot R_t \end{aligned} \quad (2.1.3)$$

Для остаточної перевірки ряду на стаціонарність, на практиці використовують тест Дікі-Фулера. В залежності від значення p-value менше/більше 0.05 робиться висновок про її присутність/відсутність.

Наступним етапом аналізу є оцінка параметрів моделі. Для визначення порядків авторегресії та ковзної середньої використовуються графіки автокореляції (ACF) та часткової автокореляції (PACF)²⁰. Фактично, вибір параметрів p та q робиться

¹⁹ Enders W. Applied Econometrics Time Series : 3nd ed. / W. Enders. – Jon Wiley and Sons, Inc

²⁰ Identifying the numbers of AR or MA terms in an ARIMA model URL: <https://bit.ly/2ZoXOZ5>

методом проб та помилок, адже графічно встановити найкращу комбінацію показників доволі складно. Тому на практиці, оцінюють одразу декілька моделей, а остаточний вибір роблять на основі критеріїв BIC та AIC.

Для оцінки параметрів моделі використовується метод максимальної правдоподібності. У випадку ARIMA функція логарифмічної правдоподібності виглядає наступним чином:

$$\ln L(\theta / y) = \sum_{t=p+1}^T \ln f(y_t / I_{t-1}, \theta) + \ln f(y_p, \dots, y_1; \theta) \quad (2.1.4)$$

де $I_t = \{y_t, \dots, y_1\}$ позначає інформацію доступну в момент часу t , а y_p, \dots, y_1 початкові значення.

При оцінці максимальної правдоподібності моделей часових рядів можна розрахувати два типи оцінок максимальної правдоподібності²¹. Перший тип заснований на максимізації першої частини рівняння - функції умовної логарифмічної правдоподібності (1). Тоді як другий заснований на максимізації повної функції логарифмічної правдоподібності (2).

$$\begin{aligned} 1) \hat{\theta}_{cmle} &= \arg \max_{\theta} \sum_{t=p+1}^T \ln f(y_t / I_{t-1}, \theta) \\ 2) \hat{\theta}_{mle} &= \arg \max_{\theta} \sum_{t=p+1}^T \ln f(y_t / I_{t-1}, \theta) + \ln f(y_p, \dots, y_1; \theta) \end{aligned} \quad (2.1.5)$$

Наступним етапом є діагностика результатів дослідження. Завдання даного кроку полягає у виборі найкращої моделі. Серед основних критерії слід виділити AIC, AICc та BIC.

Інформаційний критерій Акайке (AIC) 1973²² представляє собою математичний метод оцінки відповідності моделі даним, з яких вона була створена. У статистиці та економетриці використовується для порівняння різних моделей та визначення найкращої з них.

Даний критерій є певним компромісом між складністю моделі та її ефективністю. Оцінка AIC винагороджує моделі, які досягають високого показника придатності та штрафує їх, якщо вони стають надмірно складними.

²¹ E. Zivot Maximum Likelihood Estimation of ARMA Models URL: <https://bit.ly/3cRvYbR>

²² H. Akaike. Information theory as an extension of the maximum likelihood principle. URL: <https://bit.ly/2WNl3KG>

У випадку ARIMA моделі, рівняння AIC виглядає наступним чином:

$$AIC = -2\log(L) + 2(p + q + k + 1) \quad (2.1.6)$$

де L - оцінка максимальної правдоподібності моделі «*log-likelihood*», p та q кількість параметрів AR та MA відповідно. Для змінної k діють правила: якщо $c=0$, то $k=0$. У іншому випадку $k=I^{23}$.

Пізніше Хурвіч і Цай 1989²⁴ додатково уточнили цю оцінку, щоб скорегувати її для невеликої вибірки даних. Даний критерій слід брати до уваги, якщо кількість елементів вибірки не перевищує 40 елементів.

$$AIC_c = AIC + \frac{2(p + q + k + 1)(p + q + k + 2)}{T - p - q - k - 2} \quad (2.1.6)$$

Байєсівський інформаційний критерій (BIC) є доволі схожим за своєю суттю за AIC. У ньому також передбачена система штрафів, проте на відміну від AIC вони дещо вищі.

$$BIC = AIC + [\log(T) - 2](p + q + k + 1) \quad (2.1.7)$$

У фінальній виборі перемагає модель з найнищим значенням обраного критерію.

Для остаточної впевненості у правильності вибраної моделі необхідно перевірити модель на адекватність. Для цього ряд залишків має буди випадковою компонентною. Тест виконується шляхом порівняння величини коефіцієнта авторегресії (ACF) і часткового коефіцієнта авторегресії (PACF), отриманого із залишкової корелограми. Значення коефіцієнтів в обох випадках має бути менше за критичне²⁵.

Останнім етапом є безпосереднє прогнозування майбутніх значень ряду. У деяких випадках прогнози, отримані за допомогою ARIMA моделі, є значно кращим, у порівнянні із альтернативним підходами, особливо в короткостроковій перспективі.

²³ R. Hyndman and George Athanasopoulos Monash University, Australia URL: <https://bit.ly/2LODLeA>

²⁴ C. Hurvich, C.-L. Tsai Regression and time series model selection in small samples Biometrika, 76

²⁵ J. Shah. Basic aspects of the ARIMA model 2011.

2.2 Модель Авторегресивної умовної гетероскедастичності

Доволі популярним підходом для аналізу часових рядів в економетриці є моделі авторегресивної умовної гетероскедастичності (Autoregressive Conditional Heteroscedastic – ARCH models). В основному, вони застосовуються для моделювання волатильності фінансових інструментів²⁶.

Основний акцент робиться на рівнянні дисперсії h_t , яке враховує значення залишків попередніх періодів. Дослідити принцип роботи даної моделі можна завдяки рівнянням зазначеним нижче:

$$y_t = \phi + e_t \quad (2.2.1)$$

$$e_t / I_{t-1} \sim N(0, h_t) \quad (2.2.2)$$

$$h_t = \alpha_0 + \alpha_1 e_{t-1}^2 \quad \alpha_0 > 0, \quad 0 \leq \alpha_1 < 1 \quad (2.2.3)$$

Два останніх рівняння повністю розкривають сутність даного виду моделей. Рівняння (2.2.2) зазначає, що похибки є умовно нормальними $e_t / I_{t-1} \sim N(0, h_t)$, де I_{t-1} являє собою інформацію, доступну в момент часу $t-1$ з середнім значенням 0 і мінливу у часі дисперсію позначену як h_t . Рівняння (2.2.3) моделює h_t як функцію від константи α_0 і квадрату похибки e_{t-1}^2 .

Назва моделі – ARCH – передбачає той факт, що ми працюємо зі змінною в часі дисперсією (гетероскедастичністю), яка залежить від минулих значень ряду (автокореляція). У даному випадку ми маємо ARCH(1), дисперсія якої є функцією від константи α_0 та квадрату залишків попереднього періоду $\alpha_1 e_{t-1}^2$. Дані коефіцієнти повинні бути позитивними, щоб забезпечити позитивну дисперсію. Коефіцієнт α_1 має бути меншим за 1, оскільки в це призведе до постійного росту параметру h_t . Умовна нормальність означає, що нормальний розподіл є функцією відомої інформації в період $t-1$, для прикладу: коли $t=2$, $e_2 / I_1 \sim N(0, \alpha_0 + \alpha_1 e_1^2)$, для $t=3$, $e_3 / I_2 \sim N(0, \alpha_0 + \alpha_1 e_2^2)$ і т.д.²⁷.

²⁶ Bollerslev, T. (1986), "Generalized Autorregressive Conditional Heteroskedasticity," Journal of Econometrics, 31, 307-327.

²⁷ R. Carter Hill, William E. Griffiths, Guay C. Lim - Principles of Econometrics, 4th Edition 14.1 The ARCH Model

Для перевірки наявності ARCH ефекту використовують тест множників Лагранжа (LM). Для цього необхідно оцінити головне рівняння моделі, після чого зберегти ряд залишків e_t та отримати їх квадрати e_t^2 . Для перевірки наявності ефекту ARCH першого порядку необхідно побудувати регресію виду²⁸:

$$e_t^2 = y_0 + y_1 e_{t-1}^2 + v_t \quad (2.2.4)$$

де v_t - випадкова компонента. Нульова та альтернативна гіпотези мають вигляд:

$$H_0 : y_1 = 0 \quad H_1 : y_1 \neq 0$$

Якщо ефекту ARCH не має, то $y_1 = 0$, оцінки рівняння (2.2.4) будуть незначущими, а значення R^2 буде низьким. В протилежному випадку ми очікуємо, що e_t^2 буде напряду залежить від попередніх значень ряду, а R^2 буде відносно високим.

Статистика тесту множників Лагранжа розраховується за формулою $(T - q)R^2$, де T це величина вибірки, q - кількість e_t^2 у правій частині рівняння, R^2 - коефіцієнт детермінації. Якщо нульова гіпотеза є вірною, то статистика тесту $(T - q)R^2$ розподіляється (у великих вибірках) як $\chi_{(q)}^2$, де q це порядок лагу, а $T - q$ - кількість повних спостережень. У даному випадку $q = 1$. Якщо $(T - q)R^2 \geq \chi_{(1-\alpha, q)}^2$, тоді ми відхиляємо нульову гіпотезу та стверджуємо про наявність ARCH ефекту²⁹.

Спочатку ARCH-моделі досліджували інфляційну невизначеність. Згодом вони знайшли застосування в аналізі волатильності цін та дохідності спекулятивних активів. На основі результатів її застосування встановлено, що динаміка волатильності багатьох фінансових змінних підкоряється стійким закономірностям. В силу специфіки власної структури моделі ARCH здатні враховувати зміну

²⁸ Bera, A. K., and Higgins, M. L. (1993), "ARCH Models: Properties, Estimation and Testing," Journal of Economic Surveys, Vol. 7, No. 4, 307-366.

²⁹ Stock, J. H. (1988), "Estimating Continuous-Time Processes Subject to Time Deformation," Journal of the American Statistical Association (JASA), 83, 77-85.

діапазону коливати часового ряду та надавати якісні результати, на відміну від лінійних моделей, де дисперсія ряду h_t є константою.

2.3 Узагальнена модель GARCH та її різновиди

Модель ARCH (1) може бути розширена кількома способами. Одним з таких, є банальне внесення в модель більшої кількості лавових змінних. Загалом, для моделі ARCH (q), що містить ряд параметрів $e_{t-1}^2, \dots, e_{t-q}^2$, функція умовної дисперсії має вигляд:

$$h_t = \alpha_0 + \alpha_1 e_{t-1}^2 + \alpha_2 e_{t-1}^2 + \alpha_3 e_{t-1}^2 + \dots + \alpha_q e_{t-q}^2 \quad (2.3.1)$$

У цьому випадку, мінливість в заданому періоді залежить від величини квадратів залишків за минулі q одиниць часу.

Одним з недоліків моделі ARCH (q) є те, що існують $q+1$ параметрів для оцінки. Якщо q велике число, ми можемо втратити точність в оцінці. Узагальнена модель ARCH, або GARCH, є альтернативним способом захоплення ефектів з тривалим запізненням із меншою кількістю параметрів. Це спеціальне узагальнення моделі ARCH, і його можна вивести наступним чином. Для початку запишемо рівняння (2.3.1) як:

$$h_t = \alpha_0 + \alpha_1 e_{t-1}^2 + \alpha_2 e_{t-1}^2 + \alpha_3 e_{t-1}^2 + \dots \quad (2.3.2)$$

Іншими словами, ми наклали геометричну лаг-структуру на лагові коефіцієнти виду $\alpha_s = \alpha_1 \beta_1^{s-1}$. Надалі ми додали і відняли $\beta_1 \alpha_0$ і представили рівняння у вигляді:

$$h_t = (\alpha_0 - \beta_1 \alpha_0) + \alpha_1 e_{t-1}^2 + \beta_1 (\alpha_0 + \alpha_1 e_{t-2}^2 + \beta_1 \alpha_1 e_{t-3}^2 + \dots) \quad (2.3.3)$$

Оскільки $h_{t-1} = \alpha_0 + \alpha_1 e_{t-2}^2 + \beta_1 \alpha_1 e_{t-3}^2 + \beta_1^2 \alpha_1 e_{t-4}^2 + \dots$ ми можемо спростити до:

$$h_t = \delta + \alpha_1 e_{t-1}^2 + \beta_1 h_{t-1} \quad (2.3.4)$$

де $\delta = (\alpha_0 - \beta_1 \alpha_0)$. Ця узагальнена модель ARCH позначається як GARCH (1,1). Її можна розглядати як окремий випадок більш загальної моделі GARCH (p, q), де p - число лагових членів h, а q - кількість лагових змінних e^2 . Слід зазначити, що для

стаціонарності, нам необхідно, щоб $\alpha_1 + \beta_1 < 1$. Якщо $\alpha_1 + \beta_1 \geq 1$ ми маємо так званий «інтегрований GARCH» процес, або IGARCH³⁰.

Модель GARCH (1,1) є дуже популярною специфікацією, оскільки вона добре підходить для аналізу ряду даних. Одна з головних її переваг полягає в тому, що модель GARCH(1,1) з трьома параметрами $(\delta, \alpha_1, \beta_1)$ може мати той же ефект, як і ARCH(q) яка потребує оцінки $q+1$ параметрів, де q скажімо $q \geq 6$.

Стандартна ARCH модель оцінює «погані новини» (від'ємні $e_{t-1} < 0$) і «хороші новини» (додатні $e_{t-1} > 0$) симетрично: тобто вплив на волатильність h_t є однаковим $(\alpha_1 e_{t-1}^2)$. Однак вплив хороших і поганих новин може мати асиметричний вплив на волатильність. В цілому, коли негативні новини потрапляють на фінансовий ринок, ціни на активи, як правило, вступають в бурхливу фазу, і волатильність збільшується, але при позитивних новинах волатильність, як правило, мала, і ринок входить в період певної стабільності.

Модель порогового ARCH, або T-ARCH, є одним із прикладів, де позитивні і негативні новини розглядаються асиметрично. У версії моделі T-GARCH специфікація умовної дисперсії виглядає наступним чином:

$$h_t = \delta + \alpha_1 e_{t-1}^2 + \gamma d_{t-1} e_{t-1}^2 + \beta_1 h_{t-1}$$

$$d_t = \begin{cases} 1 & e_t < 0 & (\text{bad news}) \\ 0 & e_t \geq 0 & (\text{good news}) \end{cases} \quad (2.3.5)$$

де γ відомий як термін асиметрії.

Коли $\gamma = 0$ модель зводиться до звичайної форми GARCH. В іншому випадку, коли шок позитивний, вплив на волатильність здійснює α_1 , але коли новини негативні $\alpha_1 - \gamma$. Отже, якщо γ є значущим та позитивним, негативні шоки мають більший вплив на h_t ніж позитивні потрясіння.

³⁰ R. Carter Hill, William E. Griffiths, Guay C. Lim - Principles of Econometrics, 4th Edition 14.4.1 THE GARCH MODEL—GENERALIZED ARCH

Ще одним популярним розширенням моделі GARCH є модель «GARCH-in-mean». Позитивний взаємозв'язок між ризиком (часто вимірюється волатильністю) і прибутковістю є одним з основних принципів фінансової економіки.

У випадку збільшення ризику, зростає також і середній дохід. Інтуїтивно зрозуміло, що прибутковість ризикованих активів зазвичай вище, ніж прибутковість безпечних активів. Однак, хоча ми зазвичай і оцінюємо головне рівняння для моделювання прибутковості і разом із GARCH моделлю для визначення мінливості, що змінюється в часі, ми не використовували ризик для пояснення прибутковості. Це і покликана зробити модель «GARCH-in-mean»³¹.

Рівняння моделі «GARCH-in-mean» наведене нижче:

$$\begin{aligned} y_t &= \beta_0 + \theta h_t + e_t \\ e_t / I_{t-1} &\sim N(0, h_t) \\ h_t &= \delta + \alpha_1 e_{t-1}^2 + \beta_1 h_{t-1}, \quad \delta > 0, 0 \leq \alpha_1 < 1, 0 \leq \beta_1 < 1 \end{aligned} \quad (2.3.6)$$

Перше рівняння - це головне рівняння. Тепер воно відображає вплив умовної дисперсії на залежну змінну. Зокрема, відзначимо, що умовна дисперсія h_t впливає на y_t фактором θ . Два інших рівняння такі ж, як і раніше.

