Одеський національний університет імені І. І. Мечникова
Факультет математики, фізики та інформаційних технологій
Кафедра оптимального керування та економічної кібернетики

Дипломна робота

бакалавра

на тему: **«Моделі машинного навчання для прогнозування часових рядів при обмеженні історичних даних»**

«Machine learning models for time series forecasting with limited historical data»

Виконала: студентка денної форми навчання
спеціальності 113 Прикладна математика
Міловська Карина Миколаївна

Керівник: к. т. н., доц. Мороз В. В.
Рецензент: д. ф.-м. н., доц. Кічмаренко О. Д.

Рекомендовано до захисту:
Протокол засідання кафедри
№ ___ від «_____» _____ р.
Завідувач кафедри

_____ _____

Захищено на засіданні ЕК № _____
Протокол № ____ від «____» ____ р.
Оцінка _____ / _____ / _____
Голова ЕК

_____ _____

Одеса — 2021 р.

ЗМІСТ

ВСТУП

Протягом останніх десятиліть у світі активно відбувається процес інформатизації економіки, в умовах якої гостро постає питання взаєморозрахунків між людьми шляхом впровадження інноваційної мережі платежів та нових платіжних засобів. Набуває популярності поняття віртуальної валюти — криптовалюти. Криптовалюта – різновид цифрової валюти, створення і контроль за якою базуються на криптографічних методах. Термін закріпився після статті «Cryptocurrency», яка була опублікована у 2011 році у журналі «Forbes» [8].

Криптовалюта у сучасній системі міжнародних валютно-фінансових та кредитних відносин є зручною формою електронних розрахунків і перспективною для здійснення інвестицій. На сьогоднішній день «CoinMarketCap» включає близько 2500 цифрових активів. Користуються попитом серед споживачів ті криптовалюти, які мають високу ліквідність, постійність курсових показників, перспективи подальшого розвитку і позитивну репутацію засновників.

Найбільш популярними трендами сфери цифрових грошових одиниць є:

- Біткоін (Bitcoin, BTC) — лідер серед віртуальних грошей, на основі блокчейна постійно створюються різноманітні технічні рішення, а розробники постійно працюють над їх модернізацією;
- Ефіріум (Ethereum, ETH) — стимулює появу децентралізованих стартапів на основі блокчейна, що базуються на смарт-контрактах;
- Ripple (XRP) — найбільш швидка криптовалюта;

У результаті виявлених трендів розвитку криптовалюти зростає увага до криптовалютних торгів, які являють собою постійно зростаючий і перспективний фінансовий ринок. Для того, щоб заробити на криптовалютному трейдингу необхідно вміти правильно прогнозувати майбутній рух цін, що потенційно може привести до високого прибутку, оскільки інвестор може купувати, зберігати і продавати цифрові валюти в потрібний час на основі своїх прогнозів. Таким чином, розробка точного інструменту прогнозування має важливе значення.

Актуальність дослідження: інноваційність та великі перспективи

робить криптовалюти цінним активом, який варто прогнозувати, тому питання формування та прогнозування на ринку криптовалют є надзвичайно цікавим та актуальним.

Метою дослідження є розв'язання проблеми прогнозування ціни криптовалют за допомогою методів машинного навчання, а також виявлення залежності точності прогнозу від довжини часового ряду.

Поставлена мета зумовила необхідність вирішення таких завдань:

- Аналіз і дослідження предметної області;
- Огляд і аналіз існуючих методів прогнозування курсів криптовалют;
- Визначення ключових факторів, які впливають на об'єкт дослідження;
- Визначення ефективності наявних методів прогнозування;
- Прийняття рішення про використання методу для прогнозування курсу криптовалют;
- Написання програмного коду для прогнозування цін;
- Визначення ефективності створеного рішення.

Об'єктом дослідження є проблема прогнозування часових рядів.

Предметом дослідження є методи машинного навчання для прогнозування ціни криптовалют в залежності від розміру даних і частоти дискретизації.

Дана робота пройшла апробацію на 20-й міжнародній науково-технічній конференції «Проблеми інформатики та моделювання», яка проводилася на базі національного технічного університету «Харківський політехнічний інститут» у 2020 році. Тези було опубліковано у збірнику [35]. У 2021 році робота пройшла апробацію результатів дослідження на 77-й звітній студентській науковій конференції Одеського національного університету імені І.І.Мечникова. Тези опубліковано у збірнику [25].

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Визначення та характеристики часового ряду

Означення 1.1. [7] Часовий ряд - це ряд спостережень за значеннями $y_1, y_2, \ldots, y_T$ деякого показника (ознаки), упорядкований у хронологічній послідовності, тобто у порядку зростання змінного t-часового параметра. Будемо позначати часовий ряд з T елементами Y_T.

Окремі спостереження часового ряду називають його рівнями. Якщо кожному моменту часу відповідають значення лише одного показника, тобто, якщо ряд містить ряд містить записи однієї змінної, він є одновимірним. В іншому випадку, коли розглядаються розглядати записи більш ніж однієї змінної – ряд є багатовимірним.
Часові ряди використовуються в статистиці, обробці сигналів, розпізнаванні образів, економетриці, фінансовій математиці, прогнозуванні погоди, розумному транспорті та передбаченні траєкторій, передбаченні землетрусів, електроенцефалографії, автоматичному керуванні, астрономії, технологіях зв'язку, а також значною мірою в будь-якій області прикладної науки та інженерії, яка включає часові вимірювання.

1.1.1. Компоненти часового ряду

Вважається, що на часовий ряд в цілому впливають чотири основних компоненти, які можуть бути відокремлені від спостережуваних даних. Цими компонентами є: тренд, сезонність, циклічність і випадкова компонента.

- Тренд — це основна тенденція часового ряду, систематична складова довготривалого змінювання досліджуваного показника. Трендом називають гладку зміну, яка проявляється на великому проміжку часу і є результатом дії довготривалих факторів.

- Сезонна компонента характеризує циклічні коливання досліджуваного показника навколо трендової компоненти протягом року, протягом сезону.
- Циклічна компонента описує середньострокові зміни ряду, викликані обставинами, які повторюються в циклах.
- Виключення з часового ряду тренда і періодичних складових дозволяє отримати нерегулярну компоненту — випадкову, яка може формуватися як під впливом непередбачених природних катаклізмів (землетруси, пожежі, епідемії), так і поточних чинників — помилки вимірювання.

За наявності випадкової компоненти неможливо прогнозувати значення часового ряду без помилки. Але будь-який реальний процес включає випадковий компонент.
Таким чином, модель часового ряду можна описати як

$$y_t = u_t + s_t + c_t + \epsilon_t$$

— адитивна модель чи

$$y_t = u_t \cdot s_t \cdot c_t \cdot \epsilon_t$$

— мультиплікативна модель,
де y_t — спостереження; u_t — тренд; s_t — сезонність; c_t — циклічність; ϵ_t — випадкова компонента.
Слід зауважити, що не завжди представляється можливим характеризувати кожний компонент окремо. Іноді краще робити прогноз відносно всієї моделі, ніж намагатися виділити кожний компонент окремо.

1.2. Прогнозування часових рядів

Завдяки роботам Макрідакіса стало очевидним, що прогнози, які використовують тільки оцінку, не є такими точними, як ті, які ґрунтуються на застосуванні кількісних методів оцінки [18]. В результаті появи нових технологій, зросла довіра до методів, що включають складну техніку обробки даних. Комп'ютери, в сукупності з кількісними методами розрахунків, які завдяки їм стали загальнодоступними, для сучасних організацій є вже не

просто зручним інструментом, а фактично їх невід'ємною частиною.

Метою прогнозування є аналіз того, що може відбутися при реалізації можливих прогнозів у майбутньому та до яких наслідків це призведе.
Прогнозування відіграє важливу роль у багатьох сферах, тому сучасні організації потребують короткострокових, середньострокових, довгострокових прогнозів, залежно від конкретного застосування.
Розглянемо, в яких випадках використовуються ті чи інші прогнози на ринку криптовалют:

- Короткострокові.
  Вони складаються на один-три дні і містять максимально точну інформацію про найближчих зміни вартості електронної валюти. Такі прогнози складаються для трейдерів, які щодня проводять валютні операції, з метою своєчасного повідомлення про можливі коливання курсу.
- Середньострокові.
- Довгострокові.
  Такі прогнози дозволяють дізнатися загальну динаміку зміни курсу електронної валюти на найближчий місяць. Їх головна мета - своєчасно попередити про можливу появу факторів, що ведуть до тривалого зниження вартості електронної валюти.

1.2.1. Міри точності прогнозів

Перед тим як використовувати модель для складання реальних прогнозів, важливо оцінити точність прогнозу. Точність прогнозів можна визначити лише з урахуванням того, наскільки добре працює модель за новими даними, які не використовувались при встановленні моделі (англ. model fitting). Разом з тим, вибираючи моделі, прийнято розділяти наявні дані на дві частини: навчальний набір даних (англ. training dataset) та тестовий (англ. test dataset), де навчальний набір даних використовується для оцінки будь-яких параметрів методу прогнозування, а тестовий – для оцінки його точності, тобто для для забезпечення об'єктивної оцінки кінцевої моделі, яка відповідає навчальному набору даних[12]. Оскільки тестові дані не використовуються при визначенні прогнозів, вони повинні надати достовірну

вказівку на те, наскільки вдало модель прогнозує нові дані.

Для того, щоб оцінювати точність прогнозування конкретної моделі або для оцінки і порівняння різних моделей розглядається їх відносна продуктивність в наборі тестових даних. У зв'язку з фундаментальної важливістю прогнозування часових рядів у багатьох практичних ситуаціях, при виборі конкретної моделі слід проявляти належну обережність. З цієї причини, різні показники для оцінки роботи пропонуються в літературі для оцінки прогнозу і порівняти різні моделі. Вони також відомі як показники продуктивності. Кожна з цих перевірок є функцією фактичних і прогнозованих значень часового ряду.