³¹ Lamoureux, G. C., and Lastrapes, W. D. (1990), "Heteroskedasticity in Stock Return Data: Volume versus GARCH Effects," *Journal of Finance*, 45, 221-229.

2.4 Модель нейронних мереж ARNN

Модель NNAR є параметричною нелінійною моделлю прогнозування.

Нелінійна авторегресійна модель порядку p , NAR (p), визначається рівнянням:

$$y_t = h(y_{t-1}, \dots, y_{t-p}) + \varepsilon_t \quad (2.4.1)$$

є прямим узагальненням лінійної моделі AR, де $h(\cdot)$ є нелінійною відомою функцією. Передбачається, що $\{\varepsilon_t\}$ є послідовністю випадкових незалежних змінних і однаково розподілена з нульовим середнім і кінцевої дисперсією σ^2 .

Авторегресійна нейронна мережа (ARNN)³², що представляє собою мережу прямого поширення, являє собою нелінійне наближення $h(\cdot)$, яке визначається як:

$$\begin{aligned} \hat{y}_t &= \hat{h}(y_{t-1}, \dots, y_{t-p}) \\ \hat{y}_t &= \beta_0 + \sum_{i=1}^I \beta_i f(\alpha_i + \sum_{j=1}^p \omega_{ij} y_{t-j}) \end{aligned} \quad (2.4.2)$$

де функція $f(\cdot)$ є функцією активації, а $\Theta = (\beta_0, \beta_1, \dots, \beta_q, \alpha_1, \dots, \alpha_q, \omega_{11}, \dots, \omega_{qn})$ вектором параметрів.

Процес реалізації NNAR моделі виконується в два етапи. На першому кроці визначається порядок авторегресії для даного часового ряду. Порядок авторегресії вказує кількість попередніх значень, від яких залежить поточне значення часового ряду. На другому етапі NN навчається за допомогою набору даних, підготовленого з урахуванням порядку авторегресії. Нейронні мережі не так просто навчати і налаштовувати. Доволі поширеним підходом є перетворення даних перед навчанням нейронної мережі, щоб уникнути помилкових результатів або тривалого часу навчання через відсутність конвергенції. NN не вимагають нормальності даних (або стаціонарності в цьому відношенні), але перетворення даних (наприклад, логарифмічна перетворення) може допомогти усунути шум, зменшити дисперсію і поліпшити сигнал при прогнозуванні часових рядів³³.

³² H. Hefny, A. Badr Forecasting of nonlinear time series using ANN. URL: <https://bit.ly/2A4wvIX>

³³ i Persio, Luca, and Oleksandr Honchar. 2017. Recurrent Neural Networks Approach to the Financial Forecast.

Кількість вхідних вузлів обумовлена порядком авторегресії, а вхідні дані для NN - це попередні, лагові спостереження. Кількість прихованих вузлів часто визначається методом проб і помилок або експериментами через відсутність будь-якої теоретичної основи для їх вибору Б. Патуво 2001³⁴.

З'єднання в NN зважуються, а ваги оптимізуються з використанням алгоритму зворотного поширення (алгоритму радієнтного спуску). Цей алгоритм ітеративно і рекурсивно оновлює ваги і зміщення, щоб мінімізувати функцію втрат яка, по суті, вимірює різницю між прогнозованими і фактичними спостерігаються значеннями.

Слід зазначити, що початкові ваги на вхідному шарі приймають випадкові значення для початку і оновлюються з використанням даних спостережень. Таким чином, в прогнозах, створюваних нейронною мережею, є елемент випадковості. Тому мережа зазвичай «навчається» кілька разів з використанням різних випадкових початкових ваг, і результати усереднюються. Кількість ітерацій має бути вибрано правильно, щоб уникнути проблеми перенавчання³⁵.

Отже, описана методологія, формує собою основу аналізу даного програмного комплексу. Модель Arima є доволі поширеним методом прогнозування часових рядів, проте через сталу дисперсію, їй не вдається врахувати зміну волатильності фінансових інструментів. Шляхом поєднання з моделями Arch, Garch ми нівелюємо дану проблему створюючи гібридну модель. У такому разі, дисперсія ряду залежить не тільки від лагових значень, а також може мати додаткові розширені форми в залежності від специфіки моделі. Окрім зазначених, вже широко відомих підходів дана методологія включає також відносно новий підхід аналізу часових рядів – модель авторегресійної нейронної мережі (ARNN). Даний підхід набуває все більшої популярності в силу простоти використання та ефективності у прогнозуванні часових рядів.

³⁴ Zhang, G. P., Patuwo, B. E. and Hu, M. Y. 2001. A simulation study of artificial neural networks for nonlinear time series forecasting. Computers & Operations Research, vol. 28, pp. 381-396.

³⁵ M. Nielsen Neural. Networks and Deep Learning URL: <https://bit.ly/3cXjSxU>

РОЗДІЛ 3. СКЛАДОВІ ПРОГРАМНОГО КОМПЛЕКСУ ТА РЕЗУЛЬТАТИ АНАЛІЗУ

3.1 Налаштування програмного комплексу

Як вже зазначалося раніше, даний програмний комплекс складається з двох програмних модулів: автоматизованого алгоритму скрепінгу даних та інтерактивної панелі управління для їх аналізу. Структурно його можна відобразити за допомогою наступної схеми (рис. 3.1):

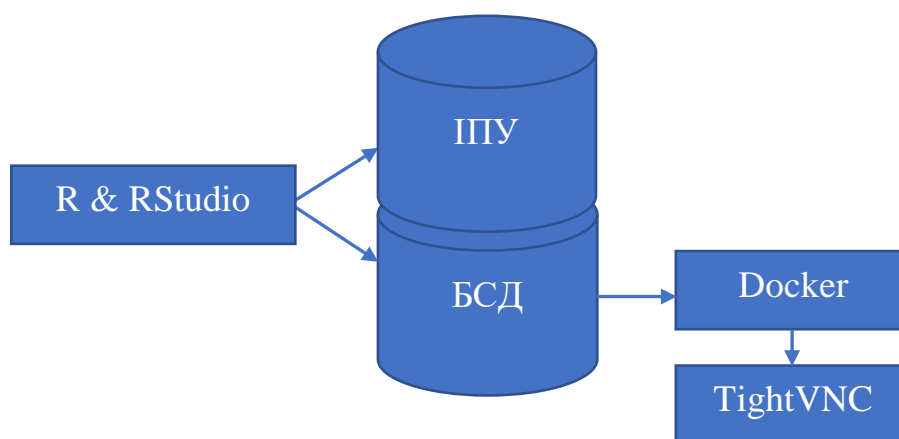


Рис. 3.1. Схема структури програмного комплексу.

Джерело: розроблено автором.

Як ми можемо бачити, окрім основних модулів, на рисунку також присутні певні програми та компоненти, наявність яких обумовлена потребами комплексу. За винятком R & RStudio, які є базовими елементами, слід акцентувати увагу на блоках «Docker» та «TightVNC».

У даній роботі, головним ресурсом забезпечення даних є платформа Yahoo! Finance, яка публікує безкоштовну інформацію інструментів фінансового ринку. Дана інформація відображається у вигляді динамічного графіку ціни, проте серед доступних датасетів для скачування, доступні лише дані з частотою в день, неділю або місяць. Таким чином, виникла необхідність у розробці 2 модулю комплексу, а саме автоматизованого алгоритму скрепінгу даних.

Web scrapping являє собою технологію сканування даних з веб-сторінок. Шляхом визначення ряду тегів та їх атрибутів, ми маємо можливість виловити наявну в них інформацію. В залежності від виду інформації: статичної або тієї, що змінюється під впливом JavaScript, web scrapping поділяється на статичний та динамічний.

У даному випадку, ми маємо справу саме з динамічним скрепінгом даних, єдиним шляхом реалізації якого в R, є використання пакету «RSelenium». Даний пакет, на відміну від інших, має особливі умови запуску. Через вірогідність виникнення помилок сумісності, його запуск став можливий лише за допомогою додаткового програмного забезпечення. Однією з таких програм є «Docker».

В силу специфіки власної структури «Docker», допомагає обійти більшість цих проблем. Тоді як програма «TightVNC» є похідним елементом, призначення якого полягає у більш зручній роботі з «Docker».

Нище зазначений технічний перелік необхідних програм та компонентів:

Таблиця 3.1. Перелік програм та компонентів.

R language	https://cran.r-project.org/bin/windows/base
R Studio	https://rstudio.com/products/rstudio/download/
Docker Desktop	https://docs.docker.com/docker-for-windows/install/
ChromeDriver	https://chromedriver.chromium.org/downloads
TightVNC	https://www.tightvnc.com/download.php

Джерело: розроблено автором.

Наступним кроком після встановлення відповідних компонентів, є робота з Docker. Фактично, Docker представляє собою засіб для упаковки, доставки і запуску додатків. Його використання дозволяє уникнути ряду помилок, які можуть виникнути при запуску програми або коду на різних комп'ютерах.

Така властивість обумовлена перш за все структурою самого Docker та його складових: «image» та «container».

«Image» представляє собою шаблон інструкцій щодо створення контейнера. До його складу входить все необхідне для запуску і роботи програми. «Container» - це працюючий екземпляр образу, місце де відбуватимуться усі операції. За своєю сутністю він добре ізольований від інших контейнерів і свого хост-комп'ютера.

Використання Docker надає розробнику велику перевагу, адже дозволяє вирішити одну з основних проблем – сумісності. Саме тому однією з можливих альтернатив запуску пакету «RSelenium» є його використання.

Для початку роботи, нам необхідно запустити Docker Desktop. Після натискання, на панелі задач з'явиться динамічний значок «whale», що символізує початок завантаження Docker. Після його переходу в статичний стан, ми маємо відкрити консоль, та переконатися що все працює нормально. Для цього достатньо ввести команду:

```
PS C:\Users\Андрей> docker info
```

Рис. 3.2. Перевірка підключення Docker

Джерело: розроблено автором.

У разі відсутності помилки, ми можемо перейти до наступних кроків.

Перш ніж створити контейнер, ми маємо завантажити відповідний «image». Для цього необхідно ввести у консолі команду:

```
PS C:\Users\Андрей> docker pull selenium/standalone-chrome-debug
```

Рис. 3.3. Завантаження образу

Джерело: розроблено автором.

За замовчуванням, для завантаження буде обрана остання версія. Проте, якщо ваша версія Chrome є дещо застарілою, то варто або оновити браузер, або знайти відповідну версію образу на Docker Hub³⁶ (це стосується і ChromeDriver).

```
PS C:\Users\Андрей> docker image ls
```

Рис. 3.4. Перевірка наявності образу

Джерело: розроблено автором.

В результаті буде відображена таблиця, що містить перелік усіх раніше завантажених образів та їх інформацію (рис. 3.5). Слід зазначити, що відмінність даного образу від аналогічного без припису «debug» полягає в тому, що ми можемо запустити VNC сервер та бачити, що відбувається в контейнері.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
selenium/standalone-chrome-debug	latest	ab22600a9c2b	4 weeks ago	939MB

Рис. 3.5. Таблиця завантажених образів.

Джерело: розроблено автором.

Наступним кроком є створення самого «container», ввівши команду:

³⁶ Джерело: Docker Hub, official website. URL: <https://hub.docker.com/r/selenium/standalone-chrome-debug/tags>

```
docker run --name malishka -v /dev/shm:/dev/shm -d -p 4445:4444 -p 5901:5900 selenium/standalone-chrome-debug
```

Рис. 3.6. Команда створення контейнера

Джерело: розроблено автором.

Як ми можемо бачити, для створення контейнеру було задіяно більшу кількість атрибутів, основний зміст яких описаний нижче:

- `--name` відповідає за назву нашого контейнера;
- `-d` означає, що контейнер Docker працює у фоновому режимі терміналу. Він не отримує введення або виведення на дисплей;
- `-p` встановлює порти (зовнішній/внутрішній); перший належить RSelenium, останній - серверу VNC;
- `selenium/standalone-chrome-debug` образ на основі якого створюється контейнер;

Для того, щоб перевірити наявність новоствореного контейнера, достатньо ввести команду:

```
PS C:\Users\Андрей> docker ps -a
```

Рис. 3.7. Перелік усіх наявних контейнерів.

Джерело: розроблено автором.

Серед наданої інформації, особливу увагу необхідно приділити полям: «CONTAINER ID», «STATUS», «PORTS» та «NAMES». Перший містить унікальне ід необхідне для звернення для контейнера. Наприклад, для запуску або вимкнення контейнера, використання команд **docker start/stop** потребує його ід. Друге поле слугує для відображення статусу контейнера (включений/вимкнутий). Третє, містить інформацію про порти, необхідну для підключення. І четверте, допомагає зорієнтуватися та знайти потрібний контейнер.

Таким чином, нам залишається лише підключити контейнер командою описаною вище.

Для подальшої роботи нам знадобиться програма TightVNC, яка дозволяє відобразити середовище новоствореного контейнера. Загальна інформація підключення зазначена у блоці Connection, нижче поля вводу (рис. 3.8.).

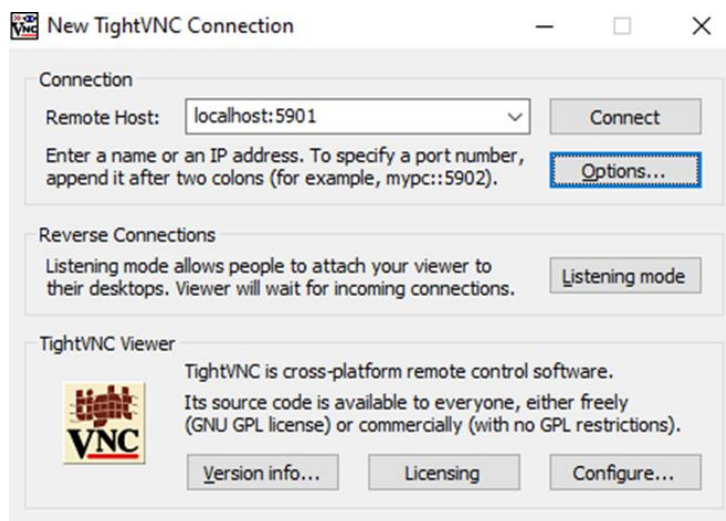


Рис. 3.8. Форма TightVNC для підключення.

Джерело: розроблено автором.

Слід зазначити, що для Docker Desktop достатньо ввести **localhost** замість IP address. Проте якщо ви використовуєте Docker Toolbox, то IP можна дізнатися за допомогою команди: **docker-machine ip**.

Щодо значення порту, то його можна побачити у полі «PORTS» ввівши команду **docker ps -a**. У даному випадку нас цікавить зовнішній порт, який ми встановлювали для VNC серверу, а саме: 5901.

STATUS	PORTS	NAMES
Up 2 seconds	0.0.0.0:4445->4444/tcp, 0.0.0.0:5901->5900/tcp	malishka

Рис. 3.9. Інформація новоствореного контейнера

Джерело: розроблено автором.

Для успішного підключення до серверу, необхідно переконатися, що контейнер включений. Перевірити це можна подивившись у поле «STATUS».

Після натискання «Connect», в залежності від умов установки, вас можливо запитають пароль. Якщо ви залишили його за замовчування, то вам необхідно ввести слово «**secret**». У іншому випадку ви маєте ввести пароль, який ви вказали на етапі встановлення.

Таким чином, виконавши необхідні налаштування програм та компонентів, все готово для використання другого модулю комплексу, а саме: автоматизованого алгоритму скрепінгу даних. Сам алгоритм детально описаний у пункті 3.2 дипломної роботи.

Завершальним етапом технічного розділу є опис роботи та функціоналу першого модулю - Інтерактивної панелі управління. Даний модуль представляє собою основну частину програмного комплексу, адже саме в ньому відбувається процеси обробки та аналізу цін інструментів фінансового ринку.

Базою його розробки слугує документ «Markdown»³⁷, який широко використовується для аналізу та формування звітності. Встановивши пакет «flexdashboard», та зробивши певні зміни в шапці документа, а саме: змінивши значення вихідного формату на «flexdashboard::flexdashboard»³⁸ та параметру «runtime:shiny», ми можемо перетворити звичайний даний документ у інтерактивну панель управління.

Умовно, її можна поділити на три складові:

- 1) Імпорт та попередню обробку часового ряду.
- 2) Оцінку найкращої Arima моделі та її комбінації з моделями Arch та Garch.
- 3) Налаштування штучної нейронної мережі ARNN та аналізу її результатів.