Основним вимірником міри точності прогнозу є його помилка — різниця між прогнозним і фактичним значенням досліджуваної змінної.
Нехай $\widehat{y_t}$ – прогноз значення часового ряду у t-тому періоді, тоді помилка прогнозу:

$$e_t = y_t - \widehat{y_t},$$

де y_t — фактичне значення змінної для t-го виміру; $\widehat{y_t}$ — прогнозне значення досліджуваної змінної.
Для міри точності прогнозів частіше використовуються такі характеристики: MSE, RMSE, MAPE, MAE.
Використання параметричних методів аналізу точності прогнозів передбачає розрахунок вище вказаних показників точності прогнозів за n кроків:

- середня квадратична похибка:

$$MSE = \frac{1}{n}\sum_t (y_t - \widehat{y_t})^2$$

Властивості MSE такі:
– Це міра середнього квадратного відхилення прогнозованих значень.
– MSE чутливо реагує на зміну масштабу і трансформацію даних.

- корінь із середньоквадратичної похибки:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_t (y_t - \widehat{y_t})^2}$$

Оскільки RMSE – це квадратний корінь обчисленного MSE, то всі властивості MSE зберігаються і для RMSE.

- середня абсолютна похибка

$$MAE = \frac{1}{n}\sum_{t}|y_t - \widehat{y}_t|$$

Властивості MAE:
– Вимірює середнє абсолютне відхилення прогнозованих значень від вихідних.
– Для хорошого прогнозу отримане значення MAE має бути якомога менше.
– MAE залежить від масштабу вимірювань і перетворення даних.

- середньоквадратичне значення відносних до фактичних значень похибок за n кроків (у відсотках):

$$MAPE = \frac{100}{n}\sum_{t}\left(\frac{y_t - \widehat{y}_t}{y_t}\right)^2$$

Властивості MAPE:
– Цей показник являє собою відсоток від середньої абсолютної похибки.
– Він не залежить від масштабу виміру, але схильний до впливу трансформації даних.
Оскільки $MAPE$ вимірюється у відносних одиницях, тому можна говорити про деякий загальний рівень адекватності моделі на основі порівняння: Представлено важливі показники для оцінки точності моделі. Кожен з цих

MAPE	Точність прогнозу
менше 10%	Висока
10% - 20%	Добра
20% - 40%	Задовільна
40% - 50%	Погана
більше 50%	Незадовільна

методів має унікальні властивості, відмінні від інших. В експериментах краще розглядати більш ніж один критерій ефективності для винесення суджень.

1.3. Модель інтегрованого ковзного середнього (ARIMA)

Моделі інтегрованого ковзного середнього (ARIMA — Autoregressive integrated moving average) складається з трьох компонентів:

1) авторегресії (AR), що використовує залежний зв'язок між спостереженням та іншими спостереженнями із затримкою в часі.
2) інтегрована компонента (I), що диференціює вхідні спостереження з метою зробити часовий ряд стаціонарним.
3) ковзного середнього (MA), що використовує залежність між спостереженням і відхиленням від ковзного середнього попередніх спостережень.

Використовується стандартне позначення $ARIMA(p, d, q)$, де параметри позначають:

- p — кількість лагових спостережень, включених в модель;
- d — кількість разів, коли до вхідних спостережень було застосовано диференціювання. Параметр також називається ступенем диференціювання;
- q — порядок ковзного середнього.

Особливість цієї моделі в тому, що в ній залежна змінна була стаціонарна, а незалежні змінні - це всі відставання залежної змінної та/або відставання помилок. Тому дану модель дуже легко модифікувати та розширити, при находженні нових факторів впливу–просто необхідно додати один або декілька регресорів до рівняння прогнозування.

Модель $ARIMA(p,d,q)$ для нестаціонарного часового ряду X_t має вигляд:

$$\triangle^d X_t = c + \sum_{i=1}^{p} a_i \triangle^d X_{t-i} + \sum_{j=1}^{q} b_j \varepsilon_{t-j} + \varepsilon_t,$$

де ε_t — стаціонарний часовий ряд; c,a_i,b_j — параметри моделі. $\triangle^d$ — оператор різниці часового ряду порядку d (послідовне взяття d раз різниць першого порядку - спочатку від часового ряду, потім від отриманих різниць першого порядку, потім від другого порядку і т.д.).

Побудову ARIMA-моделей часового ряду можна розділити на етапи:

1) Ряд даних.
2) Визначення загального класу моделей.
3) Вибір значень p,d,q для експериментальної перевірки.
4) Оцінка параметрів моделі.
5) Діагностична перевірка на адекватність і вибір моделі.
6) прогнозування на основі даної моделі, якщо вона виявилась адекватною в кроці 4.

Перевагами моделей прогнозування ARIMA є простота та прозорість моделювання, однаковість аналізу і проектування та різноманітність сфер застосування. Сама побудова задовільної моделі ARIMA вимагає великих витрат ресурсів і часу, а також вимагає великого досвіду з боку прогнозиста. До недоліків можна віднести неможливість моделювання нелінійних залежностей.

1.4. Штучна нейронна мережа

Глибинне навчання (DL) відноситься до потужних алгоритмів машинного навчання, які спеціалізуються на вирішенні нелінійних та складних проблем, що використовують, найчастіше, великі обсяги даних, щоб стати ефективними моделями прогнозування. Точне прогнозування ціни на криптовалюту за своєю природою є складною проблемою, оскільки її значення з часом мають дуже великі коливання внаслідок майже хаотичної та непередбачуваної поведінки. Тому методи глибинного навчання можуть становити належну методологію вирішення цієї проблеми. Розглянемо деякі методи:

Штучні нейрони – це спрощені моделі біологічних нейронів, їх прототипи. Під ними мають на увазі вузли штучної нейронної мережі (ШНМ). Нейрон приймає певну множину сигналів і генерує з них вихідний сигнал, який потім подає або на вихід, або на вхід наступних нейронів. У другому варіанті коли нейрони між собою з'єднані, вони утворюють ШНМ. Першою спробою змоделювати ці процеси були нейронні мережі У.Маккалок і У.Піттса [20].

ШНМ являють собою систему з'єднаних і взаємодіючих між собою простих процесорів (штучних нейронів). Кожний процесор подібної мережі має справу тільки із сигналами, які він періодично одержує, і сигналами, які він періодично посилає іншим процесорам. На рисунку 1.1 представлено структуру штучного нейрона. Модель роботи нейрона має такий алгоритм. Сигнали x_i, надходять на вхід, де реалізується функція синапсів, де кожен має свій вагових коефіцієнт w_i. Після цього сигнали стають зваженими і надходять до лінійного суматора. Після додавання сигналів переходить до блоку активаційної функції f. Потім відбувається відповідна обробка і тоді параметр подається на вихід у вигляді сигналу y. Зазвичай функція активації обмежує сигнал нейрона у певному діапазоні, [0; 1] або [-1; 1]. Також сигнал нейрону змінюється від початкового зрушення b, який подається на вхідний сигнал блоку функції активації.

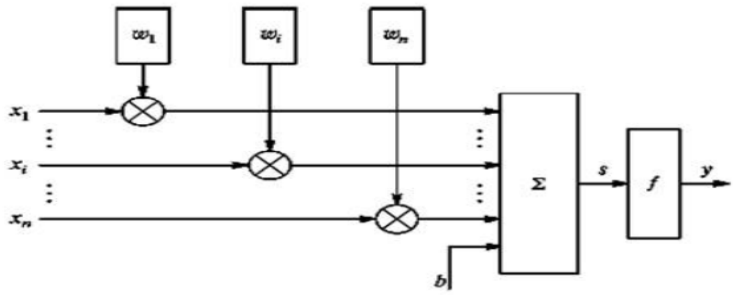

Рис. 1.1. Структура штучної нейронної мережі

Математично модель нейрона можна описати так:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right),$$

де y — вихідний сигнал нейрона; f — активаційна функція нейрона; x_i — вхідні сигнали нейрона; w_i — синаптична вага нейрона; b — зміщення нейрона. Функція активації відображає залежність вихідного сигналу нейрона від суми зважених вхідних сигналів. В нейроні функція активації f виконує нелінійне перетворення, яке здійснює нейрон. Є багато видів функцій активації: порогова бінарна функція, лінійна обмежена функція, гіперболічний тангенс, сигмоїдальна, випрямлена лінійна (Rectified linear unit, ReLU).

Штучні нейронні мережі мають перевагу в прогнозуванні часових

рядів, оскільки мають потенціал для вирішення складних проблем прогнозування. Важлива особливість ANN стосовно застосування до проблем прогнозування часових рядів полягає в здатності нейронних мереж до нелінійного моделювання, без будь-якого припущення про статистичний розподіл часового ряду. Кожна модель адаптивно формується на основі даних. З цієї причини штучні нейронні мережі керуються даними та є самоадаптивними за своєю природою.

До основних переваг нейромереж можна віднести ефективність прогнозів, можливість відтворювання складних нелінійних залежностей, адаптивність до надходження нових даних, здатність моделей до навчання. Недоліками ж є складність вибору архітектури і алгоритму навчання, складність пошуку проблеми при помилках прогнозування, ресурсомісткість процесу навчання.