Запустивши даний програмний модуль, користувач потрапляє на початкову сторінку «Page 1». До її складу входить бічна панель, яка містить в собі елементи імпорту даних, й основне тіло сторінки, де відображаються результати пошуку у вигляді графіку та відповідної таблиці. Також туди входить панель перетворень призначена для попередньої обробки даних часового ряду.

Даний модуль передбачає два шляхи імпорту: з використанням власного файлу або завантаження фінансової інформації з платформи Yahoo! Finance, вказавши тікер компанії та відповідний йому часовий проміжок. Слід зазначити, що надмірну від автоматизованого алгоритму описаного далі (пункт 3.2), частота імпортованих даних становить один день.

В ході роботи передбачається використання лише одного із запропонованих варіантів імпорту. У разів одночасного їх застосування, значення віджетів будуть оновлені та приведені початковий вид. При необхідності змінити спосіб імпорту,

³⁷ Y. Xie, J. Allaire, G.Grolemund. R Markdown: The Definitive Guide. URL: <https://bit.ly/2ytd502>

³⁸ K. Hefny. Flexdashboard: interactive dashboards for R. URL: <https://bit.ly/2ZDDGmt>

достатньо скористатися кнопкою «Reset», для очищення інформації. Для подальшої роботи з програмою слід натиснути кнопку «Submit».

Після перевірки правильності введеної інформації, результати імпорту автоматично відобразяться у блоці основної частини.

Особливу увагу необхідно приділити графічній складовій у блоці «Time series plot». Усі наявні там елементи є взаємозалежним. Користувач має можливість змінювати: часовий період в межах раніше вказаного, а також саме значення ряду. Необхідність останнього атрибуту обумовлена тим, що в цілому фінансова інформація зберігаються у форматі «OHLC», або «Open-High-Low-Close» а тому виникає потреба забезпечити можливість перегляду кожного з них. Алгоритм побудований таким чином, що до списку потрапляють виключно ряди зі числовими значеннями. Таким чином, зайва інформація автоматично ігнорується.

Сам графік є інтерактивним та багатофункціональним. Такі властивості обумовлені поєднанням пакетів «ggplot2»³⁹ та «plotly»⁴⁰.

Нище знаходиться блок «Data transformation panel», який відповідає за попередню обробку даних. Серед наявних варіантів вибору:

- Перевірка ряду на «Outlier» (наявність раптових, одиночних виплесків може значною мірою вплинути на якість оцінки моделей. Для їх усунення використовується функція «tsclean»⁴¹ пакету «forecast», шляхом знаходження середньої арифметичної між сусідніми значеннями ряду);
- Застосування методології Бокса-Кокса, логарифмування (може значною мірою зменшити дисперсію вихідного ряду, а у випадку Arima моделі звести мультиплікативну модель до адитивної);
- Округлити значення обраного ряду (іноді фінансова інформація публікується без чіткого формату)

³⁹ H. Wickham, W. Chang. Package «ggplot2» pdf manual. URL: <https://bit.ly/3d07P38>

⁴⁰ C.Sievert, C. Parmer. Package «plotly» pdf manual. URL: <https://bit.ly/2XwdOX0>

⁴¹ R. Hyndman, G. Athanasopoulos. Package «forecast» pdf manual. URL: <https://bit.ly/3ecazKQ>

Також дана секція включає в себе список «Series to analyze» де користувач має обрати тип ряду для подальшого аналізу. Слід зазначити, що даний список не корелює зі списком в блоці «Time series plot».

Останні блоком цієї сторінки є «Datatable» де відображаються початковий стан імпортованих даних. Дана інтерактивна таблиця дозволяє здійснювати пошук як в окремому стовпці, так і в цілому. Ці властивості є вкрай корисним в поєднанні з інтерактивним графіком.

Після виконання всіх необхідних трансформацій користувач може перейти на наступну сторінку для початку аналізу. Сторінка «ARIMA/ARCH/GARCH» призначена для знаходження найкращої ARIMA моделі та її поєднання з моделями Arch та Garch (у випадку з garch мова йде про базову модель та її розширення як TGARCH та Garch-in-Mean). Для початку аналізу необхідно натиснути лише кнопку «Analyze» після чого комплекс виконає усі розрахунки, а результати аналізу з'являться у відповідних блоках сторінки. Слід зазначити, що кожен блок супроводжується відповідними правилами і критеріями необхідних для оцінки результатів та їх інтерпретації. У підсумку, оцінка кожної з моделей супроводжується набором тестів, графіків і таблиць.

Аналогічну структуру має сторінка «ARNN» де методом аналізу виступає Авторегресійна модель нейронних мереж.

3.2 Опис роботи алгоритму скрепінгу даних

Даний алгоритм був розроблений з метою вилучення даних фінансових інструментів з частотою в 1 хв, їх накопичення та використання у подальшому аналізі. В даному розділі ми зупинимося на основних аспектах його роботи, надамо детальне пояснення чим був обумовлений вибір кожного підходу.

Перше, на що слід звернути увагу, це на структуру коду. Відкривши «document outline» (рис. 3.10.) ми бачимо, що код розбитий на окремі блоки. Це значно спрощує процес переміщення всередині файлу та надає уявлення про роль кожного з них.

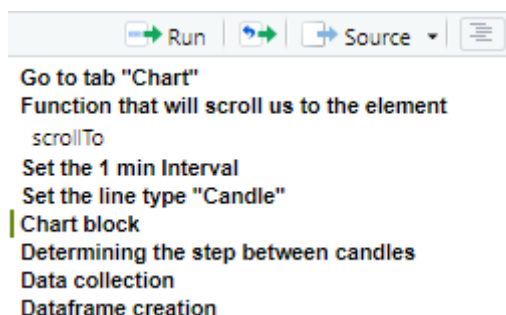


Рис. 3.10. Структура коду

Джерело: розроблено автором.

Для підключення до нашого контейнера ми використовуємо функцію **remoteDriver** куди вносимо необхідну інформацію. Визначившись з фінансовим інструментом, ми маємо присвоїти відповідний тикер змінній «symb». З його допомогою ми формуємо посилання, яке використовуємо для пошуку у новоствореному браузері (рис. 3.11.).

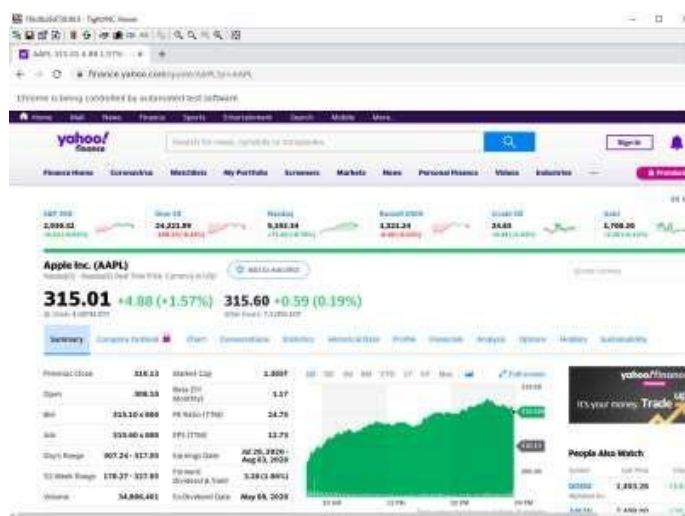


Рис. 3.11. Середовище контейнера після виконання початкових рядків коду.

Джерело: розроблено автором.

Зверніть увагу, що у верхній частині вікна з'явився напис: «Chrome is being controlled by automated test software». Незважаючи на те, що всі дії виконуються виключно скриптом, ми все одно можемо вносити свої зміни у вікні TightVNC. Проте це вкрай не рекомендовано, адже може призвести до виникнення ряду помилок під час виконання коду.

Наступним кроком є перехід на вкладку «Chart» для відображення динамічного графіку ціни. Алгоритм пошуканий виконати дану роботу знаходиться у блоці «Go to tab Chart».

В цілому, метод пошуку елементів є доволі одноманітним. Використовуючи комбінацію клавіш Ctrl+Shift+I для відображення коду сторінки й Ctrl+Shift+C для виділення необхідного елемента та його місця в коді, можна легко встановити набір атрибутів, які допоможуть до нього звернутися (рис. 3.12.)

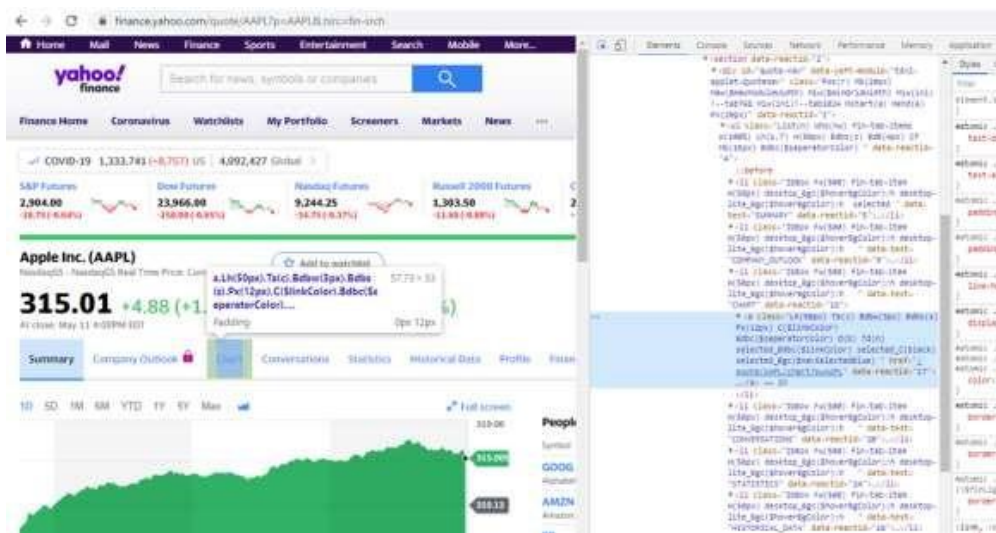


Рис. 3.12. Принцип пошуку елементів сторінки.

Джерело: розроблено автором.

Для пошуку елемента ми використовуємо функцію **findElements**, вказавши тип локатору та відповідний йому запис. Оскільки усі вкладки мають однаковий клас, то результатом пошуку буде список із n елементів. Для того, щоб серед інших знайти необхідну, нам треба провести фільтрацію.

У нашому випадку, після формування списку на основі локатора `xpath`, ми проводимо відбір спираючись на текст заголовку. Таким чином, отримавши індекс

потрібної нам вкладки, ми підставляємо його у відповідне місце в коді та проводимо натискання (рис. 3.13.)

```
Tabs <- remDr$findElements(using = "xpath", "//a[starts-with(@class, 'Lh(50px)')]")
ChartIndex <- Tabs %>% sapply(., function(x) x$getElementText()) %>% grep1("^chart$",.) %>% which
Tabs[[ChartIndex]]$clickElement()
```

Рис. 3.13. Код блоку «Go to tab Chart».

Джерело: розроблено автором.

Аналогічним чином відбувається пошук усіх наступних елементів.

Доволі цікавим за своєю суттю є блок «Function». Після переходу на необхідну вкладку, частина графіку буде знаходитися за межами екрану. Для того, щоб спостерігати за виконанням коду стало зручно, була створена спеціальна функція в основі якої лежить JavaScript. Таким чином, при виконанні даного блоку коду, фокус сторінки спрямовується на динамічний графік інструменту.

Після встановлення частоти даних в 1 хвилину та зміни типу графіку на «Candle» ми переходимо до основної частини коду, а саме збирання інформації. Доцільним є зазначити ряд передумов, які зіграли вирішальну роль у виборі даного підходу.

Показники, які представляють для нас найбільший інтерес знаходиться у блоці, поява якого обумовлена переміщенням курсору в середині графіку (рис. 3.14.). Вилучення даних сетау як такого неможливе, адже зміна значень відбувається під дією скрипта. Таким чином, єдиним способом збору даних є поетапне встановлення курсору на місці кожної з свічок.



Рис. 3.14. Блок фінансової інформації

Джерело: розроблено автором.

Досліджуючи принцип роботи графіку, вдалося встановити місце в коді, де відображається положення свічок відносно його лівого краю. Таким чином, порівнюючи показники двох сусідніх, ми можемо встановити крок. Слід пам'ятати, що дана відстань автоматично оновлюється в залежності від обраного масштабу.

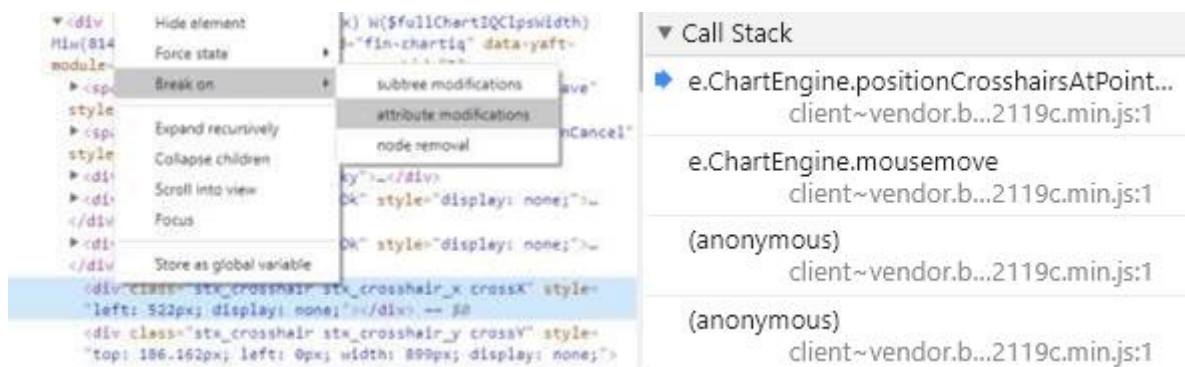


Рис. 3.15. Знаходження скрипту

Джерело: розроблено автором.

Використовуючи опцію **Break on** → **attribute modifications**, ми визначили скрипт, який відповідає за зміну стилю (рис. 3.13.). Проте через велику кількість вхідних даних та складність розрахунків, був придуманий алгоритм в обхід.

Сутність полягає в тому, що атрибут «left» відповідає за положення конкретної свічки. Зміна відбувається лише в тому випадку, коли курсор перетинає умовний екватор із однією з сусідніх. Таким чином, поетапно рухаючи курсор з кроком в 1 піксель, ми робимо запис та рахуємо кількість пікселів необхідних для переходу на сусідній бар.

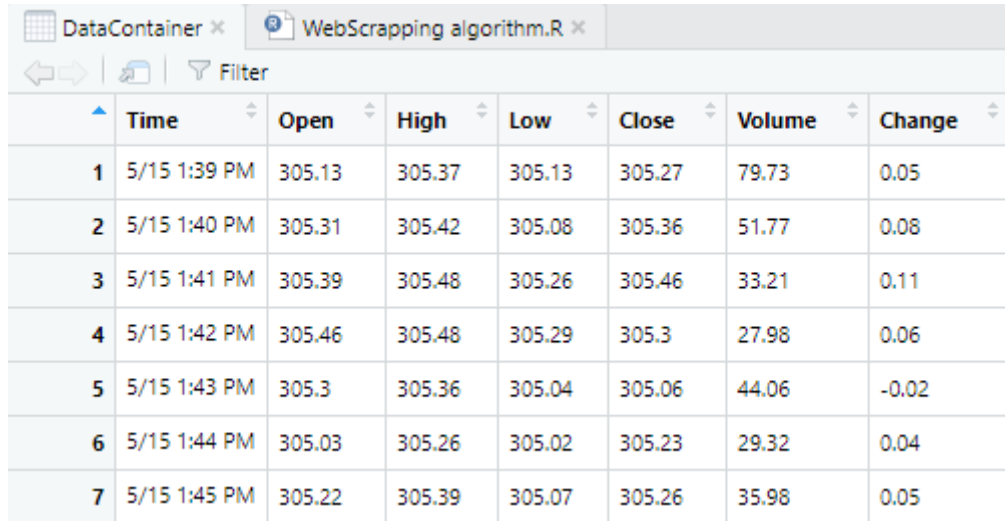
Зібравши дані параметрів самого графіку у блоці «Chart block», ми переходимо до визначення кроку. Для цього умовно курсор переноситься до лівого краю графіку та поступово рухається зі зсувом в 1 піксель. Кожна зміна положення супроводжується записом значення атрибута «left». Подолавши відстань у 40 пікселів, ми трансформуємо дані списку у таблицю, на основі якої знаходимо моду – наш крок (рис. 3.16.). Слід зазначити, даний підхід дозволяє визначити крок в незалежності від масштабу: початкового або зміненого користувачем.