1.5. Рекурентна нейронна мережа

Рекурентна нейронна мережа (Recurrent Neural Network, RNN) — являє собою мережу із циклів, які можуть зберігати інформацію. У рекурентної нейронної мережі є деякі входи, декілька прихованих шарів і декілька виходів. Це зображено на малюнку нижче:

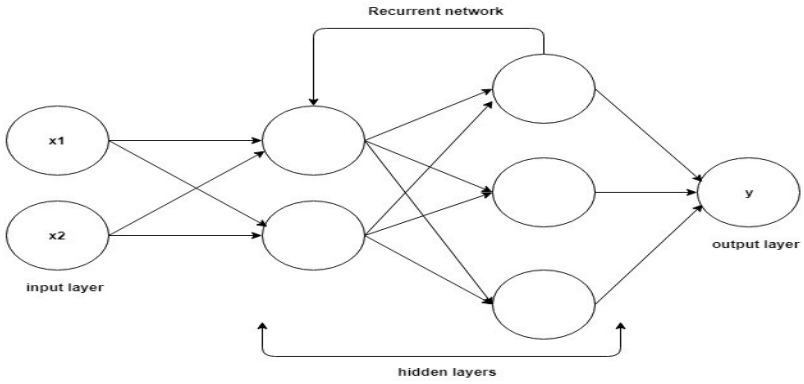

Рис. 1.2. Базова схема рекурентної нейронної мережі

Сигнал з вихідних нейронів або нейронів схованого шару частково передається назад на входи нейронів вхідного шару. Як будь-яка система, що має зворотний зв'язок, рекурентна мережа прагне до стійкого стану. Така нейронна мережа здатна знаходити не тільки прості взаємозв'язки, а й

взаємозв'язки між взаємозв'язками. У моделі RNN внутрішній шар служить для запам'ятовування інформації з попередніх спостережень. Головною проблемою є запам'ятовування невеликої кількості попередніх спостережень, що не підходить для довгих (фінансових, економічних) періодів.

1.5.1. Модель довгої короткострокової пам'яті

LSTM мережі (Long Short-Term Memory Units) — архітектура рекурентних нейронних мереж, запропонована 1997 року Зеппом Хохрайтером та Юргеном Шмідгубером [11].

Метою розробки модулів LSTM було формування більш стійкої пам'яті мережі, що значно би спростило фіксацію довгих контекстних залежностей. Загальним для всіх рекурентних мереж є те, що вони складені з ланцюга повторюваних блоків. Цей блок зазвичай має дуже просту структуру. Структура LSTM також нагадує ланцюжок, модулі якого містять чотири шари, і ці шари взаємодіють особливим чином. Цю комірку зображено на малюнку нижче:

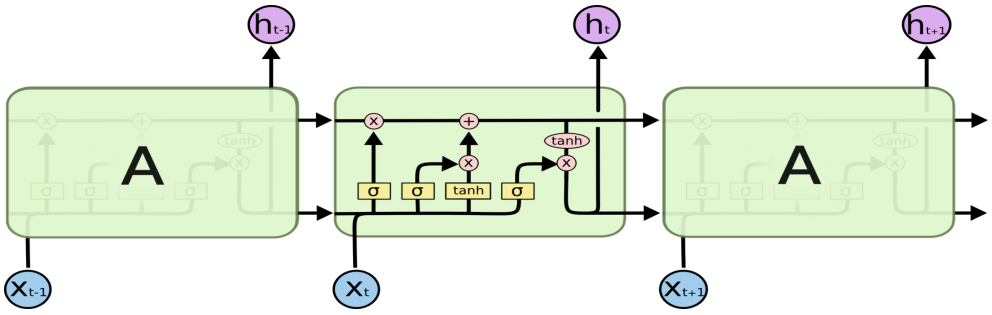

Рис. 1.3. Архітектура LSTM

Головний компонент LSTM — це стан комірки (cell state) — горизонтальна лінія, що проходить по верхніх частинах схеми. Інформація у ній може зберігатися без змін. LSTM-модуль має дуже корисну здатність до обробки послідовностей із довгостроковими залежностями: здатність додавати і видалять інформацію у стан комірки. При цьому цей процес регулюється "фільтрами" — регуляторами, які відповідають за процес забування чи збереігання інформації у комірці. Комірка пам'яті LSTM зберігає значення для довгих і коротких періодів часу. Це досягається за рахунок активаційної функції для кожної комірки пам'яті.

LSTM допомагають зберегти помилку, яку можна розповсюджувати через час і шари. Підтримуючи більш постійну помилку, вони дозволяють повторним мережам продовжувати вивчати протягом багатьох кроків часу. LSTM містять інформацію за межами нормального потоку повторюваної мережі в закритій комірці. Інформацію можна зберігати, записувати або читати з клітини, подібно до даних у пам'яті комп'ютера. Клітинка приймає рішення про те, що зберігати, і коли дозволити читання, запис і стирання, через ворота, які відкриваються і закриваються. На відміну від цифрового зберігання на комп'ютерах, ці гейти є аналоговими, реалізованими з елементарним множенням на сигмоїди, які все знаходяться в діапазоні 0-1. Ці ворота діють на сигнали, які вони отримують, і подібно до вузлів нейронної мережі, вони блокують або передають інформацію на основі своєї сили та імпорту, яку вони фільтрують своїми власними наборами ваг. Тобто, клітини вивчають, коли дозволяють вводити дані, залишати їх або видаляти через ітераційний процес внесення припущень, помилок, що поширюються назад, і коригування ваг за допомогою градієнтного спуску.

1.6. Згорткова нейронна мережа

Згорткова нейронна мережа – спеціальна архітектура нейронних мереж, від самого початку націлена на ефективне розпізнавання зображень. Ця мережа за основу схему з'єднання нейронів зорової кори тварин [19]. Згорткова нейронна мережа зазвичай являє собою чергування шарів згортки (convolution layers, C-layers), шарів субдискретізації (subsampling layers, S-layers) і за наявності повнозв'язних шарів (fully-connected layer, F-layers) на виході. Всі три види шарів можуть чергуватися в довільному порядку (рис. 1.4).

Згортка (convolution) – операція над парою матриць A і B, результатом якої є матриця $C = A \cdot B$.
Кожен елемент обчислюється як скалярний добуток матриці B і деякої підматриці A такого ж розміру. Операція згортки (рис. 1.5) визначається

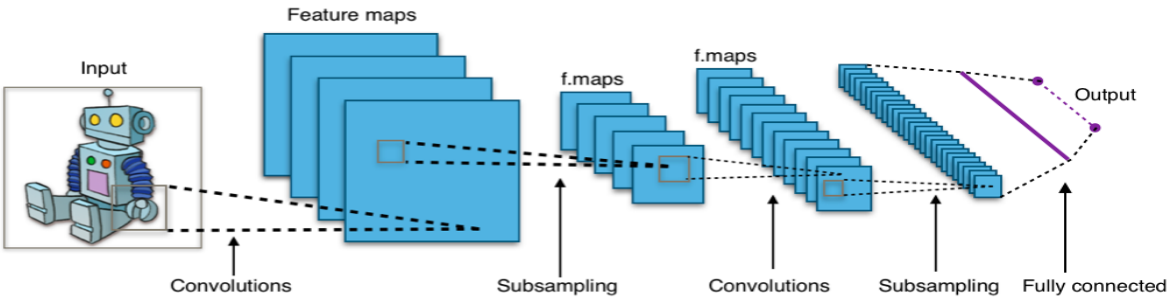

Рис. 1.4. Архітектура згорткової нейронної мережі

таким виразом:

$$C_{i,j} = \sum_{u=0}^{m_x-1} \sum_{v=0}^{m_y-1} A_{i+u,j+v} B_{u,v},$$

де $B_{u,v}$ — значення елемента ядра згортки (u, v), $C_{i,j}$ — значення пікселя зображення, що отримаємо, $A_{i+u,j+v}$ — значення пікселя вхідного зображення, $m_x - 1,\ m_y - 1$ — розмір ядра згортки.

Шари субдискретизації (Subsampling) виконують зменшення розмірності (зазвичай в кілька разів). Це можна робити різними способами, але найчастіше використовується метод вибору максимального елемента (max-pooling). Вся матриця ознак поділяється на квадрати, з яких вибираються найбільші за значенням елементи (рис. 1.6).

Рис. 1.5. Алгоритм згортки

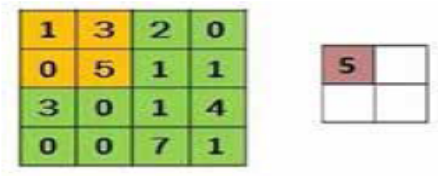

Рис. 1.6. Алгоритм субдискретизації (max poling)

Головною перевагою згорткових нейронних мереж є те, що використовуються згорткові шари, щоб виявити ознаки мережі, що дозволяє тренувати нейронну мережу без складної попередньої обробки, оскільки корисні функції будуть вивчені під час навчання. Використання згорткових нейромереж дає можливість обробляти складні нелінійні шаблони. Разом з тим вони потребують великої кількості даних, що у результаті дає точний прогноз.

1.7. Комбіновані нейронні моделі

1.7.1. CNN — LSTM

Основна ідея використання цих моделей для проблем часових рядів полягає в тому, що моделі LSTM можуть ефективно отримувати інформацію про послідовність, завдяки їх спеціальній архітектурі, тоді як моделі CNN можуть фільтрувати шум вхідних даних і виділяти ознаки, які у результаті дали б найбільш точний прогноз. Запропонована модель CNN—LSTM полягає в тому, щоб ефективно комбінувати переваги цих методів. Дана модель складається з двох компонентів: перша компонента — згортковий шар та шар субдискретизації (пулінгу) (англ. pooling layer), в яких виконуються складні математичні операції для розробки ознак вхідних даних [26], в той час як друга компонента використовує створені ознаки за допомогою LSTM.