-4px	12px	20px	28px	36px	4px
1	8	8	8	8	8

Рис. 3.16. Форматування таблиці із значень списку

Джерело: розроблено автором.

Таким чином, встановивши відстань між свічками, ми можемо приступити до формування дата сету. Для цього ми знову встановлюємо курсор у лівій частині графіку та вилучаємо інформацію, поступово змінюючи положення на величину кроку.



	Time	Open	High	Low	Close	Volume	Change
1	5/15 1:39 PM	305.13	305.37	305.13	305.27	79.73	0.05
2	5/15 1:40 PM	305.31	305.42	305.08	305.36	51.77	0.08
3	5/15 1:41 PM	305.39	305.48	305.26	305.46	33.21	0.11
4	5/15 1:42 PM	305.46	305.48	305.29	305.3	27.98	0.06
5	5/15 1:43 PM	305.3	305.36	305.04	305.06	44.06	-0.02
6	5/15 1:44 PM	305.03	305.26	305.02	305.23	29.32	0.04
7	5/15 1:45 PM	305.22	305.39	305.07	305.26	35.98	0.05

Рис. 3.17. Результат роботи коду.

Джерело: розроблено автором.

3.3 Моделювання та прогнозування цін акцій

Результатом роботи комплексу є – автоматизований, детально викладений, поетапний аналіз біржових інструментів. Першим кроком у побудові ARMA-GARCH/ARCH моделі є визначення оптимальної ARIMA. Для проведення аналізу ми скористаємося даними ціни акцій компанії MSFT за період «2018-03-17 - 2020-01-25» (рис. 3.18.)

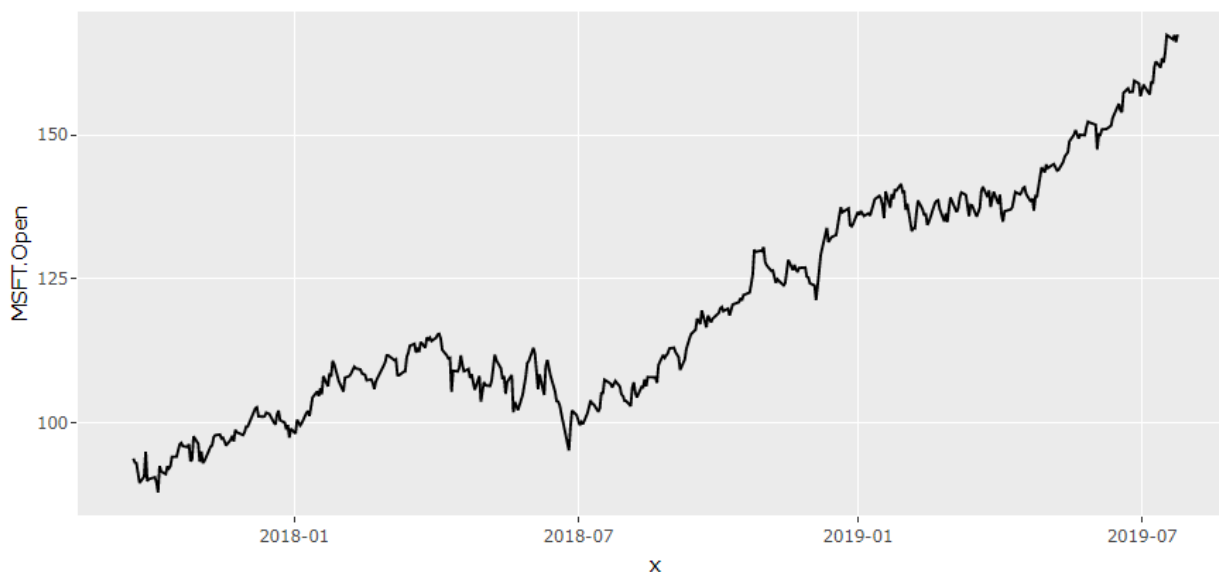


Рис. 3.18. Часовий ряд «Open» компанії MSFT з 2018-03-17 до 2020-01-25.

Джерело: розроблено автором. URL: <https://yhoo.it/3dvpqQK>

Доволі поширеною є практика перетворення ряду цін шляхом логарифмування. Це пов'язано з тим, що більшість інвесторів цікавить саме прибутковість акцій, а не їх вартість. Також, логарифмування може зменшити дисперсію ряду, що позитивно відобразиться на оцінці моделей. Таким чином, ми отримуємо ряд відсоткових змін акцій.

Як ми можемо бачити, ряд не є стаціонарними. Для впевненості перевіримо це за допомогою тесту Діккі-Фуллера (рис. 3.19.):

Augmented Dickey-Fuller Test

```
data: TSeries
Dickey-Fuller = -1.811, Lag order = 7, p-value = 0.6579
alternative hypothesis: stationary
```

Рис. 3.19. Тест Діккі-Фуллера для логарифмованого ряду.

Джерело: розроблено автором.

Значення p-value більше за 0,05 , що свідчить про присутність нестационарності. Спробуємо позбутися її шляхом диференціювання. Запровадивши диференціювання першого порядку, ми знову перевіряємо ряд на стаціонарність (рис. 3.20.).

```
Augmented Dickey-Fuller Test

data: mData$Tseries
Dickey-Fuller = -8.5678, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

Рис. 3.20. Тест Діккі-Фуллера для диференційованого ряду.

Джерело: розроблено автором.

Виходячи з зазначених результатів ($p\text{-value} < 0,05$), ряд став стаціонарним. Графічне зображення ряду відображене нижче (рис. 3.21.).

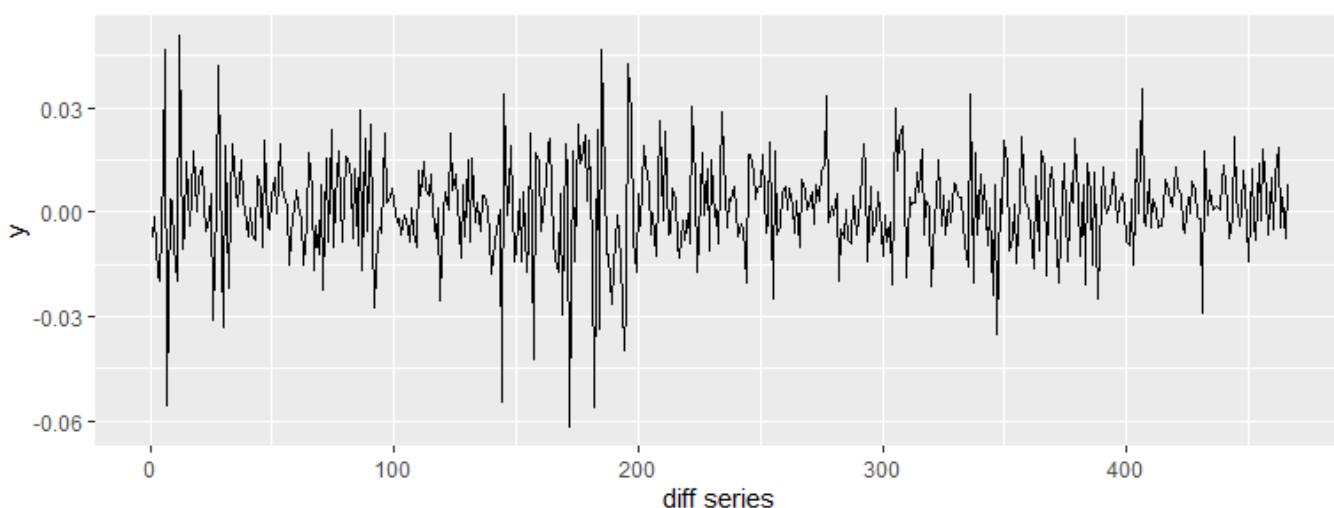


Рис. 3.21. Вигляд ряду після набуття стаціонарності.

Джерело: розроблено автором.

Наступним кроком є знаходження параметрів ARMA(p, q). Для цього розглянемо графіки ACF та PACF (рис. 3.22). Як ми можемо бачити, ми маємо значущі лаги 1, 2, 9 на ACF, та 1,9,18 на графіку PACF. Зважаючи на отримані результати доцільно буде припустити наступну специфіку моделі - ARMA(1,2).

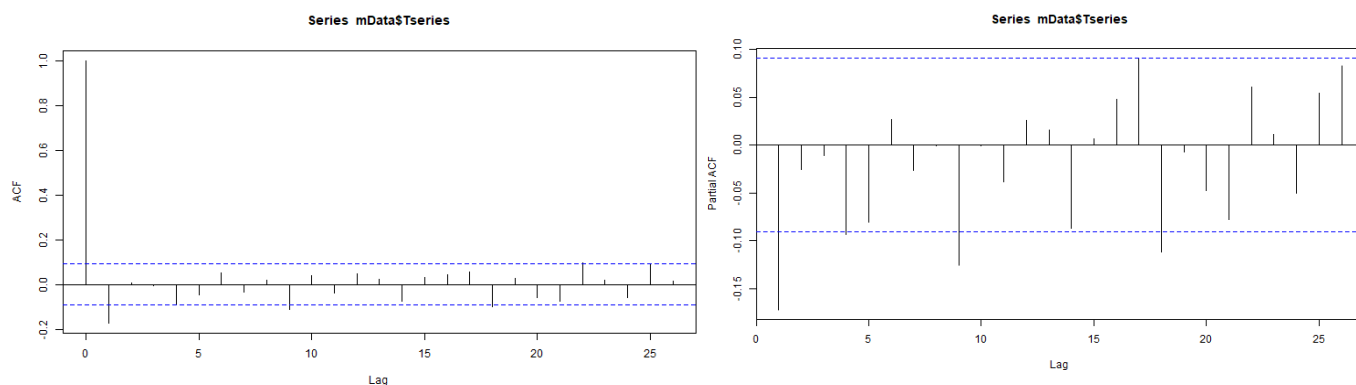


Рис. 3.22. Графіки автокореляційної та частково автокореляційної функцій.

Джерело: розроблено автором.

Проте незважаючи на весь потенціал цього методу, доволі складно вручну вибрати порядок параметрів $ARMA(p, q)$, яка буде добре працювати при прогнозуванні набору даних. Для вирішення цієї проблеми чудово підійде функція `auto.arima` з пакету `forecast`.

Дана функція передбачає можливість знаходження оптимальної ARIMA спираючись на відповідний критерій. У нашому випадку це критерій AIC (Інформаційний критерій Акаїке).

```
Series: mData$Tseries
ARIMA(1,1,2) with drift

Coefficients:
      ar1      ma1      ma2      drift
      0.8131 -0.9964  0.1007  0.0013
s.e.  0.0956   0.1051  0.0560  0.0004

sigma^2 estimated as 0.0001975:  log likelihood=1328.26
AIC=-2646.51  AICc=-2646.38  BIC=-2625.79
```

Рис. 3.23. Результати функції `auto.arima` для логарифмованого ряду.

Джерело: розроблено автором.

Спираючись на отримані результати ми можемо стверджувати, що модель $ARIMA(1, 1, 2)$ – є оптимальною (рис. 3.23.).

Для остаточної впевненості у правильності вибраної моделі перевіримо її на адекватність. Модель є адекватною, якщо ряд залишків є випадковою компонентою, тобто ACF залишків не повинна істотно відрізнятися від нуля.

Для цього скористаємося критерієм Льюнга-Бокса (*Ljung-Box test*):

```
Box-Ljung test

data: residuals(mData$arima)
X-squared = 11.445, df = 12, p-value = 0.4912
```

Рис. 3.24. Результати тесту Льюнга-Бокса для ARIMA(1, 1, 2).

Джерело: розроблено автором.

Як видно з результатів тесту (рис. 3.24.) , ми не відхиляємо нульову гіпотезу, що данні є білим шумом ($p\text{-value} > 0,05$). Аналогічного висновку можна дійти, подивившись на графік залишків, ACF та PACF (рис. 3.25.)

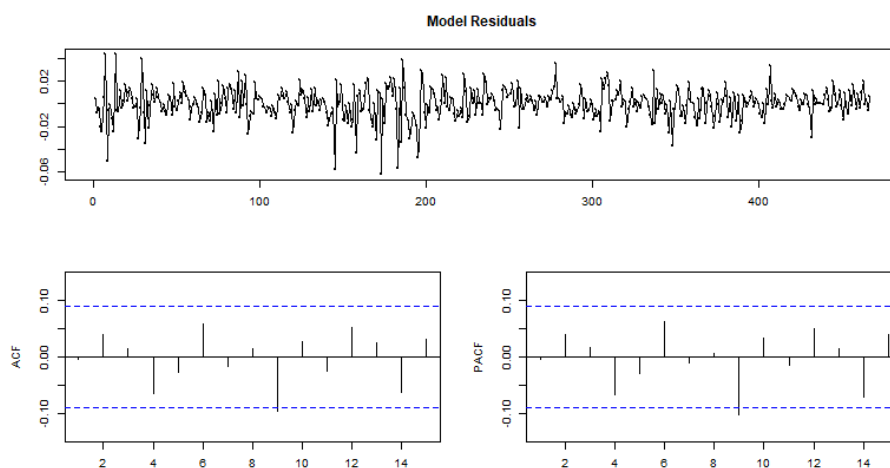


Рис. 3.25. Графік залишків, ACF та PACF для ARIMA(1, 1, 2).

Джерело: розроблено автором.

Останнім кроком у ході аналізу моделі Arima є проведення прогнозу. Оскільки побудована Arima(1, 1, 2) для прогнозування базується на минулих двох значеннях ряду, то доцільним буде здійснити прогноз на 2 періоди вперед, для забезпечення його точності та якості. Відповідні результати зазначені нижче:

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
	All	All	All	All	All
468	167.376351050212	164.389038105182	170.417950088369	162.829293750673	172.050386300764
469	167.422580483018	163.574632819688	171.361047690636	161.573582634677	173.483313289959

Рис. 3.26. Прогноз Arima(1,1,2) на наступні 2 періоди

Джерело: розроблено автором.

Наступним кроком у побудові гібридної моделі ARMA-GARCH/ARCH є визначення найкращої Arch моделі. Для цього необхідно перевірити ряд на наявність Arch ефекту, присутність якого, є обов'язковою умовою їх застосування.

Строгий білий шум не може бути передбачений ні лінійно, ні нелінійно, в той час як загальний білий шум не може бути передбачений лише лінійно. Якщо залишки представляють собою строгий білий шум, вони незалежні з нульовим середнім, нормально розподілені, і ACF & PACF квадратів залишків не покажуть значних лагів (рис. 3.27.).

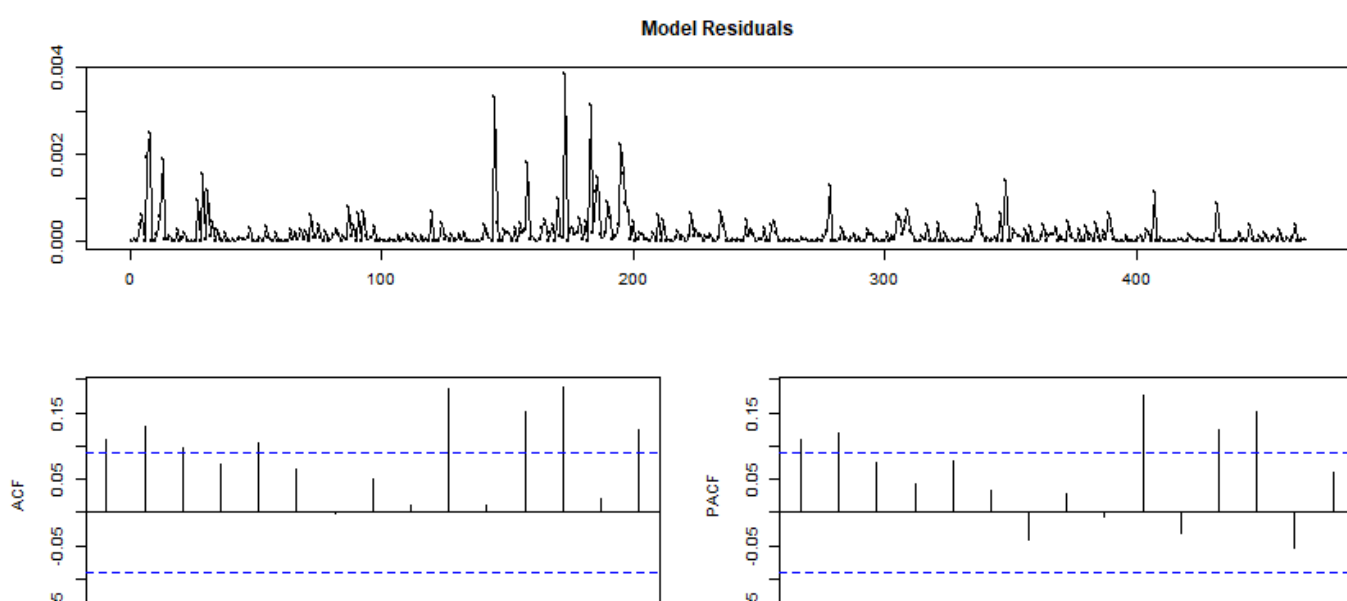


Рис. 3.27. ACF та PACF для квадратів залишків.

Джерело: розроблено автором.

В якості альтернативи тесту Енгеля для ARCH (LM test), ми можемо перевірити послідовну залежність (наявність ефекту ARCH) у ряді залишків, виконавши Q-тест Льюнга-Боксу для перших m лагів квадратів залишків.

Кількість лагів $m = P + Q$, де GARCH(P, Q), ARCH($P * Q$). Оскільки ми ще не знаємо значення параметрів моделей, ми виконаємо перевірку для деякого ряду випадків (рис. 3.28.):

	lag 1	lag 2	lag 3	lag 4	lag 5
1	0.0174560737114994	0.00113572959277564	0.000421821932154898	0.000386964956790803	0.000100342857325963

Рис. 3.28 Q-тест Льюнга-Боксу для перших m лагів квадратів залишків

Джерело: розроблено автором.

Оскільки значення p -value в усіх 5 випадках менше за 5%, ми можемо стверджувати про наявність ARCH ефекту.

Наступним кроком є знаходження найкращої ARMA-ARCH моделі. Нижче зображені результати регресії для ARCH(1)-ARCH(4) відповідно (рис. 3.29):

	AIC	BIC	SIC	HQIC
ARIMA-Arch(1)	-5.69713646824028	-5.64377785492536	-5.69746244283585	-5.67613630834475
ARIMA-Arch(2)	-5.75134593211469	-5.68909421658061	-5.75178838079333	-5.7268457455699
ARIMA-Arch(3)	-5.76039709879406	-5.68925228104083	-5.7609733828176	-5.73239688560001
ARIMA-Arch(4)	-5.76786318545435	-5.68782526548196	-5.76859052230542	-5.73636294561105

Рис. 3.29. Результати інформаційних критеріїв для моделей ARIMA-ARCH

Джерело: розроблено автором.

Виходячи з наведених результатів, найкращою є модель Arima-Arch(4), оскільки їй відповідає найменше значення критерію AIC -5.7678. Проте, з урахуванням значимості оцінених коефіцієнтів, ми робимо вибір у сторону моделі Arima-Arch(2). Відповідні результати зображені нижче:

```

Error Analysis:
      Estimate Std. Error t value Pr(>|t|)
mu      4.176e-04  1.495e-04   2.793 0.005224 **
ar1      7.523e-01  7.293e-02  10.316 < 2e-16 ***
ma1     -8.900e-01  8.757e-02 -10.162 < 2e-16 ***
ma2      2.136e-02  5.682e-02   0.376 0.707041
omega    1.079e-04  1.434e-05   7.527 5.17e-14 ***
alpha1    1.043e-01  4.944e-02   2.110 0.034817 *
alpha2    4.441e-01  1.269e-01   3.499 0.000466 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Рис. 3.30. Результати оцінки моделі Arima-Arch(2)

Джерело: розроблено автором.

Error Analysis:				
	Estimate	Std. Error	t value	Pr(> t)
mu	4.575e-04	1.519e-04	3.012	0.002591 **
ar1	7.420e-01	7.187e-02	10.324	< 2e-16 ***
ma1	-9.314e-01	8.696e-02	-10.711	< 2e-16 ***
ma2	4.878e-02	5.874e-02	0.831	0.406244
omega	8.480e-05	1.305e-05	6.496	8.24e-11 ***
alpha1	6.612e-02	4.659e-02	1.419	0.155848
alpha2	3.703e-01	1.078e-01	3.434	0.000595 ***
alpha3	1.244e-01	5.922e-02	2.101	0.035678 *
alpha4	8.707e-02	6.043e-02	1.441	0.149652

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				

Рис. 3.31. Результати оцінки моделі Arima-Arch(4)

Джерело: розроблено автором.

Після вибору найкращої моделі ARMA-ARCH, наступним кроком є побудова прогнозу. Відповідні результати зазначені нижче:

ARMA-ARCH(2) ⚡	
T+1	167.518311477315
T+2	167.619385111179

Рис. 3.32. Прогноз Arima-Arch(2) моделі на наступні 2 періоди.

Джерело: розроблено автором.

Наступним етапом є знаходження оптимальної ARMA-GARCH моделі. Значною перевагою GARCH моделі перед ARCH є те, що умовна дисперсія залежить також від минулих значень самої дисперсії. Таким чином частково нівелюється проблема умови коефіцієнтів, оскільки GARCH(P, Q) може дати не гірші результати за ARCH (P + Q). Виходячи з цього припущення розглянемо 4 основні моделі GARCH: (1, 1); (1, 2); (2,1); (2,2). Виходячи із результатів, зображених на рис. 3.33. найкращою моделлю є Arima-Garch(2,2).

	ARIMA-Garch(1,1) ⚡	ARIMA-Garch(1,2) ⚡	ARIMA-Garch(2,1) ⚡	ARIMA-Garch(2,2) ⚡
Akaike	-5.81474700750352	-5.81045523644269	-5.81751580724442	-5.81755533576004
Bayes	-5.74360218975029	-5.7304173164703	-5.73747788727204	-5.7286243135685
Shibata	-5.81532329152707	-5.81118257329376	-5.8182431440955	-5.81845080092081
Hannan-Quinn	-5.78674679430948	-5.77895499659939	-5.78601556740112	-5.78255506926748

Рис. 3.33. Результати інформаційних критеріїв для моделей ARIMA-GARCH

Джерело: розроблено автором.

Проте, зважаючи на значимість коефіцієнтів, модель Arima-Garch(1,1) є найкращою з представлених (рис. 3.34).

GARCH Model	: sGARCH(1,1)			
Mean Model	: ARFIMA(1,0,2)			
Distribution	: std			
Optimal Parameters				

	Estimate	Std. Error	t value	Pr(> t)
mu	0.001705	0.000278	6.1321	0.000000
ar1	0.819439	0.067199	12.1941	0.000000
ma1	-0.975701	0.060783	-16.0523	0.000000
ma2	0.067113	0.035641	1.8830	0.059695
omega	0.000009	0.000001	6.5283	0.000000
alpha1	0.074133	0.013044	5.6831	0.000000
beta1	0.876857	0.024487	35.8098	0.000000
shape	5.590065	1.265308	4.4179	0.000010

Рис. 3.34. Найкраща Arima-Garch модель

Джерело: розроблено автором.

Після знаходження оптимальної Arma-Garch, наступним кроком є побудова прогнозу. Відповідні результати зазначені нижче:

ARMA-sGARCH(1,1) ♦	
T+1	167.480954968141
T+2	167.583778857824

Рис. 3.35. Прогноз Arima-Garch(1,1) моделі на наступні 2 періоди.

Джерело: розроблено автором.

Одним з розширенням стандартної моделі Garch є модель TGarch, основним припущенням якої є наявність асиметричного ефекту на волатильність, обумовленого значенням збурення e_{t-1} .

В цілому, коли негативні новини потрапляють на фінансовий ринок, ціни на активи, як правило, вступають в бурхливу фазу, і волатильність збільшується, але при позитивних новинах волатильність, як правило, невелика, і ринок вступає в період стабільності.

Нище представлені результати аналізу моделей Arma-TGarch для раніше вказаних порядків (Рис) На основі наведених результатів, найкращою моделлю є Arma-TGarch(1,1) із значенням AIC -5.8322.

	ARIMA-TGarch(1,1) ♦	ARIMA-TGarch(1,2) ♦	ARIMA-TGarch(2,1) ♦	ARIMA-TGarch(2,2) ♦
Akaike	-5.83226483550039	-5.82797295892301	-5.82388292620953	-5.81945874677401
Bayes	-5.75222691552801	-5.73904193673147	-5.72605880179884	-5.71274152014416
Shibata	-5.83299217235147	-5.82886842408379	-5.82496345491121	-5.82074113571773
Hannan-Quinn	-5.80076459565709	-5.79297269243046	-5.78538263306772	-5.77745842698294

Рис. 3.36. Результати інформаційних критеріїв для моделей ARIMA-TGARCH

Джерело: розроблено автором.

На етапі оцінки параметрів, дана модель також показала найкращі результати (рис. 3.37).

GARCH Model	: fgARCH(1,1)			
fgARCH Sub-Model	: TGARCH			
Mean Model	: ARFIMA(1,0,2)			
Distribution	: std			
Optimal Parameters				

	Estimate	Std. Error	t value	Pr(> t)
mu	0.001538	0.000306	5.0311	0.000000
ar1	0.850697	0.077411	10.9894	0.000000
ma1	-0.988048	0.066624	-14.8302	0.000000
ma2	0.066807	0.031016	2.1540	0.031243
omega	0.001378	0.000851	1.6199	0.105252
alpha1	0.093899	0.051802	1.8126	0.069886
beta1	0.825076	0.092485	8.9212	0.000000
eta11	0.995803	0.497676	2.0009	0.045402
shape	5.484563	1.335159	4.1078	0.000040

Рис. 3.37. Найкраща Arima-TGarch модель

Джерело: розроблено автором.

Останнім етапом у роботі з Arma-TGarch є побудова прогнозу. Відповідні результати для наступних двох періодів зазначені нижче:

ARMA-TGARCH(1,1) ♦	
T+1	167.457241198459
T+2	167.551841222922

Рис. 3.38. Прогноз Arima-TGarch(1,1) моделі на наступні 2 періоди.

Джерело: розроблено автором.

Ще одним популярним розширенням моделі GARCH є модель «GARCH-in-mean», де акцент робиться на позитивному взаємозв'язку між ризиком (який часто вимірюється волатильністю) і прибутковістю фінансових інструментів.

Однак, хоча ми оцінили середнє рівняння для моделювання прибутковості і оцінили модель GARCH для визначення мінливості, що змінюється в часі, ми не використали ризик для пояснення прибутковості. Саме це і покликана зробити модель «GARCH-in-mean».

У ході дослідження були отримані результати оцінки критеріїв гібридних моделей, де найкращою стала модель Arima-MGarch(1,1), із значенням AIC -5.8010.

	ARIMA-MGarch(1,1) ⚡	ARIMA-MGarch(1,2) ⚡	ARIMA-MGarch(2,1) ⚡	ARIMA-MGarch(2,2) ⚡
Akaike	-5.80104550600618	-5.79658413469207	-5.79226307332246	-5.78846125680901
Bayes	-5.70322138159549	-5.68986690806223	-5.67665274447346	-5.66395782574086
Shibata	-5.80212603470786	-5.79786652363579	-5.79376398237211	-5.79019721065661
Hannan-Quinn	-5.76254521286437	-5.75458381490101	-5.74676272688214	-5.73946088371943

Рис. 3.39. Результати інформаційних критеріїв для моделей ARIMA-MGARCH

Джерело: розроблено автором.

Такий же висновок був отриманий на основі оцінок параметрів моделей. Результат найкращої моделі зазначений нижче (рис.3.40.):

```

GARCH Model      : fGARCH(1,1)
fGARCH Sub-Model   : APARCH
Mean Model        : ARFIMA(1,0,2)
Distribution       : std

Optimal Parameters
-----
      Estimate Std. Error  t value Pr(>|t|)
mu      0.001472   0.001340   1.098796 0.271857
ar1     -0.305621   0.877974  -0.348098 0.727767
ma1      0.161584   0.875653   0.184530 0.853598
ma2     -0.014941   0.145308  -0.102825 0.918102
archm    0.343703   7.709999   0.044579 0.964443
omega    0.000000   0.000001   0.105437 0.916029
alpha1   0.014001   0.011789   1.187684 0.234958
beta1    0.922739   0.075545  12.214361 0.000000
eta11    0.546782   0.407789   1.340844 0.179971
lambda   2.826618   0.270276  10.458271 0.000000
shape    5.965129   1.680047   3.550573 0.000384

```

Рис. 3.40. Найкраща Arima-MGarch модель

Джерело: розроблено автором.

Після знаходження оптимальної Arma-MGarch, наступним кроком є побудова прогнозу. Відповідні результати зазначені нижче:

ARMA-MGARCH(1,1) ♦	
T+1	167.593286377338
T+2	167.885555097156

Рис. 3.41. Прогноз Arima-MGarch(1,1) моделі на наступні 2 періоди.

Джерело: розроблено автором.

На даному етапі аналізу ми використали вже доволі знайомі та поширені методи моделювання часових рядів. Кожен з них має свої особливості будови та умови, дотримання яких є необхідним для їх ефективного застосування. Проте, наразі все більшої популярності набувають нові методики, які не ґрунтуються на базових аспектах, як стаціонарність, лінійність, нормальність, тощо, а використовують принципово нові алгоритми аналізу. Серед таких методів є модель Авторегресійної нейронної мережі (ARNN).

Для побудови ARNN ми використовуємо функцію *nnetar* з пакету *forecast*. У нашому випадку, кількість ітерацій дорівнює 20 – оптимальне значення, аби уникнути недо/пере-навчання.

```
Series: netData$Series
Model:  NNAR(1,1)
Call:   nnetar(y = netData$Series, repeats = 20, lambda = 0.5)

Average of 20 networks, each of which is
a 1-1-1 network with 4 weights
options were - linear output units

sigma^2 estimated as 4.306e-05
```

Рис. 3.42. Результати оцінки ARNN моделі.

Джерело: розроблено автором.

Слід зазначити, що процес реалізації даної моделі виконується у два етапи. На першому визначається порядок авторегресії (вибір якого обумовлений критерієм АІС для моделі AR)⁴², а на другому відбувається безпосереднє навчання нейронної мережі.

В основі алгоритму лежить метод градієнтного спуску, основна мета якого – мінімізувати функцію втрат (вимірює різницю між прогнозованими та фактичними спостереженнями).

Досягається це шляхом оновлення значень компонентів моделі – вагів та зміщень, які початково вибираються довільно. Таким чином, в прогнозах, створених нейронною мережею є елемент випадковості. Саме тому, ми зазначили кількість у 20 ітерацій (процесів навчання), фінальним результатом яких є їх середнє значення.

Нище наведені результати прогнозу моделі на наступні 4 періоди.

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
468	167.26555208512	164.345681920601	170.403363302923	162.705612383243	172.322622079739
469	167.033961005754	162.760189756905	171.149659305173	160.387842553875	174.26190535068
470	166.814389776169	161.625153037543	171.829172978364	158.877715763505	174.575438971782
471	166.606070557013	160.706329453258	172.144228524346	158.131690443297	175.256868828968

Рис. 3.43. Результати прогнозу на наступні 4 періоди.

Джерело: розроблено автором.

А також, графічне відображення прогнозу:

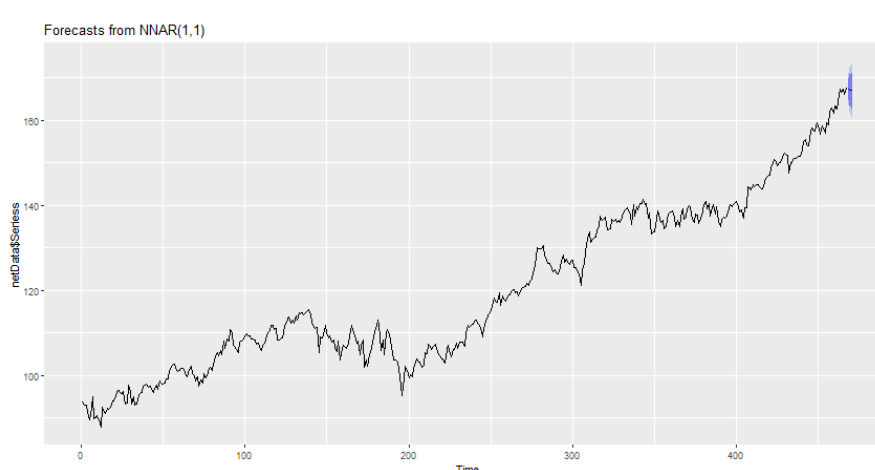


Рис. 3.44. Графічне відображення прогнозу моделі ARNN.