На малюнку 1.7 наведена архітектура моделі CNN–LSTM з двома згортковими шарами, шару пулінгу, шару LSTM та вихідного шару:

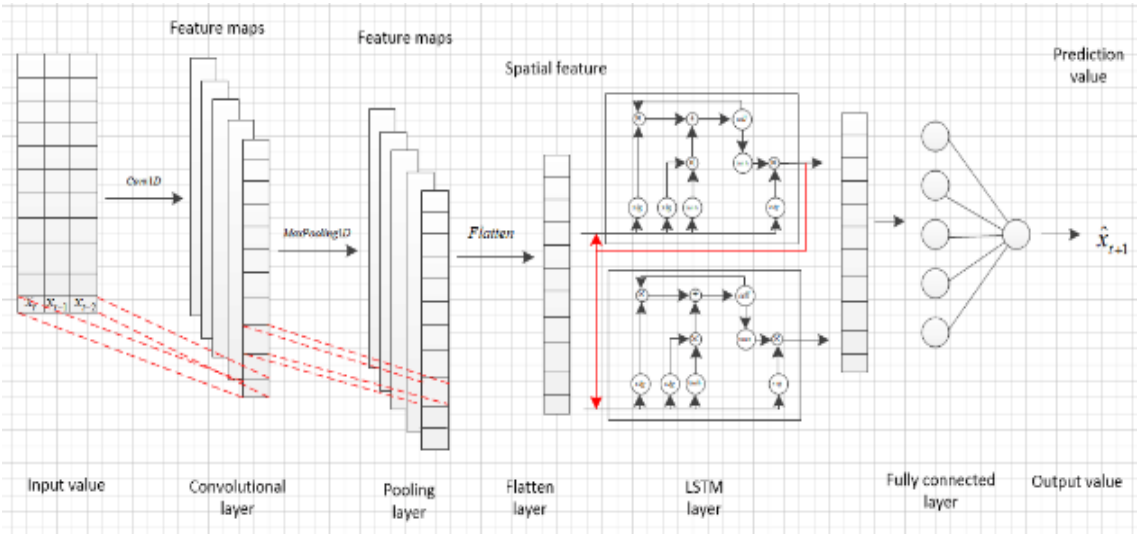

Рис. 1.7. Архітектура моделі CNN–LSTM

1.7.2. ARIMA — CNN — LSTM

Комбінована модель, яка поєднала лінійну модель ARIMA та моделі глибокого навчання: CNN і LSTM для виділення лінійних та нелінійних

ознак даних. За допомогою ARIMA спочатку обчислюються залишки цієї моделі, далі за рахунок моделі CNN вилучаються просторово-часові ознаки залишків, і отримуються довгострокові залежності цих ознак за моделлю LSTM. Зазначимо, що перевага моделі CNN полягає саме в тому, що вона може фіксувати просторові ознаки в спостереженнях.
Використання такої моделі дозволяє досягти кращої точності прогнозування порівняно з іншими моделями.

1.8. Висновки до розділу

У першому розділі був проведений аналіз предметної області. Наведено визначення часового ряду, його основні компоненти. Висвітлені загальні відомості про прогнозування часових рядів та перераховано основні міри точності прогнозів: MSE, RMSE, MAPE, MAE. Розглянуто моделі на основі експоненціального згладжування, ARIMA моделі, апарат штучної нейронної мережі, його сутність. Також розглянуто нейронні мережі, в основі яких лежить застосування глибокого навчання: рекурентні мережі – з довгою короткостроковою пам'яттю (LSTM) та згорткові (CNN). Були виділені їх переваги та недоліки. Також було розглянуто архітектуру нових комбінованих моделей CNN — LSTM та ARIMA — CNN — LSTM.

РОЗДІЛ 2

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ЗАДАЧІ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

2.1. Основні поняття в сфері криптовалют

Поява та функціонування криптовалют у світі створило нову течію досліджень у науковій сфері. Обіг криптовалют набирає обертів, що зумовлює інтерес науковців до їх сутності, аналіз якого викладено у працях А. Бейкверді [3], Г. Хілемана, М. Раухс [9], Д. Вахрушева, О. Железова. Не дивлячись на таку кількість праць, тематика криптоіндустрії та криптовалютного обігу в науковому контексті є недостатньо розроблена, зокрема більшість опублікованих матеріалів в основному мають описовий характер.

Криптовалюта – різновид цифрової валюти, створення і контроль за якою базуються на криптографічних методах. Вона має надійний захист від підробки та може використовуватись у якості засобу інвестування. Це підтверджується створенням та успішним функціонуванням бірж криптовалют, інвестиційних проектів.

Початкові ідеї створення криптовалюти почали виникати в кінці 90-х років минулого століття. Ідеологи криптовалют насамперед прагнули забезпечити повну незалежність нової грошової одиниці від держави, що гарантувало б учасникам транзакцій з криптовалютою бажану анонімність. У 2008 році світ побачив першу повноцінну криптовалюту. Сатосі Накамото (існує також версія, що це не одна людина, а група вчених під псевдонімом) висунув свою концепцію щодо даного виду валюти, розробив протокол криптовалюти Біткоїн і створив першу версію програмного забезпечення, в якому цей протокол був реалізований [24]. Вже в 2009 була запущена мережа Bitcoin. Успіх її застосування, наявність протоколу та відкритого програмного коду сприяли стрімкому зростанню кількості криптовалют та учасників, що зацікавлені в їх існуванні.

2.2. Прогнозування курсу криптовалют

Проблема прогнозування розвитку фондового або страхового ринків, на сьогодні є достатньо вивченою. Натомість прогнозування на ринку криптовалют є актуальною та цікавою задачею. Це пов'язано з тим, що ринок криптовалют характеризується високим ступенем волатильності, невизначеності та сильними коливаннями цін з часом, а самі криптовалюти знаходяться на перехідній стадії, через що є нестабільними. На сьогодні, криптовалюта — один з найновіших способів інвестування фінансів, криптовалютні торги являють собою постійно зростаючий і перспективний фінансовий ринок. Для того, щоб заробити на криптовалютному трейдингу необхідно вміти правильно спрогнозувати майбутній рух цін.

Розглядаючи тему прогнозування криптовалют згадаємо Сословського В.Г. [33], який пропонує використовувати системний підхід у дослідженні криптовалюти. Він визначає ринок криптовалюти як відкриту, складну, стохастичну систему, яка знаходиться у фазі активного формування й у зовнішньому середовищі взаємодіє з існуючими суб'єктами економіки, функціонує як складова фінансової системи, та може потенційно вплинути на її подальший розвиток.

На ринку присутні рішення задачі прогнозування курсу криптовалют, проте зазвичай вони представлені у вигляді комерційного сервісу і не розкривають деталі реалізації. Серед факторів, що впливають на курс криптовалют, дослідники відмітили розповсюдження самої цифрової валюти. В останній час біткоін отримав масові реклами у всіх світових засобах масової інформації, що безумовно сприяло зростанню курсу. Також пости зі згадуванням відповідної криптовалюти у соціальній мережі Twitter можуть вплинути на обсяг торгів BTC на наступний день [31]. Крім того, [15] встановив, що існує вплив коментарів користувачів на онлайн-платформах на коливання цін криптовалют, а саме BTC особливо корелює з числом позитивних коментарів у соціальних мережах.

Як вже було зазначено вище, однією з головних проблем криптовалют є висока волатильність, але саме через це задача прогнозування їх курсу є дуже важливою. Розглядаючи більш детально криптовалюту Bitcoin (BTC),

можемо сказати, що ціни BTC демонструють нестаціонарну поведінку, де статистичний розподіл даних змінюється з часом. На малюнку 2.1 можна побачити вартість BTC за весь період:

Рис. 2.1. Динаміка курсу Bitcoin за всю історію

Ціни на BTC в цей період були вкрай нестабільними. До 2013 року інтерес до біткоіну, його використання в віртуальних транзакціях і його ціни були низькими. А вже далі криптовалюти почали поширюватися, що викликало інтерес у людей, у результаті ціни збільшувалися. Але незважаючи на те, що ціни на BTC демонструють надзвичайну волатильність, BTC, як цифровий актив, досить стійкий, оскільки може відновити свою вартість після значних падінь, і навіть коли на ринку висока невизначеність, наприклад, під час пандемії COVID-19 [32].

2.3. Прогнозування часових рядів

Проблема прогнозування цін відноситься до аналізу часових рядів. Тому розглядаючи прогнозування курсу криптовалют з боку прогнозування часових рядів, зазначимо, що існують різні типи підходів, що базуються на моделях для прогнозування часових рядів відповідно до [12].

2.3.1. Метод використання інтегрованої моделі авторегресії

Інтегрована модель авторегресії — змінного середнього (Autoregressive Integrated Moving Average — ARIMA), мабуть, найпопулярніший метод, коли справа стосується прогнозування часових рядів. Вперше систематичний підхід до побудови моделі ARIMA був викладений Г.Е.П. Боксом і Г.М. Дженкінсом у 1976 році [5]. Популярність використання цієї моделі пояснюється тим, що ARIMA здатна забезпечити подібну та іноді навіть кращу точність порівняно з іншими методами машинного навчання [18]. До того ж, моделі ARIMA легше пояснити, зокрема, порівняно з альтернативами глибокого навчання.

У дослідженні [1] описуються результати використання ARIMA моделі для прогнозування цін на Bitcoin (BTC), Ripple (XRP) та Ethereum (ETH). Експеримент повторили для щоденних, щотижневих та щомісячних наборів даних шляхом застосування ARIMA з різними параметрами. У таблиці 2.2 узагальнено результати помилок на навчальній та тестовій виборках.

Test.	Bitcoin			XRP			Ethereum		
	D.	W.	M.	D.	W.	M.	D.	W.	M.
MAE.	313.894	764.24	1937.78	0.041	0.125	0.254	12.949	31.249	94.05
MSE	294560	1389560	10012856	0.0097	0.0681	0.1822	410.01	1712	14818
RMSE	542.735	1178.8	3164.31	0.096	0.261	0.427	20.25	41.380	121.72

Рис. 2.2. Експериментальні результати для помилок MAE, MSE та RMSE

Дослідники помітили, що використання моделі ARIMA на щоденній повторній вибірці перевершила інші моделі з точки зору MAE, MSE та RMSE для прогнозування ціни на BTC, XRP та ETH. Модель ARIMA працювала краще на щотижневих наборах даних, ніж на щомісячних значеннях для BTC та XRP з точки зору MAE, MSE та RMSE, тоді як помилка тієї ж моделі в ETH гірша через менший розмір набору даних.