Джерело: розроблено автором.

⁴² R. Hyndman, G. Athanasopoulos. 11.3 Neural network models URL: <https://bit.ly/2ZIBH09>

Виходячи з результатів аналізу різних моделей, ми можемо підвести фінальний підсумок, оцінивши ефективність вибраних моделей та порівнявши якість їх прогнозів із фактичними даними.

У ході аналізу основу методології склали наступні моделі: лінійна модель Arima й її гібридне поєднання з моделями авторегресивної умовної гетероскедастичності Arch та Garch (й їх різновидів), а також моделі нейронних мереж ARNN.

Виходячи із результатів проведеного дослідження ми можемо порівняти їх ефективність моделювання обраного ряду на основі спільного критерію – AIC. Відповідні значення відображені в таблиці нижче:

Таблиця 3.2. Інформаційний критерій AIC для оцінених моделей

	AIC	AIC (rugarch)
Arima(1, 2)	-2646.51	-5.6792
Arma(1, 2) - Arch(2)	-2684,30	-5.7603
Arma(1, 2) - Garch(1, 1)	-2709,65	-5,8147
Arma(1, 2)-TGarch(1, 1)	-2717,81	-5,8322
Arma(1, 2)-MGarch(1, 1)	-2703,27	-5,8010

Джерело: розроблено автором.

Слід зазначити, що для побудови моделей Garch використовувався пакет rugarch, де формула AIC дещо видозмінена (фінальний результат поділений на кількість періодів у вибірці). Цим обумовлена необхідність створення двох колонок AIC, для можливості інтерпретації результатів та їх порівняння.

Згідно наведених даних, модель Arma(1,2)-TGarch(1,1) є найкращою з представлених. Їй належить найменше значення AIC у -2717,81 одиниць. Проте, фактичне порівняння з моделлю нейронних мереж ARNN неможливе, в силу відсутності цієї оцінки у моделі.

Альтернативним способом перевірки ефективності наведених моделей є порівняння якості їх прогнозу із реальними даними. Дані результати наведені нижче:

Таблиця 3.3. Інформаційний критерій АІС для оцінених моделей

	T+1	T+2
Arima(1, 2)	167,38	167,42
Arma(1, 2) - Arch(2)	167,52	167,62
Arma(1, 2) - Garch(1, 1)	167,48	167,58
Arma(1, 2)-TGarch(1, 1)	167,46	167,55
Arma(1, 2)-MGarch(1, 1)	167,59	167,88
ARNN	167,26	167,03
Фактичне значення ряду	167.54	167,91

Джерело: розроблено автором.

Головною метою будь-якого алгоритму моделювання даних є знаходження максимально наближених до фактичних результатів, шляхом визначення ряду прихованих закономірностей всередині самих даних. Виходячи з наведених значень, ми можемо констатувати, що для даного часового проміжку, найкраще з прогнозом впоралась модель Arma(1, 2)-MGarch(1, 1). Проте, в цілому усі моделі показали доволі близькі до реальних результати. Виходячи із розрахунків критеріїв АІС, модель Arma(1, 2)-TGarch(1, 1) здатна краще виявити дані закономірності у довгостроковій перспективі. А тому, для більш якісного моделювання даного часового ряду є доцільним спиратися на результати декількох моделей.

Отже, виходячи з інформації зазначеної в розділі, ми можемо стверджувати, що даний програмний комплекс представляє собою набір програмних і апаратних рішень, для самостійного функціонування і моделювання цін фінансових інструментів. Кожен з модулів, відіграє важливу роль у ході аналізу: бот – дозволяє вилучити по хвилині дані фінансових інструментів, інтерактивна модель управління – обробити та проаналізувати часовий ряд біржових індексів. Його результатом є поетапний, автоматизований аналіз, приклад якого був описаний вище. Таким чином, ми маємо можливість швидко та якісно проводити комплексний аналіз, що надає безумовну перевагу серед біржових гравців.

ВИСНОВКИ

В умовах глобалізації та стрімкого розвитку комп'ютерних технологій виникає потреба швидкого реагування на зміни, які несуть в собі фінансові ринки. Поява нових інструментів та збільшення об'єму інформації змушують учасників торгів шукати нові методи дослідження біржової кон'юнктури.

Обробка та комплексний аналіз великих масивів даних стали першочерговим завданням біржових гравців. Таким чином, виникла потреба у створенні автоматизованих комплексів для моделювання цін фінансових інструментів.

Згідно завдань дипломної роботи:

- були проаналізовані теоретико-методологічні основи функціонування фондових ринків, досліджені особливості поведінки різних фінансових інструментів, а також роботи вчених та дослідників стосовно класичних та відносно нових підходів їх аналізу.
- обрана методологія, як формує собою основу аналізу розробленого програмного комплексу. До її складу увійшли: лінійна модель Arima, її гібридна комбінація з моделями Arch, Garch, а також авторегресійна модель нейронних мереж ARNN.
- був розроблений автоматизований алгоритм скрепінгу по хвилинних даних, необхідний для аналізу цін біржових індексів. Даний бот формує собою перший програмний модуль комплексу та здатний вилучити інформацію будь-якого інструменту з платформи Yahoo! Finance, яка публікує безкоштовну фінансову інформацію.
- розроблена інтерактивна панель управління, мета якої полягає у попередній обробці та аналізі часових рядів – біржових індексів. Сам процес аналізу - автоматизований та представлений у вигляді поетапного відображення результатів оцінок моделей, вибору найкращої з них та прогнозування майбутніх значень ряду.
- проведений аналіз результатів комплексу. Можемо констатувати, що для акцій компанії MSFT для часового проміжку у період з 2018-03-17 по 2020-01-25, найкраще з прогнозом впоралась модель Arma(1, 2)-MGarch(1, 1). Проте, в

цілому усі моделі показали доволі близькі до реальних результати. Згідно розрахунків критеріїв AIC, модель Arma(1,2)-TGarch(1,1) здатна краще виявити приховані в даних закономірності у довгостроковій перспективі. А тому, для більш якісного моделювання часового ряду є доцільним спиратися на результати одразу декількох моделей.

Головною перевагою розробленого програмного комплексу є його цілісність і самодостатність. Він являє собою комплексний набір програмних і апаратних рішень, необхідних для самостійного функціонування і моделювання цін фінансових інструментів.

Практична цінність даного комплексу обумовлена можливістю швидкого та якісного аналізу, а також створення більш комплексного програмного забезпечення на його основі. Нище зазначений перелік його можливих поліпшень:

- Розширення існуючої методології та поглиблення в аналіз;
- Налаштування роботи бота на збирання даних з частотою менше ніж одна хвилина;
- Можливість одночасного скрепінгу та моделювання біржових індексів в режимі реального часу;
- Написання програмного коду бота для автоматизації процесу торгівлі.

У порівнянні з існуючими аналогами, відсутність прямих ір адрес даних, робить проблематичним аналіз показників біржових індексів у режимі реального часу. Проте наявність автоматизованого алгоритму скрепінгу даних – першого програмного модулю, частково нівелює цю проблему.

В цілому, комплекс заслуговує уваги, адже його фактична модернізація зазначена вище, дозволить не тільки досягнути функціоналу аналогів, а й значно його перевершити.

Актуальність даної теми обумовлена прагненням інвесторів та дослідників досліджувати біржову кон'юнктуру, з метою написання програм та алгоритмів, їх застосування для заробітку. Адже фондовий ринок, як і фінансовий ринок в цілому складають значну частину світової економіки. Тому вміння написати власний програмний комплекс є першим кроком для досягнення цієї мети.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ставицький А.В., Ніколайчук С.А. Застосування GARCH-моделей для прогнозування волатильності ПФТС-індексу. Вісник КНУ: Економіка. – №75-76. – с. 98-102. URL: <https://bit.ly/2TR087X>
2. Ставицький А.В. Нові напрями прогнозування. Вісник КНУ: Економіка. - 2000. - №44. - С. 67-71.
3. Черняк О.І., Ставицький А.В., Баженова О.В., Шебаніна О.В. Економетрика: підручник. 2-ге вид., перероб. та доп. - За ред. О.І.Черняка. –Миколаїв: МНАУ, 2014. – 414 с.
4. Базилевич В.Д., Шелудько В.М., Вірченко О.В., Ставицький А.В. та ін. Фондовий ринок: підручник. Книга 2. За ред. В.Д. Базилевича. – К.: Знання, 2016. – 686 с.
5. Ляшенко О.І., Кравець Т.В., Хрущ Л.З. Застосування пакетів прикладних програм в економетричному моделюванні фінансових часових рядів // Економіко-математичне моделювання соціально-економічної систем. – 2017. – Випуск 22. – С. 5-32. URL: <https://bit.ly/2XXUQc0>
6. Engle R. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation/ R. Engle // Econometrica. – 1982. - Vol. 50, Is. 4. – P. 987-1008.
7. Clifford H. “Best Linear Forecast vs. Best Optimal Forecast.” New York University. New York, NY. URL: <https://bit.ly/2QoS4rS>
8. Pedersen J. H. ARMA(1,1)-GARCH(1,1) Estimation and forecast using rugarch 1.2-2 URL: <https://bit.ly/2wjxgch>
9. Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. Journal of Statistical Software, 27(1), 1–22 URL: <https://bit.ly/2W36hkq>
10. Identifying the numbers of AR or MA terms in an ARIMA model URL: <http://people.duke.edu/~rnau/411arim3.htm>
11. Abugaber D. Using ARIMA for Time Series Analysis URL: <https://bit.ly/2MdJT3a>

12. Nau K., Hansen B. "Stationary and Differencing." Duke University. Durham, NC. n.d
13. Jong R. "Stationary Time Series." Ohio State University. Columbus, OH. n.d.
14. Christoffersen P., Hahn J., and Inoue A. Testing and comparing value-at-risk measures. *Journal of Empirical Finance*, 8(3):325–342, 2012.
15. Engle R.F. and Bollerslev T. Modelling the persistence of conditional variances. *Econometric Reviews*, 5(1):1–50, 1986.
16. Dickey D. A. Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root / D. A. Dickey // *Econometrica*. – 1981. – № 49. – P. 1057– 1072.
17. Enders W. *Applied Econometrics Time Series* : 3rd ed. / W. Enders. – Jon Wiley and Sons, Inc
18. Hill R. C., Griffiths W. E., Guay C. *Principles of econometrics* / Lim.—4th ed.
19. Frances P.H. *Time Series Models for Business and Economic Forecasting* / Philip Hans Frances. – Cambridge University Press
20. Bollerslev T. Generalized Autoregressive Conditional Heteroscedasticity / T. Bollerslev // *Journal of Econometrics*. – 1986
21. Davidson R. Several Tests for Model specification in the Presence of Alternative Hypothesis / R. Davidson, J. MacKinnon // *Econometrica*. – 1981.
22. Aas K. and Haff I.H. The generalized hyperbolic skew student's t-distribution. *Journal of Financial Econometrics*, 4(2):275–309, 2012.
23. Bollerslev T. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986.
24. Conrad C. and Berthold R.H. Inequality constraints in the fractionally integrated garch model. *Journal of Financial Econometrics*, 4(3):413–449, 2013.
25. Hansen B.E. Autoregressive conditional density estimation. *International Economic Review*, 35:705–730, 1994.
26. Ghalanos. A. Introduction to the rugarch package. URL: <https://bit.ly/2F5H9kS>
27. Prvulovic. D. VaR with GARCH(1,1) package. URL: <https://bit.ly/2X7CuDF>
28. Shiller R.J. Lecture on the stock market at the Yale University. URL: <https://bit.ly/3akgnjD>

29. Summers L., Shiller R.J. Options Markets, Yale University lecture. 2008. URL: <https://bit.ly/39cxZ0u>
30. Eugene F. 1995. Random walks in stock market prices. *Financial Analysts Journal* 51: 75–80.
31. Zhong I., Xiao L., and Enke D. 2017. Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications* 67: 126–39
32. Hu E., Yong W., Liu K., Zhang X., Su L., Ngai M., and Liu M. 2015. Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Applied Soft Computing* 36: 534–51
33. Avijan D., Bandopadhyay G., and Sengupta S. 2012. Prediction of Stock Performance in Indian Stock Market Using Logistic Regression. *International Journal of Business and Information* 7: 105–36.
34. Shiller, R. J. 1980. Do Stock Prices Move Too Much to Be Justified by Subsequent Changes in Dividends? Cambridge: National Bureau of Economic Research.
35. Fama F., Eugene F. 1970. Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance* 25: 383–417.
36. Nesbitt V., Keith V., and Stephen Barrass. 2004. Finding trading patterns in stock market data. *IEEE Computer Graphics and Applications* 24: 45–55
37. Arévalo K., Rubén I., García J., Guijarro F., and Peris A. 2017. A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting. *Expert Systems with Applications* 81: 177–92
38. Fu Y., Chung F., Luk R. 2005. Preventing Meaningless Stock Time Series Pattern Discovery by Changing Perceptually Important Point Detection. Paper presented at the International Conference on Fuzzy Systems and Knowledge Discovery, Changsha, China, August 27–29. Berlin/Heidelberg: Springer.
39. Shen U., Shunrong D., Haomiao J., and Tongda Z. 2012. Stock Market Forecasting Using Machine Learning Algorithms. Stanford: Department of Electrical Engineering, Stanford University, pp. 1–5.

40. Bhardwa J., Adity A., Narayan Y., and Dutta M. 2015. Sentiment analysis for Indian stock market prediction using Sensex and nifty. *Procedia Computer Science* 70: 85–91.
41. Ballings T., Michel P., Poel D., Hespeels N., and Gryp. R. 2015. Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications* 42: 7046–56.
42. Bao Y., Wei V., Yue J., and Rao Y. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE* 12: e0180944.
43. Michel P. Identifying the numbers of AR or MA terms in an ARIMA model URL: <https://bit.ly/2ZoXOZ5>
44. Shumwa R., Stoffer S. Time Series Regression and ARIMA Models. URL: <https://bit.ly/2zhKg75>
45. Zivot E. Maximum Likelihood Estimation of ARMA Models URL: <https://bit.ly/3cRvYbR>
46. Akaike. H. Information theory as an extension of the maximum likelihood principle. URL: <https://bit.ly/2WNI3KG>
47. Athanasopoulos R.H., Athanasopoulos G. Monash University, Australia. URL: <https://bit.ly/2LODLeA>
48. Akaike, H. (1973) Information theory and an extension of the maximum likelihood principle. *Proc. 2nd International Symposium on Information Theory*, In: B N Petrov and F Csaki (Eds.) *Akademiai Kiado, Budapest*, pp 267–81.
49. Hughes, A. W. (2015) Selection of order of an ARCH model, *Economics Letters*, 83, 269-275.
50. Brooks C. and Burke, S.P. (2014) Information Criteria for GARCH model selection, *The European Journal of Finance*, 9, 557-580.
51. Glosten L., Jagannathan R. and Runkle D. (2018) On the relation between the expected value and the volatility of nominal excess return on stocks, *Journal of Finance*, 48, 1779-1801.

52. Song Y., Jae L. (2018), Augmenting Recurrent Neural Networks with High-Order User-Contextual Preference for Session-Based Recommendation. URL: <https://arxiv.org/pdf/1805.02983.pdf>
53. Ioffe S., Szegedy C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning
54. He X. and Chua T. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In Proceedings of the 40th International ACM Special Interest Group on Information Retrieval (SIGIR). pages 355–364.