Перевагами моделей прогнозування ARIMA є простота та прозорість моделювання, чітке математико-статистичне обгрунтування. Але сама побудова моделі ARIMA вимагає великих витрат ресурсів і часу. До недоліків можна віднести неадаптивність — при отриманні нових даних, модель потрі-

бно періодично переоцінювати, а іноді переідентифіковувати. Таку модель доцільно використовувати при короткотривалому прогнозуванні, може використовуватися на середньотривалому і недоцільна на довготривалому прогнозах, що було доведено у статті [1].

2.3.2. Метод використання штучних нейронних мереж

З огляду на специфіку ринку біткоінів, використання методів машинного навчання може надавати об'єктивні та адекватні результати, зважаючи на їх ефективність у схожих сферах. Численні дослідження свідчать про усе частіше використання штучних нейронних мереж (Artificial Neural Network — ANN) для аналізу та прогнозування часових рядів, зокрема у фінансовій сфері.

У роботі [10] показано, що ANN мають кращі прогностичні властивості, ніж моделі часових рядів та інші алгоритми ML для задач прогнозування фінансових показників. Разом з тим, останні дослідження показали перспективність нейромережевих моделей завдяки їх нестандартному характеру обробки інформації, що забезпечує більш ефективну роботу системи прогнозу в цілому. У разі успішного навчання, нейромережа повертає прийнятний результат на підставі даних, які були відсутні у навчальній вибірці, неповних та зашумлених даних [30]. Однією з головних переваг нейронних мереж перед традиційними алгоритмами — є можливість навчання. В процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними.
У разі застосування ANN для прогнозування ціни на біткоіни, будується мережа прогнозування тенденцій. Метою цієї мережі є визначення імовірностей таких трьох станів: у наступному періоді ціна зросте, знизиться або залишиться без змін [21].

До основних переваг нейромереж можна віднести ефективність прогнозів, можливість відтворювання складних нелінійних залежностей, адаптивність до надходження нових даних, здатність моделей до навчання. Недоліками ж є складність алгоритму навчання, ресурсомісткість процесу навчання, відсутність доступу до проміжних результатів складання прогнозу, а також можливість перенавчання.

2.3.3. Метод використання рекурентної мережі з довгою короткостроковою пам'яттю

Одним з найперспективніших сучасних технологій машинного навчання є застосування глибинних нейронних мереж, в основі яких лежить застосування глибокого навчання (англ. Deep Learning).
Рекурентна нейронна мережа (Recurrent Neural Network, RNN) — являє собою мережу із циклів, які можуть зберігати інформацію. Така нейронна мережа здатна знаходити не тільки прості взаємозв'язки, а й взаємозв'язки між взаємозв'язками. У моделі RNN внутрішній шар служить для запам'ятовування інформації з попередніх спостережень. Головною проблемою є запам'ятовування невеликої кількості попередніх спостережень, що не підходить для довгих (фінансових, економічних) періодів.

Зупинімося докладніше на LSTM мережі (Long Short-Term Memory Units) — архітектурі рекурентних нейронних мереж, запропонована 1997 року Зеппом Хохрайтером та Юргеном Шмідгубером [11].

В останні роки з'явилося багато досліджень щодо використання LSTM мережі для прогнозування курсу криптовалют. Наприклад, у [22] порівнюються LSTM і ARIMA моделі для прогнозування протягом одного, семи, тридцяти та дев'яноста днів. З результатів видно, що моделі машинного навчання (LSTM) потребують набагато більше часу для прогнозування через складні розрахунки, ніж традиційні моделі. Час компіляції моделі LSTM становить 61 мілісекунду, а для моделі ARIMA — 4 мілісекунди. Але розрахунки помилки RMSE показали, що модель машинного навчання LSTM перевершує традиційну модель ARIMA. Дослідження фокусується лише на ціні біткоіна для розробки прогнозної моделі та не бере до уваги інші економічні фактори, такі як новини про біткоіни, державну політику та ринкові настрої, з якими можна набагато точніше передбачити ціну. Отже, доведено, що моделі, розроблені з використанням LSTM, мають більшу точність, ніж традиційні моделі, тому вони є ефективними у питанні прогнозування біткоінів.

2.3.4. Метод використання згорткової нейронної мережі

Здатність згорткових нейронних мереж (Convolutional Neural Network — CNN) вивчати і автоматично отримувати ознаки з необроблених вхідних даних може застосовуватися до завдань прогнозування часових рядів. Послідовність спостережень можна розглядати як одновимірне зображення, яке модель CNN може зчитувати і розкладати на основні елементи [6]. Порівняно з іншими нейроними мережами, літератури з прогнозування часових рядів за допомогою згорткових нейромереж справді мало, оскільки ці типи мережі набагато частіше застосовуються в проблемах класифікації.

Розглянемо останню роботу, в яких дослідники займалися питанням використання CNN для прогнозування на фондовому ринку та ринку криптовалют. У [29] порівнюються архітектури LSTM, RNN та CNN для короткострокового прогнозування майбутніх цін на акції. У цьому підході архітектура CNN перевершує LSTM і RNN. Це пов'язано з раптовими змінами, що відбуваються на фондових ринках. Ці зміни можуть не завжди мати регулярний характер або не завжди йти за одним і тим же циклом. Залежно від компаній й секторів існування тенденцій, період їх існування будуть відрізнятися. Аналіз цих типів тенденцій і циклів дасть більший прибуток інвесторам. Тому дослідники вважають, що для аналізу такої інформації повинні використовувати мережі, такі як CNN, оскільки вони використовують поточну інформацію, надану в конкретний момент та здатні фіксувати різкі зміни в системі.

Головною перевагою згорткових нейронних мереж є те, що використовуються згорткові шари, щоб виявити ознаки мережі, що дозволяє тренувати нейронну мережу без складної попередньої обробки, оскільки корисні функції будуть вивчені під час навчання. Разом з тим вони потребують великої кількості даних, що у результаті дає точний прогноз.

2.3.5. Метод використання комбінованих нейронних моделей

Розглянемо популярну сучасну тенденцію створення комбінованих моделей прогнозування, наприклад, CNN— LSTM [14], ARIMA— CNN— LSTM [13]. Такий підхід дає можливість компенсувати недоліки одних моделей за допомогою інших і спрямований на підвищення точності прогнозування, як одного з головних критеріїв ефективності моделі.

2.3.5.1. CNN — LSTM

Використання комбінованої моделі, у склад якої входить мережі довгої короткострокової пам'яті (LSTM) та згорткові нейронні мережі (CNN) - це, мабуть, найбільш популярний, ефективний та широко використовуваний метод. Основна ідея використання цих моделей для проблем часових рядів полягає в тому, що моделі LSTM можуть ефективно отримувати інформацію про послідовність, завдяки їх спеціальній архітектурі, тоді як моделі CNN можуть фільтрувати шум вхідних даних і виділяти ознаки, які у результаті дали б найбільш точний прогноз. Запропонована модель CNN— LSTM полягає в тому, щоб ефективно комбінувати переваги цих методів.

У роботі [16] модель CNN— LSTM використовується для прогнозування цін на золото. Було запропоновано дві версії даної моделі, кожна з яких має два згорткових шари з різною кількістю фільтрів. Наведені результати показали, що моделі LSTM з додатковими згортковими шарами забезпечують значне покращення точності прогнозування, порівняно з традиційними моделями для прогнозування зміни цін, а також дає найменше значення середньоквадратичної помилки. У дослідженні було зазначено, що таку модель можна використовувати для прогнозування фондових ринків, цін на криптовалюту. Тому у своїй наступній роботі [17] вчені продовжили вивчати комбіновану модель CNN— LSTM, але вже для прогнозування на криптовалютному ринку. CNN— LSTM оцінювали на основі прогнозування ціни криптовалюти на наступну годину (регресія), а також на основі прогнозу, якщо ціна на наступну годину буде зростати або зменшуватися щодо поточної ціни (класифікація). Крім того, надійність кожної моделі прогно-

зування та ефективність її прогнозування оцінюється шляхом вивчення автокореляції помилок. Детальний експериментальний аналіз показує, що такий метод може бути корисними одне для одного для розробки сильних, стабільних та надійних моделей прогнозування.

2.3.5.2. ARIMA — CNN — LSTM

У 2019 році на Міжнародній конференції з інформаційних технологій та кількісних методів у менеджменті (The 7th International Conference on Information Technology and Quantitative Management) було представлено нову комбіновану модель для прогнозування цін на вуглець [13], яка поєднала лінійну модель ARIMA та моделі глибокого навчання: CNN і LSTM для виділення лінійних та нелінійних ознак даних.

У даній роботі [13] було проаналізовано архітектуру моделі ARIMA — CNN — LSTM та особливості її використання. Так, за допомогою ARIMA спочатку обчислюються залишки цієї моделі, далі за рахунок моделі CNN вилучаються просторово-часові ознаки залишків, і отримуються довгострокові залежності цих ознак за моделлю LSTM. Зазначимо, що перевага моделі CNN полягає саме в тому, що вона може фіксувати просторові ознаки в спостереженнях.

Проведено прогнозування цін на вуглець за допомогою наступних моделей: CNN, LSTM, ARIMA, ARIMA-CNN-LSTM, та обчислено оцінки точності прогнозування часових рядів: RMSE, MAPE. Отримані результати свідчать про те, що модель ARIMA — CNN — LSTM дозволяє досягти кращої точності прогнозування порівняно з іншими моделями. Для прогнозування на ринку криптовалют ця модель використовується не так часто, але на основі попередніх досліджень, можемо зробити висновок, що результат такого прогнозування буде кращим в порівнянні з використанням традиційних моделей, або нейромереж по окремості, оскільки така комбінація методів дасть змогу забезпечити точний прогноз.