ДОДАТКИ

Додаток А

CME Group							
TRADING CLEARING REGULATION DATA TECHNOLOGY EDUCATION ABOUT							
<div>Exchange: CME</div> <div>Asset Class: WEATHER</div> <div>Product: K2-K2 CHICAGO CDD FUTURES</div> <div>Download Data</div>							
Exchange	Asset Class	Product	Product Code	Start Period	End Period	Maintenance	Maint. Vol. Scan
CME	WEATHER	K2 CHICAGO CDD FUTURES	K2	04/2021	10/2021	82 USD	30.000%
CME	WEATHER	K2 CHICAGO CDD FUTURES	K2	04/2019	09/2019	76 USD	30.000%
CME	WEATHER	K2 CHICAGO CDD FUTURES	K2	04/2020	07/2020	609 USD	30.000%
CME	WEATHER	K2 CHICAGO CDD FUTURES	K2	08/2020	08/2020	546 USD	30.000%
CME	WEATHER	K2 CHICAGO CDD FUTURES	K2	10/2020	10/2020	151 USD	30.000%
CME	WEATHER	K2 CHICAGO CDD FUTURES	K2	10/2019	10/2019	128 USD	30.000%
CME	WEATHER	K2 CHICAGO CDD FUTURES	K2	09/2020	09/2020	1,027 USD	30.000%
About This Report							

Рис. 1.1. Ф'ючерси на погоду на Чиказькій товарній біржі (CME)

Джерело: Chicago Mercantile Exchange, official website. URL: <https://bit.ly/2TLlbKU>

RECENT ISSUES		Research Recent Issues	
Company name	Market	Expected first day of trading	Expected money raised
Applegreen plc ("Applegreen" or the "Company")	AIM	25/10/2018	156.3m
CareTech Holdings PLC ("CareTech" or the "Company")	AIM	19/10/2018	nil
Shearwater Group plc	AIM	17/10/2018	Primary:16.7m Secondary:0.9m under open offer.

Рис. 1.2. Альтернативний інвестиційний ринок «AIM» на Лондонській фондовій біржі

Джерело: London Stock Exchange, official website. URL: <https://bit.ly/2Jrg9hV>

Додаток В

Код автоматизованого алгоритму скрепінгу даних:

```
library(RSelenium)
library(magrittr)
library(readr)

remDr <- remoteDriver(
  remoteServerAddr = "localhost",
  port = 4445L,
  browserName = "chrome"
)

symb="AAPL"

remDr$open()
remDr$maxWindowSize()

link<-paste0("https://finance.yahoo.com/quote/", symb,"?p=",symb)

remDr$navigate(link)

#Go to tab "Chart" ----
Tabs <- remDr$findElements(using = "xpath", "//a[starts-with(@class, 'Lh(50px)')]")

ChartIndex <- Tabs %>% sapply(., function(x) x$getElementText()) %>%
grepl("^Chart$",.) %>% which

Tabs[[ChartIndex]]$clickElement()

Sys.sleep(10)
```

#Function that will scroll us to the element ----

```
scrollTo <- function(remDr, webElem){
  remDr$executeScript("arguments[0].scrollIntoView(false);", args = list(webElem))
  webElem$highlightElement()
}
```

```
Focus <- remDr$findElement(using = "xpath", "//div[starts-with(@class, 'stx-holder stx-
panel-chart')]")
```

```
scrollTo(remDr, Focus)
```

#Set the 1 min Interval ----

```
BoxInstr <- remDr$findElements(using = "xpath", "//div[@class = 'Pos(r) D(ib) O(n):f
Cur(p)']")
```

```
IntervalIndex <- BoxInstr %>% supply(., function(x) x$getElementText()) %>%
grepl("Interval",.) %>% which
```

```
BoxInstr[[IntervalIndex]]$clickElement()
```

```
#C($c-fuji-grey-j) Lh(15px) Fw(400) Fz(s) Py(8px) Px(20px) W(100%) Ta(start)
```

```
InterChoice <- remDr$findElements(using = "xpath", "//button[starts-with(@class, 'C($c-
fuji-grey-j)')]")
```

```
IntervalIndex <- InterChoice %>% supply(., function(x) x$getElementText()) %>%
grepl("^1 min$",.) %>% which
```



```
InterChoice[[IntervalIndex]]$clickElement()
```

```
#Set the line type "Candle" ----
```

```
LineIndex <- BoxInstr %>% sapply(., function(x) x$getElementText()) %>%  
grepl("Line",.) %>% which
```

```
BoxInstr[[LineIndex]]$clickElement()
```

```
#C($c-fuji-grey-j) Lh(15px) Fw(400) Fz(s) Py(8px) Px(22px) W(100%) Ta(start)
```

```
LineChoice <- remDr$findElements(using = "xpath", "//button[starts-with(@class, 'C($c-  
fuji-grey-j)'))")
```

```
LineIndex<-LineChoice %>% sapply(., function(x) x$getElementText()) %>%  
grepl("^Candle$", .) %>% which
```

```
LineChoice[[LineIndex]]$clickElement()
```

```
# Chart block ----
```

```
ChartBox <- remDr$findElement(using = "class", "stx-subholder")
```

```
ChartBox$highlightElement()
```

```
# Extract block characteristics
```

```
BoxInfo <- ChartBox$getElementLocation()
```

```
#Determining the step between candles ----
```

```
v <- c()
```

```
n <- 0
```

```

repeat{
  remDr$mouseMoveToLocation(-BoxInfo$width/2 + n, 1, webElement = ChartBox)

  cross_x <- remDr$findElements(using = "xpath", "//div[@class = 'stx_crosshair
stx_crosshair_x crossX']")
  left <- cross_x %>% sapply(., function(x) x$getElementValueOfCssProperty("left"))

  v <- append(v, left[[1]])

  if(n == 40){break}
  n = n + 1
}

freq <- table(table(v), dnn = NULL) #define frequency
step <- as.numeric(names(freq)[which.max(freq)])

#Data collection ----
n=0
i=1

ListContainer<-list()

while(n < BoxInfo$width){

  remDr$mouseMoveToLocation(-BoxInfo$width/2 + 1 + n, 1, webElement = ChartBox)

  ValueBox <- remDr$findElements(using = "xpath", "//div[@class = 'stx-tooltip']")

  values <- ValueBox %>% sapply(., function(x) x$getElementText())

```

```
TimeFrame <- remDr$findElement(using = "xpath", "//div[@class = 'stx-float-date floatDate']")
```

```
time <- TimeFrame$getElementText()
```

```
t<- strsplit(values[[1]], "\n")
```

```
t[[1]][c(2:7)]<-parse_number(t[[1]][c(2:7)])
```

```
t[[1]][1] <- time[[1]]
```

```
ListContainer[i]=t
```

```
n = n + step
```

```
i=i+1
```

```
}
```

```
#Dataframe creation ----
```

```
DataContainer<-data.frame(Reduce(rbind, ListContainer),row.names=NULL, stringsAsFactors = FALSE)
```

```
colnames(DataContainer) <- c("Time", "Open", "High", "Low", "Close", "Volume", "Change")
```

Код інтерактивної панелі управління:

title: "Untitled"

runtime: shiny

output:

flexdashboard::flex_dashboard:

orientation: columns

vertical_layout: scroll

source_code: embed

```{r setup, include=FALSE}

library(flexdashboard)

library(shiny)

library(shinyjs)

library(shinyalert)

library(lobstr)

library(htmltools)

library(readr)

library(magrittr)

library(DT)

library(ggplot2)

library(plotly)

library(quantmod)

library(forecast)

library(tseries)

library(fGarch)

library(rugarch)

```

```

```
PAGE1 {data-icon="fa-table"}
```

```
Column_1 {.sidebar}
```

```
```{r}
```

```
useShinyjs(rmd = TRUE)
```

```
fileInput(inputId="Upload_1", label="Choose the file", multiple = FALSE, accept =
".csv")
```

```
textInput(inputId="Upload_2", label="Symbol", value="")
```

```
dateRangeInput("Dates_set", label = h3("Date range"), start = Sys.Date()-800, end =
Sys.Date())
```

```
actionButton("reset", label = "Reset", width="80px")
```

```
actionButton("submit", label = "Submit", class="btn btn-primary")
```

```
---
```

```
```{r}
```

```
observeEvent(input$submit11,{
```

```
 shinyjs::reset("Dates_plot")
```

```
 value$Date_min<-NULL
```

```
 value$Date_max<-NULL
```

```
})
```

```

```

```
```{r}
```

#evaluation featres of isTruthy() - <https://cran.r-project.org/web/packages/shiny/shiny.pdf>

```
observeEvent(input$submit,{

  if (fileCheck() && input$Upload_2=="") {

    Import1<-read_csv(input$Upload_1$datapath)
    if(ncol(Import1)==1){
      Import1<-read_csv2(input$Upload_1$datapath)
    }
    value$data<-as.data.frame(Import1)
  }
  else if (isFALSE(fileCheck()) && isTruthy(input$Upload_2)==TRUE){

    tryCatch({
      if(is.na(input$Dates_set[1]) || is.na(input$Dates_set[2])){
        Import2<-getSymbols(input$Upload_2, src="yahoo", auto.assign = FALSE)
      }
      else{
        Import2<-getSymbols(input$Upload_2, from=input$Dates_set[1],
to=input$Dates_set[2]+1, src="yahoo", auto.assign = FALSE)
      }

      value$data<-data.frame(Date=index(Import2), coredata(Import2)) },

    error = function(c) {runjs('alert("Error: symbol does not exist.");'),
    finally={gc()}}
  )
}
```

```

else {
  erase()
}
})
```

```{r}
observeEvent(input$reset,{erase()})
```

```{r}
erase<-function() {
  reset("Upload_1")
  reset("Upload_2")
  value$addr_old <- value$addr_new
  runjs('alert("Controls reset!");')
}
```

```{r}
#Data storage
value <- reactiveValues(
  data = NULL,
  addr_old=NULL,
  addr_new=NULL,
  DateColInd=NULL,
  Date_min=NULL,
  Date_max=NULL,
  chose=NULL,

```

```

    series=NULL
  )
  ...

  ``{r}
  #block of code, that activates once. Needed for fileInput control reset
  observeEvent(handlerExpr=NULL,{
    value$addr_old<-obj_addr(input$Upload_1)}, once = TRUE)
  ...

  ``{r}
  #Here we check, whether fileInput has been updated.
  fileCheck<-eventReactive(c(input$Upload_1, value$addr_old),{

    value$addr_new <- obj_addr(input$Upload_1)

    if(value$addr_new == value$addr_old) {FALSE}
    else {
      value$addr_old == value$addr_new
      TRUE
    }
  })
  ...

  ``{r}
  #Initialization block
  observeEvent(value$data,{
    req(is.data.frame(value$data)==TRUE)

```



```
value$data<-na.omit(value$data) #modify a bit
```

```
is.date <- function(x) inherits(x, 'Date')
```

```
index <- sapply(value$data, is.date) %>% which
```

```
if(length(index) == 1) {
```

```
  value$Date_min <- value$data[1,index]
```

```
  value$Date_max <- value$data[nrow(value$data),index]
```

```
  value$DateColInd<-index
```

```
}
```

```
else {
```

```
  is.convertible.to.date <- function(x) !anyNA(as.Date(as.character(x), tz = 'UTC', format
= '%Y-%m-%d'))
```

```
  index <- sapply(value$data, is.convertible.to.date) %>% which
```

```
  if(length(index) != 1) {
```

```
    ### make an alert
```

```
  }
```

```
  else {
```

```
    value$Date_min <- value$data[1,index]
```

```
    value$Date_max <- value$data[nrow(value$data),index]
```

```
    value$DateColInd<-index
```

```
  }
```

```
}
```

```
updateDateRangeInput(session, "Dates_plot",
  start= value$Date_min,
  end= value$Date_max
)
```

```
value$choise <- as.list(colnames(value$data)[unlist(lapply(value$data, is.numeric),
use.names = FALSE)])
```

```
updateSelectizeInput(session, "select", choices = value$choise, selected = NULL, options
= list(), server = FALSE)
```

```
updateSelectizeInput(session, "finalSelect", choices = value$choise, selected = NULL,
options = list(), server = FALSE)
```

```
})
```

```
```
```

```
```{r}
```

```
observe({
```

```
  req(is.null(value$data)==FALSE) #TryCatch - не работает, необходимо дописать!
```

```
  if (input$Finaldates[1]<input$Finaldates[2]){
```

```
    colIndex <- colnames(value$data) %>% grepl(input$finalSelect,.) %>% which
```

```
    value$series <- value$data[(value$data[,value$DateColInd]>=input$Finaldates[1] &
value$data[,value$DateColInd]<=input$Finaldates[2]), colIndex]
```

```
  } else {
```

```
    colIndex <- colnames(value$data) %>% grepl(input$finalSelect,.) %>% which
```

```

    value$series <- value$data[, colIndex]
  }
})
...

## Column_2

### Time series plot
```{r}
fillCol(flex=c(8.5,1.5),
 plotlyOutput("MainPlot", height="100%"),

 fillRow(
 dateRangeInput("Dates_plot", label = "Date range"),
 selectizeInput("select", label = "Select box",
 choices = NULL,
 selected = 1, options=list(dropdownParent="body"))
)
)
...

```{r}
output$MainPlot <-renderPlotly({

  req(is.data.frame(value$data), isTruthy(value$choise))

  p<-ggplot(value$data)+aes_string(x=value$data[,value$DateColInd], y= input$select) +
  coord_cartesian(xlim = c(input$Dates_plot[1],input$Dates_plot[2])) + geom_line()

```

```

plotly::ggplotly(p)

})
```

Data transformation panel

```{r}
fillRow(
  selectizeInput("finalSelect", label = h3("Series to analyze"),
    choices = NULL,
    selected = 1, options=list(dropdownParent="body")),

  dateRangeInput("Finaldates", label = h3("Range")), height = "100px"
)

checkboxGroupInput("checkGroup", label = h3("Checkbox group"),
  choices = list("Outlier" = 1, "Round" = 2, "Log" = 3),
  selected = 1)

```

```{r}
observe({
  updateDateRangeInput(session, "Finaldates",
    start= value$Date_min,
    end= value$Date_max
  )
})

```

```
```
```

```
Datatable
```

```
```{r}
```

```
output$DataTable<- renderDataTable(
```

```
  datatable(value$data, filter="top") #%>%
```

```
  #formatRound(colnames(value$data)[unlist(lapply(value$data, is.numeric), use.names
```

```
= FALSE)], 2)
```

```
)
```

```
```{r}
```

```
dataTableOutput('DataTable')
```

```
```
```

```
```{r}
```

```
output$a111<- renderPrint({ mData$Tseries })
```

```
verbatimTextOutput("a111")
```

```
```
```

```
# ARIMA/ARCH/GARCH
```

```
```{r}
```

```
mData <- reactiveValues(
```

```
 Tseries=NULL,
```

```
 arima=NULL,
```

```

p=NULL,
q=NULL,
d=NULL,
adf=NULL,
diffSeries=NULL,
LMtestTable=NULL,
ArchModels=NULL,
crARCH=NULL,
GarchModels=NULL,
crGARCH=NULL,
TGarchModels=NULL,
crTGARCH=NULL,
MGarchModels=NULL,
crMGARCH=NULL
)
...

```{r}
actionButton("button1", "Analyze")
...

```

Foreword

In financial time series, it is often that the series is transformed by logging and then the differencing is performed. People often look at the returns of the stock rather than the its prices. Differences of log prices represent the returns and are similar to percentage changes of stock prices.

Estimated Arima model

```

```{r}
observeEvent(input$button1,{
 req(is.null(value$series)==FALSE)
 mData$Tseries <- ts(value$series)

 if(any(input$checkGroup %>% grepl('^1$',..))) mData$Tseries <- tsclean(mData$Tseries)
 if(any(input$checkGroup %>% grepl('^2$',..))) mData$Tseries <-
round(mData$Tseries,2)
 if(any(input$checkGroup %>% grepl('^3$',..))) mData$Tseries <- log(mData$Tseries)

 mData$arima<-auto.arima(mData$Tseries, ic="aic", trace=TRUE)
})
```

```

```

```{r}
output$EstimatedArima<- renderPrint({ mData$arima })
verbatimTextOutput("EstimatedArima")
```

```

Stationarity and differencing of time series data set:

Check of stationarity:

Ho: series is non-stationary (p-value > 0.05)

H1: series is stationary (p-value < 0.05)


```

```{r}

```

#calculations block !!!!!

```

observeEvent(mData$arima,{

#Getting the orders of arima model
mData$p <- mData$arima$arima[1]
mData$q <- mData$arima$arima[2]
mData$d <- mData$arima$arima[length(mData$arima$arima)-1]

if(mData$d ==0){
 mData$adf<-adf.test(mData$Tseries, alternative = "stationary")
}
else {

 i=0
 for(i in mData$d-1){
 mData$Tseries<-diff(mData$Tseries, differences = 1)
 mData$adf<-adf.test(mData$Tseries, alternative = "stationary")
 }
}
mData$diffSeries<-mData$Tseries
})
...

```{r}
output$AdfTest<- renderPrint({ mData$adf })
#output$Tseries<- renderPrint({ mData$Tseries }) #nigde net
...