2.4. Висновки до розділу

У другому розділі було розглянуто основні поняття у сфері криптовалют та їх прогнозування. Показана актуальність даної задачі, проведено огляд поточного стану проблемної області й огляд існуючих рішень.

Аналіз застосування традиційних моделей прогнозування показує, що вони спираються на припущення щодо лінійного характеру залежності між показниками та пристосовані для даних, в яких можна виділити тенденції, сезонність та шум, а криптовалюти характеризуються низькою сезонністю. Традиційні моделі не відповідають сучасним потребам, оскільки часові ряди криптовалют характеризуються високим рівнем нелінійності і нестаціонарності, тому ці методи не дуже ефективні для прогнозування цін саме криптовлют. За результатами порівняння методів прогнозування курсу криптовалют були виділені їх переваги та недоліки.

Таким чином, перспективним і потужним підходом до прогнозування стають методи обчислювального інтелекту, до числа яких, в першу чергу, слід віднести нейронні мережі. Здійснений аналіз дає змогу підтвердити перспективність та доцільність їх використання для прогнозування. Але складність вибору алгоритму навчання нейромереж та обчислювальна складність процесу самого навчання вимагають розвитку створення комбінованих моделей. Такий підхід дає можливість в одній моделі поєднати усі переваги та компенсувати недоліки одних за допомогою інших, через що точність прогнозування, як одного з головних критеріїв ефективності моделі, покращиться.

РОЗДІЛ 3

ПРОБЛЕМА ПРОГНОЗУВАННЯ ПРИ ОБМЕЖЕННІ ІСТОРИЧНИХ ДАНИХ

3.1. Обмеженість історичних даних

Досить часто можна зіткнутися з ситуацією, коли даних, необхідних для моделювання і прогнозування, недостатньо, або ці статистичні дані спотворені. В результаті, при застосуванні прогнозних моделей, може статися збій в програмі при відсутності необхідної керуючої змінної, або кінцеві дані можуть показати неприйнятний, для ситуації, що розглядається, результат. Під обмеженим масивом історичних даних ми будемо розуміти деякий набір даних, обмежених як за кількістю (за показниками), так і за датою. Іншими словами, дослідник в даній ситуації має в своєму розпорядженні не повний набір всієї необхідної інформації, а лише деякі дані, обмежені деяким проміжком часу.

Криптовалютні набори даних відрізняються повнотою опису торгів криптовалюти, які відбулися. Залежно від часових проміжків фіксації стану торгів і наявності даних за заявками виділяють кілька типів даних: Intraday data, Tick data, Time and sales data. Розглянемо перший тип:
Intraday data – загальна характеристика даних, що показує стан торгів на біржі протягом дня. На відміну від класичних бірж, торги на кріптовалютних біржах не зупиняються на ніч. Залежно від того, наскільки часто ми будемо знімати дані про стан торгів ми будемо отримувати різну ступінь повноти картини того, що відбувається на біржі. Від повноти картини залежить правильність настройки алгоритмів та їх похибка.

У даній роботі буде розглядатися 3 набори даних, які були зібрані з біржі криптовалют, у період з 1 травня 2021 року 00:00 по 1 червня 2021 року 00:00, тобто у нас є дані про курс біткоіна за місяць. Наші дані – це OHLCV, вони характеризуються тим, що показують котирування на момент відкриття (open) торгового дня (початок дня), найвищий (high) рівень цін досягнутий за день, найменший (low) рівень цін досягнутий за день, рівень

цін на момент закриття (close) торгового дня (на кінець дня), а також обсяг торгів (volume) за день.

Такі дані вважаються не зовсім повними, оскільки це дані торгів, що відбулися на біржі. Однак ці дані не включають заявки (orders) на покупку (bid orders) і продаж (sell orders), які не відбулися, а це також важливо знати. Наприклад, може складатися ситуація, коли трейдер виставив на продаж біткоіни на 200 мільйонів доларів, але продав лише на 70 мільйонів доларів. 130 мільйонів залишилися невиконаними. Зазначимо, що це важлива інформація для розуміння подальшої можливої зміни курсу. Для відображення таких ситуацій використовують іншу групу типів даних про торги на біржі.

Як було зазначено раніше, у даній роботи використовуємо 3 набори даних, а саме дані з 1, 5, 15 хвилинними таймстемпами. Обмеженість симулюємо шляхом експериментів по прогнозуванню на вибірках різної довжини - 1, 5 та 15 хвилинні кендели за місяць, два тижні, тиждень і так далі. Завдання полягає у тому, щоб дослідити, як довжина вибірки та частота дискретизації впливають на точність прогнозу.

3.2. Онлайнове машинне навчання

У ситуаціях, коли перед нами постає питання прогнозування при обмеженні або відсутності історичних даних також використовується онлайнове машинне навчання. Розглянемо основні переваги та реалізції онлайн-навчання.

Онлайнове машинне навчання (далі ОМН, онлайн-навчання) – це метод машинного навчання, який передбачає доступність даних у послідовному порядку та оновлення моделі безпосередньо перед тим, як вимагається прогнозування або після останнього спостереження, тобто наявні дані використовуються для поновлення кращого передбачення для наступних даних. ОМН є спільною технікою, що використовується в галузі машинного навчання, коли неможливе тренування на всьому наборі даних, наприклад, коли виникає необхідність в алгоритмах, які працюють із зовнішньою пам'яттю. А також використовується в ситуаціях, коли алгоритму доводиться дина-

мічно пристосовувати нові схеми в даних або коли самі дані утворюються як функція від часу, наприклад, при прогнозі цін на фондовому ринку, а у нашому випадку – на криптовалютному ринку. В онлайновому навчанні модель буде змінюватися так само часто, як змінюються дані, щоб уловити та використати ці зміни.

3.3. Різниця між оффлайн та онлайн машинним навчанням

Розглянемо детально, що відрізняє офлайн (пакетне) та онлайн-навчання. У найпростішому розумінні офлайн-навчання – це підхід, який поглинає всі дані одночасно для побудови моделі, тому кращий прогноз генерується за один раз, виходячи з повного тренувального набору даних, тоді як онлайн-навчання – це підхід, який поглинає дані по одному спостереженню за раз.

Традиційно машинне навчання виконується в автономному режимі, що означає, що у нас є пакет даних, і ми оптимізуємо рівняння

$$f(\theta) = \frac{1}{N} \sum_{i=1}^{N} f(\theta, z_i),$$

де $z_i = (x_i, y_i)$ – у випадку навчання з вчителем, або x_i – навчання без вчителя та $f(\theta, z_i)$ – якась функція втрат.
Однак, якщо у нас є потокові дані, нам потрібно проводити онлайн-навчання, щоб ми могли оновлювати свої оцінки по мірі надходження кожної нової точки даних, а не чекати «кінця» (що може ніколи не відбутися) [23].

Алгоритми пакетного навчання мають ряд критичних недоліків через необхідність навчати модель з нуля при отриманні нових даних: низька ефективність за часом і пам'яттю, погана масштабованість для великих систем. Онлайн-навчання вирішує ці проблеми, оскільки модель оновлюється на основі даних, що надходять в кожен момент часу.
Завдяки цьому алгоритми онлайн-навчання є набагато ефективнішими в додатках, де дані не тільки мають великий розмір, але і надходять з високою

швидкістю [28].

При онлайн-навчанні для побудови моделі необхідний один прохід за даними, що дозволяє не зберігати їх для подальшого доступу в процесі навчання і використовувати менший обсяг пам'яті. Однак зміна виду вхідних даних, вихід сервера з ладу і багато інших причин можуть призвести до некоректної роботи системи. Оцінити якість роботи системи при онлайн-навчанні складніше, ніж при пакетному: немає можливості отримати репрезентативний тестовий набір даних, тому єдиний варіант, у найзагальнішому випадку, – поглянути на те, наскільки добре працює алгоритм останнім часом. Алгоритми онлайн-навчання широко використовуються електронною комерцією та індустрією соціальних мереж, оскільки є можливість фіксувати будь-яку нову тенденцію, помітну з часом.

Offline Learning	Features	Online Learning
Less complex as model is constant	**Complexity**	Dynamic complexity as the model keeps evolving over time
Fewer computations, single time batch-based training	**Computational Power**	Continuous data ingestions result in consequent model refinement computations
Easier to implement	**Use in Production**	Difficult to implement and manage
Image Classification or anything related to Machine Learning - where data patterns remains constant without sudden concept drifts	**Applications**	Used in finance, economics, heath where new data patterns are constantly emerging
Industry proven tools. E.g. Sci-kit, TensorFlow, Pytorch, Keras, Spark Mlib	**Tools**	Active research/New project tools: E.g. MOA, SAMOA, scikit-multiflow, streamDM

Рис. 3.1. Порівняння між онлайн та офлайн-навчанням

Зазначимо, що порівнювати онлайн-моделі із пакетними моделями насправді не чесно, оскільки вони не призначені для вирішення одних і тих же проблем. Моніторинг моделі – це безперервний процес. Якщо продуктивність моделі починає погіршуватися, то потрібно її реструктуризувати, щоб адаптуватися до постійно мінливого набору даних. У машинному навчанні на виробничому рівні моделі повинні орієнтуватися на надання правильної інформації користувачеві або його застосуванню.

Разом з тим, можна провести аналогію з градієнтним спуском, а саме:

офлайн-навчання подібне до пакетного (batch) градієнтного спуску, коли на кожній ітерації навчальна вибірка переглядається цілком, тільки після чого змінюється. Такий підхід потребує великих обчислювальних затрат. Онлайн-навчання, навпаки, є аналогом стохастичного (stochastic/online) градієнтного спуску, коли на кожній ітерації алгоритму з навчальної вибірки випадковим чином обирається лише один об'єкт [4].