```{r}

```



```
verbatimTextOutput("AdfTest")
```

```
```
```

```
### Series after differencing
```

```
```{r}
```

```
output$diffplot <-renderPlotly({
```

```
 req(is.null(mData$diffSeries)==FALSE)
```

```
 diff<-ggplot()+aes_string(x=1:length(mData$diffSeries), y= mData$diffSeries) +
 xlab("diff series") + geom_line()
```

```
 plotly::ggplotly(diff)
```

```
})
```

```
```
```

```
```{r}
```

```
plotlyOutput("diffplot", height="100%")
```

```
```
```

```
### Residuals plot: ACF & PACF
```

Observing residual plot and it's ACF & PACF diagram

If ACF & PACF of the model residuals show no significant lags, the selected model is appropriate.

```
### ACF plot
```

```
```{r}
```

```
plotOutput(outputId = "ACF")
```

```
output$ACF<- renderPlot({
```

```
 req(is.null(mData$diffSeries)==FALSE)
```

```
 acf(mData$diffSeries)
```

```
})
```

```
...
```

```
PACF plot
```

```
``{r}
```

```
plotOutput(outputId = "PACF")
```

```
output$PACF<- renderPlot({
 req(is.null(mData$diffSeries)==FALSE)
 pacf(mData$diffSeries)
})
```

```
...
```

```
Ljung-Box test
```

Ljung-Box is a test of autocorrelation in which it verifies whether the autocorrelations of a time series are different from 0.<br>

More formally, the Ljung-Box test can be defined as follows:<br>

H0: The data are random. - independent and uncorrelated;<br>

H1: The data are not random. - remains serial correlation;<br>

Note: If the p-value is greater than 0.05 then the residuals are independent which we want for the model to be correct.<br>

```
``{r}
```

```
output$LBtest <- renderPrint({
 req(is.null(mData$arima)==FALSE)
```

```

Box.test(residuals(mData$arima),lag=12, type="Ljung-Box")
})
verbatimTextOutput("LBtest")
```

```

Diagnostic checking

1.1) Observing residual plot and it's ACF & PACF diagram

If ACF & PACF of the model residuals show no significant lags, the selected model is appropriate.


```

```{r}
plotOutput(outputId = "ModelResiduals")

output$ModelResiduals<- renderPlot({
 req(is.null(mData$arima)==FALSE)
 tsdisplay(residuals(mData$arima), lag.max=15, main='Model Residuals')
})
```

```

ARIMA forecast

```

```{r}
observeEvent(mData$arima,{

 ArimaPred<-as.data.frame(forecast(mData$arima, h=4))

 if (any(input$checkGroup %>% grepl('^3$',.))){
 mData$ArimaPred<-exp(ArimaPred)
 } else {

```

```

mData$ArimaPred<-ArimaPred
}
}))

...

```{r}
output$ArimaPrediction <- renderDataTable(

  datatable(mData$ArimaPred, filter="top")
)
...

```{r}
dataTableOutput('ArimaPrediction')
...

```

### ### Remark

Although ACF and PACF of residuals have no significant lags, the time series plot of residuals shows some cluster of volatility.<br>

It is important to note that ARIMA is a method to @linear model the data and the forecast width remains constant because the model does not reflect recent changes or incorporate new information.<br>

In other words, it provides best linear forecasts for the series, and thus plays little role in forecasting model nonlinearly.<br>

In order to model volatility, ARCH/GARCH method is used.<br>

### ### Testing for ARCH effect

- 1) Firstly, check if residual plot displays any cluster of volatility. <br>
- 2) Observe the squared residual plot; Are there any volatility clusters? <br>
- 3) Observe ACF & PACF of squared residuals. <br>

```
```{r}
```

```
plotOutput(outputId = "ArchEffect")
```

```
output$ArchEffect<- renderPlot({
  req(is.null(mData$arima)==FALSE)
  tsdisplay(residuals(mData$arima)^2, lag.max=15, main='Model Residuals')
})
```
```

### ### Remark

A strict white noise cannot be predicted either linearly or nonlinearly while general white noise might not be predicted linearly yet done so nonlinearly. If the residuals are strict white noise, they are independent with zero mean, normally distributed, and ACF & PACF of squared residuals displays no significant lags.

### ### LM test

As an alternative to Engle's ARCH test, you can check for serial dependence (ARCH effects) in a residual series by conducting a Ljung-Box Q-test on the first  $m$  lags of the squared residual series, where  $m = P + Q$ , ARCH( $P*Q$ ), GARCH( $P,Q$ ).

```
```{r}
```

```
observe({
  req(is.null(mData$arima)==FALSE)
```

```
LMtestTable <- data.frame(matrix(ncol = 5, nrow = 1))
```

```

for(i in 1:5){
  u <-Box.test(residuals(mData$arima)^2,lag=i, type="Ljung-Box")
  LMtestTable[1,i]<-u$p.value
}

colnames(LMtestTable) <- c("lag 1","lag 2","lag 3","lag 4", "lag 5")

mData$LMtestTable <- LMtestTable
})
```



```

```{r}
output$TestResultTable <- renderDataTable(
 datatable(mData$LMtestTable)
)
```

```{r}
dataTableOutput('TestResultTable')
```

```


```

### ### ARCH notes

The general rules for ARCH variance coefficients are following:

$$ht = a_0 + a_1 e^2 + a_2 e^2 \dots, a_0 > 0, 0 < (\text{sum of } a_i) \leq 1$$

### ### ARCH model selection

```

```{r}

```

```

observe({
  req(is.null(mData$LMtestTable)==FALSE)

  list1<-list()
  list2<-list()
  for(i in 1:4){
    t<-garchFit(substitute(~arma(p,q)+garch(i,0),list(p=mData$p, q=mData$q, i=i)),
data=mData$Tseries)
    t1<-attributes(t)$fit$ics
    list1[[i]]<-t
    list2[[i]]<-t1
  }

  mData$ArchModels<-list1
  #mData$scrARCH<-list2
  ctr<-as.data.frame(list2)
  ctr<-as.data.frame(t(ctr))
  rownames(ctr)<-c("ARIMA-Arch(1)", "ARIMA-Arch(2)", "ARIMA-Arch(3)",
"ARIMA-Arch(4)")
  mData$scrARCH<-ctr
})
```

```{r}
output$ArchModelslist<- renderPrint({ mData$ArchModels
})
verbatimTextOutput("ArchModelslist")
```

```

```
ARCH models estimated criteria
```

```
`r`
```

```
output$ARCH_CR <- renderDataTable(
 datatable(mData$crARCH)
)
`r`
```

```
`r`
```

```
dataTableOutput('ARCH_CR')
`r`
```

```
GARCH model selection
```

```
`r`
```

```
observe({
 req(is.null(mData$LMtestTable)==FALSE)
```

```
list3<-list()
```

```
list4<-list()
```

```
z=1
```

```
for(i in 1:2){
```

```
 for(j in 1:2){
```

```
 garchSpec <- ugarchspec(
```

```
 variance.model=list(model="sGARCH",
```

```
 garchOrder=c(i,j)),
```

```
 mean.model=list(armaOrder=c(mData$p,mData$q)),
```

```
 distribution.model="std")
```



```
garchFit <- ugarchfit(spec=garchSpec, data=mData$Tseries)
```

```
t3<-garchFit
```

```
t4<-infocriteria(garchFit)
```

```
list3[[z]]<-t3
```

```
list4[[z]]<-t4
```

```
z=z+1
```

```
}
```

```
}
```

```
mData$GarchModels<-list3
```

```
ctrr<-as.data.frame(list4)
```

```
colnames(ctrr)<-c("ARIMA-Garch(1,1)", "ARIMA-Garch(1,2)", "ARIMA-Garch(2,1)",
"ARIMA-Garch(2,2)")
```

```
mData$crGARCH<-ctrr
```

```
})
```

```
```
```

```
```{r}
```

```
output$GarchModelslist<- renderPrint({ mData$GarchModels
```

```
})
```

```
verbatimTextOutput("GarchModelslist")
```

```
```
```

```
### GARCH models estimated criteria
```

```
``{r}
output$GARCH_CR <- renderDataTable(
  datatable(mData$crGARCH)
)
``
```

```
``{r}
dataTableOutput('GARCH_CR')
``
```

```
### TGARCH model selection
```

```
``{r}
observe({
  req(is.null(mData$LMtestTable)==FALSE)
```

```
list3<-list()
```

```
list4<-list()
```

```
z=1
```

```
for(i in 1:2){
  for(j in 1:2){
    garchSpec <- ugarchspec(
      variance.model=list(model="fGARCH",
        garchOrder=c(i,j),
        submodel="TGARCH"),
```

```

        mean.model=list(armaOrder=c(mData$p,mData$q)),
        distribution.model="std")
garchFit <- ugarchfit(spec=garchSpec, data=mData$Tseries)

t3<-garchFit
t4<-infocriteria(garchFit)

list3[[z]]<-t3
list4[[z]]<-t4

z=z+1
}
}

mData$TGarchModels<-list3

ctrr<-as.data.frame(list4)
colnames(ctrr)<-c("ARIMA-TGarch(1,1)", "ARIMA-TGarch(1,2)", "ARIMA-
TGarch(2,1)", "ARIMA-TGarch(2,2)")
mData$scrTGARCH<-ctrr

})
```

```{r}
output$TGarchModelslist<- renderPrint({ mData$TGarchModels
})
verbatimTextOutput("TGarchModelslist")

```

```
```
```

```
TGARCH models estimated criteria
```

```
```{r}
```

```
output$TGARCH_CR <- renderDataTable(
  datatable(mData$crTGARCH)
)
```

```
```
```

```
```{r}
```

```
dataTableOutput('TGARCH_CR')
```

```
```
```

```
GARCH-in-Mean model selection
```

```
```{r}
```

```
observe({
  req(is.null(mData$LMtestTable)==FALSE)
```

```
list3<-list()
```

```
list4<-list()
```

```
z=1
```

```
for(i in 1:2){
```

```
  for(j in 1:2){
```

```
    garchSpec <- ugarchspec(
```

```
      variance.model=list(model="fGARCH",
```

```

        garchOrder=c(i,j),
        submodel="APARCH"),
    mean.model=list(armaOrder=c(mData$p,mData$q),
        include.mean=TRUE,
        archm=TRUE,
        archpow=2),
    distribution.model="std")
garchFit <- ugarchfit(spec=garchSpec, data=mData$Tseries)

t3<-garchFit
t4<-infocriteria(garchFit)

list3[[z]]<-t3
list4[[z]]<-t4

z=z+1
}
}

mData$MGarchModels<-list3

ctrr<-as.data.frame(list4)
colnames(ctrr)<-c("ARIMA-MGarch(1,1)", "ARIMA-MGarch(1,2)", "ARIMA-
MGarch(2,1)", "ARIMA-MGarch(2,2)")
mData$crMGARCH<-ctrr

})
...

```

```

```{r}
output$MGarchModelslist<- renderPrint({ mData$MGarchModels
})
verbatimTextOutput("MGarchModelslist")
```

```

GARCH-In-Mean models estimated criteria

```

```{r}
output$MGARCH_CR <- renderDataTable(
 datatable(mData$crMGARCH)
)
```

```

```

```{r}
dataTableOutput('MGARCH_CR')
```

```

ARNN

```

```{r}
netData <- reactiveValues(
 Seriess=NULL,
 modelfit=NULL,
 NNF=NULL,
 fcast=NULL
)
```

```

```

```{r}
actionButton("button2", "Analyze")
```

```{r}
observeEvent(input$button2,{
 req(is.null(value$series)==FALSE)
 netData$Seriesess <- ts(value$series)

 if(any(input$checkGroup %>% grepl('^1$',..))) netData$Seriesess <-
tsclean(netData$Seriesess)

 if(any(input$checkGroup %>% grepl('^2$',..))) netData$Seriesess <-
round(netData$Seriesess,2)

 if(any(input$checkGroup %>% grepl('^3$',..))) netData$Seriesess <- log(netData$Seriesess)

 netData$modelfit <- nnetar(netData$Seriesess, repeats = 20, lambda=0.5)
})
```

```

Foreword

Feed-forward neural networks with a single hidden layer and lagged inputs for forecasting univariate time series. The `nnetar` function in the `forecast` package for R fits a neural network model to a time series with lagged values of the time series as inputs (and possibly some other exogenous inputs). So it is a nonlinear autoregressive model, and it is not possible to analytically derive prediction intervals. Therefore we use simulation. The neural networks is fit by the function:


```
nnetar(y, p, P = 1, size, repeats = 20, xreg = NULL, lambda = NULL, model = NULL,
subset = NULL, scale.inputs = TRUE, x = y, ...)
```

```

```{r}
output$ResultNet<- renderPrint({ netData$modelfit
})
verbatimTextOutput("ResultNet")
```

### ARNN Forecast
```{r}
observeEvent(netData$modelfit,{
 req((is.null(netData$modelfit)==FALSE))

 if (any(input$checkGroup %>% grepl('^3$',.))){
 netData$NNF <- forecast(netData$modelfit, PI=TRUE, h=4)
 netData$fcast <- exp(as.data.frame(netData$NNF))
 } else {
 netData$NNF <- forecast(netData$modelfit, PI=TRUE, h=4)
 netData$fcast <- as.data.frame(netData$NNF)
 }

})

```

```{r}
output$ARNNForecast <- renderDataTable(
 datatable(netData$fcast)
)
```

```



```

```{r}
dataTableOutput('ARNNForecast')
```

```

Forecast plot

```

```{r}
plotOutput(outputId = "netPlot")
```

```

```

```{r}
output$netPlot <- renderPlot({
 req(is.null(netData$NNF)==FALSE)
 forecast::autoplot(netData$NNF)
})
```

```

Київський національний університет імені Тараса Шевченка
Економічний факультет
Кафедра економічної кібернетики

ЗАВДАННЯ

на кваліфікаційну бакалаврську роботу

студента 4 курсу спеціальності 6.030502 „Економічна кібернетика”

Атояна Андрія Гарійовича

- 1. Тема роботи:** «Розробка програмного комплексу для моделювання та прогнозування цін акцій на фондовій біржі»
- 2. Термін завершення роботи:** 22.05.2020
- 3. Попередній захист роботи:** 25.05.2020
- 4. Об'єкт дослідження:** біржові індекси акцій компаній.
- 5. Предмет дослідження:** є автоматизація процесу аналізу та прогнозування акцій з використанням статистичних методів, та методів машинного навчання.
- 6. Мета дослідження:** розробка програмного комплексу для аналізу та прогнозування цін акцій фондових ринків.
- 7. Завдання дослідження:**
 - 7.1.** дослідження наукових робіт стосовно методів та підходів аналізу інструментів фінансового ринку.
 - 7.2.** вибір ефективної методології прогнозування часових рядів.
 - 7.3.** розробка алгоритму збору даних, та його автоматизація.
 - 7.4.** створення інтерактивної панелі управління для попередньої обробки даних та їх аналізу.
 - 7.5.** оцінка результатів роботи програмного комплексу, порівняння моделей та виборі найкращої з них.
 - 7.6.** розробка прогнозу часового ряду на наступні n періодів.

Науковий керівник: д.е.н., доцент Ставицький Андрій Володимирович

Студент: Атоян Андрій Гарійович

Затверджено на засіданні кафедри економічної кібернетики

протокол № ... від..... 2020 р.

Календарний план виконання бакалаврської роботи

| № | Етапи бакалаврської роботи | Терміни виконання | Відмітка керівника про виконання |
|----|---|---------------------------|----------------------------------|
| 1 | Вибір теми бакалаврської роботи | 10.10.2019-
5.11.2019 | |
| 2 | Розробка та затвердження завдання на роботу бакалавра | 6.11.2019-
13.11.2019 | |
| 3 | Огляд літератури | 01.01.2020-
01.02.2020 | |
| 4 | Написання першого розділу | 01.02.2020-
14.03.2020 | |
| 5 | Вибір методології | 14.03.2020-
18.03.2020 | |
| 6 | Написання другого розділу | 19.03.2020-
07.04.2020 | |
| 7 | Виконання практичної частини у середовищі програмування R | 07.04.2020-
24.04.2020 | |
| 8 | Оформлення практичних результатів у третій розділ | 24.04.2020-
30.04.2020 | |
| 9 | Опрацювання правок та зауважень | 30.04.2020-
24.04.2020 | |
| 10 | Подання роботи до попереднього захисту | 25.05.2020 | |
| 11 | Остаточне коригування роботи, зважаючи на коментарі наукового керівника | 26.05.2020-
28.05.2020 | |

Науковий керівник: д.е.н., доцент Ставицький Андрій Володимирович

Студент: Атоян Андрій Гарійович