3.4. Стохастичний градієнтний спуск

Одним із прикладів онлайн-навчання є стохастичний або онлайн-градієнтний спуск.

Означення 3.1. [34] Стохастичний градієнтний спуск (англ. stochastic gradient descent – SGD) – ітеративний метод оптимізації градієнтного спуску за допомогою стохастичного наближення. Використовується для прискорення пошуку цільової функції шляхом використання обмеженого за розміром тренувального набору, який вибирається випадкового при кожній ітерації.

Розробку метода приписують Герберту Роббінсу та Саттону Монро, які описали його у статті 1951 року «Метод стохастичного наближення» [27].
У стохастичному градієнтному спуску, істинний градієнт $Q(w)$ апроксимується градієнтом по одному об'єкті

$$w := w - \eta \nabla Q_i(w).$$

Ідея стохастичного градієнтного спуску полягає у використанні однієї вибірки випадковим чином для оновлення градієнта за ітерацію, а не безпосереднього обчислення точного значення градієнта. Стохастичний градієнт є неупередженою оцінкою реального градієнта [27].
З точки зору оптимального використання обчислювальних ресурсів алгоритм стохастичного градієнта дуже цікавий.

3.4.1. Алгоритм стохастичного градієнтного спуску

Вхід:

- X^n – навчальна вибірка.
- η – темп навчання.
- λ – параметр згладжування Q.

Вихід:

- Вектор ваг w.

Алгоритм:

1) Ініціалізувати ваги w_j, $(j = 0, \ldots, k,$ де $k-$розмірність простору ознак.
2) Ініціалізувати поточну оцінку функціоналу:

$$Q := \sum_{i=1}^{n} L(a(x_i,w), y_i);$$

3) Повторювати, поки значення Q не стабілізується та/або ваги w не припинять змінюватись:
   а) Вибрати об'єкт x_i із X^n (наприклад, випадковим чином);
   б) Обчислити вихідне значення алгоритму $a(x_i,w)$ та помилку:

$$\varepsilon_i := L(a(x_i,w),y_i);$$

   в) Зробити крок градієнтного спуску:

$$w := w - \eta L'_a\left(a(x_i,w),y_i\right)\varphi'\left(\langle w,x_i\rangle\right)x_i;$$

   г) Оцінити значення функціоналу:

$$Q := (1-\lambda)Q + \lambda\varepsilon_i;$$

3.4.2. Переваги та недоліки методу

Перевага методу стохастичного градієнта в порівнянні з пакетним полягає в тому, що даний алгоритм дозволяє реалізувати онлайн-навчання.

Метод стохастичного градієнта дозволяє використовувати вибірки надвеликих розмірів для навчання. Також в методі можна варіювати сам алгоритм навчання - якщо вибірка надто велика, то після перегляду об'єкту з неї цей об'єкт можна видаляти з вибірки і тим самим оптимізувати використання ресурсів комп'ютерів. Якщо ж вибірка занадто мала, то можна вибирати об'єкти з поверненням і тим самим виконувати бутстреппінг вибірки.
Однак метод стохастичного градієнта не позбавлений недоліків, зокрема:

- можлива повільна збіжність алгоритму;
- якщо функціонал $Q(w)$ має велику кількість екстремальних точок, то є велика ймовірність застрявання алгоритму в локальних мінімумах. Для боротьби з цим використовується метод випадкової модифікації вектора w, внаслідок якої одна або кілька координат вектора трохи змінюються і тим самим знижується ймовірність застрявання.

3.4.3. Різновиди і модифікації

Існує безліч модифікацій алгоритму стохастичного градієнтного спуску:

- Неявний стохастичний градієнтний спуск (англ. implicit stochastic gradient descent, ISGD) – є варіантом стандартного SGD, за допомогою якого ми використовуємо неявні оновлення в основному оновленні алгоритму. Це додає чисельної стабільності та стійкості до специфікації швидкості навчання, що є відомою проблемою стандартного методу.
- Імпульс або SGD з імпульсом (Momentum) – це метод, який допомагає прискорити вектори градієнтів у правильних напрямках, що призводить до більш швидкого зближення. На відміну від класичного SGD, метод намагається зберегти просування в тому ж напрямку, запобігаючи коливання. Імпульс успішно використовувався для навчання штучних нейронних мереж протягом декількох десятиліть.
- AdaGrad (англ. Adaptive gradient algorithm) – адаптивний градієнтний алгоритм, є модифікацією стохастичного алгоритму градієнтного спуску з окремою для кожного параметра швидкістю навчання. Ця стратегія збільшує швидкість збіжності в порівнянні зі стандартним стохастическим методом градієнтного спуску в умовах, коли дані рід-

кісні і відповідні параметри більш інформативні. Прикладами таких додатків є обробка природних мов і розпізнавання образів.

- Adam (скорочення від Adaptive Moment Estimation) – це оновлення оптимізатора RMSProp. У цьому оптимізаційному алгоритмі використовуються ковзаючі середні як градієнтів, так і других моментів градієнтів. Він поєднує в собі і ідею накопичення руху, і ідею слабшого поновлення ваг для типових ознак.

3.5. Висновки до розділу

У даному розділі було розглянуто криптовалютні набори даних, а саме один з його типів – Intraday data. Один з прикладів таких даних – є OHLCV (Open, High, Low, Close, Volume), саме такий набір даних буде використовуватися для прогнозування курсу криптовалют. Також було описано, як саме буде моделюватися обмеженість даних у цій роботі.

Крім того, було проаналізовано метод машинного навчання – онлайн-навчання, який дає змогу вдало прогнозувати на основі обмеженних даних. Перевагами цього методу є обчислювальна швидкість та ефективність, більш легке реалізування в порівнянні з пакетним машинним навчанням. Онлайн-навчання широко використовується в ситуаціях, коли самі дані утворюються як функція від часу, наприклад, при прогнозі цін на фондовому ринку, тому використання цього методу при вирішенні проблеми прогнозування курсу криптовалют може дати прийнятні результати, навіть за наявності обмеженних або відсутніх даних.

Було наведено приклад онлайн-навчання – стохастичний градієнтний спус, представлено алгоритм цього метода, а також приведено переваги та недоліки стохастичного градієнтного спуску. Перевага методу полягає в тому, що даний алгоритм дозволяє використовувати вибірки надвеликих розмірів для навчання та варіювати сам алгоритм навчання, у випадках коли вибірка надто велика, або надто мала. Серед недоліків виділено можливу повільну збіжність алгоритму.

РОЗДІЛ 4

ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

4.1. Вибір мови програмування та бібліотек

У цьому розділі буде наведено обчислювальні результати, пов'язані з навчальними процесами, а також детальний опис реалізації дослідження та архітектури програмного продукту.

В якості мови програмування була обрана мова Python. Перевагами саме такого вибору стали наявність широкого вибору бібліотек та фреймворків для роботи з даними, стислість та читабельність коду (що особливо доцільно для задач машинного навчання).

Для прогнозування курсу криптовалюти біткоін на конкретну дату було обрано метод глибинного навчання, а саме: рекурентна нейрона мережа з довгою короткостроковою пам'яттю (LSTM).

Для реалізації мережі LSTM було застосовано бібліотеку методів глибинного навчання Keras, тому що там шари додаються поступово, а не визначають всю мережу відразу. Так ми можемо швидко змінювати кількість і тип шарів, оптимізуючи нейронну мережу.

Зауважимо, що наш аналіз буде зосереджений лише на інформації про ціни, залишаючи осторонь будь-який фактор, який може вплинути на ціну біткоіна, як, наприклад, новини, які можуть відігравати дуже важливе правило. З цієї причини, треба сказати, що відповідний проект призначений лише для навчальних цілей і не доречно використовувати його у трейдерстві. Це надто спрощена модель, яка допоможе зрозуміти прогнозування часових рядів з використанням Python та рекурентних нейронних мереж, а також методу онлайн-навчання та проаналізувати вплив довжини вибірки та частоти дискретизації на точність прогнозу.

4.2. Вхідні дані

Існує досить багато ресурсів, які ми можемо використовувати для отримання історичних даних про ціни на біткоіни. Деякі з цих ресурсів дозволяють користувачам завантажувати файли CSV вручну, інші пропонують API, який можна підключити до його коду.

У роботі використовуються 3 набори даних Intraday data – OHLCV, вони характеризуються тим, що показують котирування на момент відкриття (open) торгового дня (початок дня), найвищий (high) рівень цін досягнутий за день, найменший (low) рівень цін досягнутий за день, рівень цін на момент закриття (close) торгового дня (на кінець дня), а також обсяг торгів (volume) за день.

У даній роботі буде розглядатися 3 набори даних, які були зібрані з біржі криптовалют, у період з період з 1 травня 2021 року 00:00 по 1 червня 2021 року 00:00, тобто у нас є дані про курс біткоіна за місяць. Наші набори різняться періодичністю збору даних: 1, 5, 15 хвилинні таймстемпи. Обмеженість симулюємо шляхом експериментів по прогнозуванню на вибірках різної довжини - 1, 5 та 15 хвилинні кендели за місяць, два тижні, тиждень і так далі. Завдання полягає у тому, щоб дослідити, як довжина вибірки та частота дискретизації впливають на точність прогнозу.

4.2.1. Data Exploration – дослідження даних

Першим чином, завантажимо наші набори даних за допомогою бібліотеки Pandas, на малюнку нижче можемо побачити, як виглядають дані з 15-хвилинним інтервалом.

	Date	Time	Open	High	Low	Close	Volume	Datetime
0	01/05/21	00:15	57798.77	58179.45	57514.39	58086.08	540	2021-01-05 00:15:00
1	01/05/21	00:30	58086.08	58195.50	57893.67	58128.64	176	2021-01-05 00:30:00
2	01/05/21	00:45	58128.63	58141.71	57850.00	57891.70	173	2021-01-05 00:45:00
3	01/05/21	01:00	57891.70	57988.37	57734.15	57905.85	134	2021-01-05 01:00:00
4	01/05/21	01:15	57905.85	57905.85	57563.20	57716.37	229	2021-01-05 01:15:00

Рис. 4.1. Вхідні дані з 1-хвилинними інтервалом

Можна об'єднати стовпці "Date"та "Time"в один новий стовпець "Datetime". Отримаємо:

	Date	Time	Open	High	Low	Close	Volume	Datetime
0	01/05/21	00:15	57798.77	58179.45	57514.39	58086.08	540	2021-01-05 00:15:00
1	01/05/21	00:30	58086.08	58195.50	57893.67	58128.64	176	2021-01-05 00:30:00
2	01/05/21	00:45	58128.63	58141.71	57850.00	57891.70	173	2021-01-05 00:45:00
3	01/05/21	01:00	57891.70	57988.37	57734.15	57905.85	134	2021-01-05 01:00:00
4	01/05/21	01:15	57905.85	57905.85	57563.20	57716.37	229	2021-01-05 01:15:00

Рис. 4.2. Оновлені дані

Для наших цілей нас цікавить стовпець *Close*, який містить значення цін біткоінів на кінець дня, для цієї конкретної дати. Тепер побудуємо графік, який показує нам зміну ціни біткоіна протягом місяця, використовуючи наш набір даних.

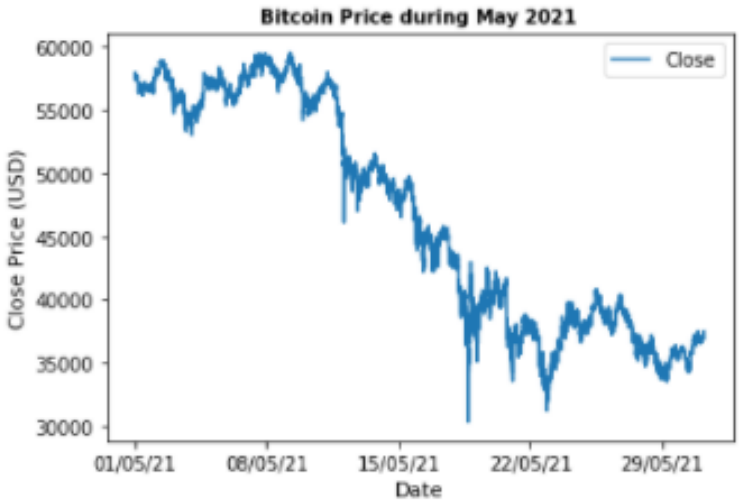

Рис. 4.3. Зміна курсу Bitcoin протягом травня 2021 року

Найвища ціна за цей період – 59575.78$, 10.05.21 о 04:06
Найнижча ціна за цей період – 30305.31$, 19.05.21 о 13:10

Тепер перевіримо, чи містяться у стопчику *Close* будь-які нульові значення, нульові значення нам не корисні, і ми повинні їх замінити, якщо такі є. Зробимо це за допомогою функції $.info()$, яка використовується для отримання короткого огляду набору даних. Це дуже зручно при дослідницькому аналізі даних. Перевіривши, отримали, що нульових значень немає, тому продовжуємо працювати з даними.

4.2.2. Data Preprocessing – попередня обробка даних

- Крок 1: Нормування

Ціни криптовалют, що використовуються в обчисленнях, часто збільшуються з часом, що призводить до виходу більшості значень в тестовому наборі за допустимі межі. Виходить так, що мережа повинна передбачити значення, які вона ніколи не бачила раніше. Очевидно, що рано чи пізно модель перестає вести себе адекватно. Метою нормування є зміна значень числових стовпців у наборі даних до загальної шкали, не спотворюючи відмінностей в діапазонах значень.

Тому в першу чергу для адекватної роботи мережі в цілому, потрібно унормувати дані. Для цього завдання з бібліотеки scikit-learn був викликаний клас MinMaxScaler. Для кожного значення в датасеті MinMaxScaler віднімає мінімальне значення з датасета і ділить на діапазон - різницю між вихідним максимумом і мінімумом (він становить від 0 до 1). Використання даного класу не випадково – MinMaxScaler дозволяє об'єкту "запам'ятовувати"атрибути даних, в які був поміщений. Це не призводить до значної зміни інформації, вбудованої в вихідні дані.

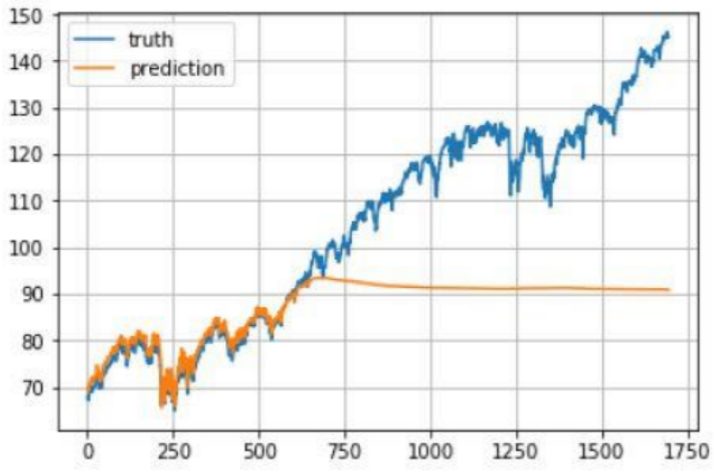

Рис. 4.4. Приклад використання ненормованих даних

Для адекватної роботи мережі також важливо центрування даних - наявність у даних середнього значення і середньоквадратичного відхилення. Відбувається це наступним чином: віднімається середнє значення, а потім цей результат ділиться на стандартне відхилення. Спочатку ця процедура відбувається на тренувальному наборі даних, але потім цю властивість по-

трібно перенести на тренувальну вибірку. Однак, необхідно використовувати ті ж два параметра, які використовувалися для центрування навчального датасета. Функція $fit\_transform()$ з бібліотеки scikit-learn, використовувана в цій роботі, допомагає зробити ці дві операції одночасно (обчислити і застосувати до вибірки ці параметри).

- Крок 2: Розбиття даних

На цьому кроці ми фактично вирішимо 2 проблеми, перша полягає в тому, що нам потрібно розділити наш набір даних на тренувальний та тестовий набори даних. Тренувальний набір даних використовується для навчання моделі, тоді як тестовий набір використовується для забезпечення об'єктивної оцінки кінцевої моделі (як базова лінія для порівняння наших прогнозів). Це дуже важливо, оскільки потрібно переконатись, що наші прогнози мають сенс, але ми не можемо перевіряти ті самі дані, якими ми навчаємо нашу мережу, оскільки ми можемо зіткнутися з ризиком перенавчання. Для тренувальних даних візьмемо 90%, для тестових – 10%.

- Крок 3: Формування даних

Завершальним етапом попередньої обробки даних є переформування вхідного масиву, щоб він був сумісним із шарами LSTM.

4.3. Побудова моделі LSTM

Після підготовки наших даних настав час для побудови мережі, яку ми згодом будемо навчати, використовуючи нормалізовані дані. Для цього, в першу чергу, необхідно завантажити пакети даних з бібліотеки Keras:

- Sequential використовується в якості графа шарів - моделі послідовної мережі.
- Dense застосовується для створення прихованих (повнозв'язних) шарів мережі.
- LSTM призначений для створення шарів з довгою короткостроковою пам'яттю для довготривалого зберігання інформації.

Створення першого LSTM-шару мережі відбувалося наступним чином:

Аргументом $units$ задавалося число внутрішніх блоків LSTM-мережі.

Налаштування *return_sequences* визначає, чи повертати останній висновок як результат роботи програми, або "відправити цей висновок далі". Для цього аргументу був обраний параметр True, так як передбачається подальше включення додаткових шарів LSTM-типу (False для зупинки).

Вхідний сигнал для кожного шару LSTM повинен мати три розмірності (Терміни з англомовної літератури):

1) Sample. Один ряд - один приклад (sample). Batch – пакет даних, складається з одного або декількох прикладів.
2) Time Step. Часовий крок - точка спостереження в прикладі (наприклад, один день).
3) Feature. Ознака (особливість) - спостереження в тимчасовій крок.

Ставлячи по черзі, як аргументи в *input_shape*, ці величини можна визначити вхідний LSTM-шар. Наприклад, *input_shape* = (5,3) означає, що мережа очікує як мінімум один приклад, 5 тимчасових кроків і 3 ознаки.

Функція активації є важливим параметром шару, який визначає вихідне значення нейрона в залежності від результату зваженої суми входів і порогового значення. У моєму випадку функція активації – *sigmoid*.

Варіюючи кількість шарів мережі, в результаті доводиться застосувати параметр False для *return_sequences* (останній, вихідний шар) і аргумент $units = 1$ для шару Dense.

Далі варто вибрати алгоритм оптимізації моделі навчання, використовуваний для знаходження ваг або коефіцієнтів, і функцію втрат loss в якості параметрів методу compile. Для даної задачі був використаний оптимізатор "*Adam*". Даний вид оптимізації являє собою метод стохастичного градієнтного спуску, заснований на оцінці моментів першого і другого порядків. З документації Keras: цей метод ефективний в обчисленнях, має невелику потребу в пам'яті. Мета функції втрат – обчислити помилку, яку модель повинна прагнути мінімізувати в процесі навчання. В рамках даного дослідження в якості опції втрат була обрана середньоквадратична помилка моделі (*mean squared error*).

Використовуючи fit можна почати тренування мережі, попередньо поставивши параметри *batch_size* (підмножина навчальної вибірки фіксо-

ваного розміру (кількість прикладів)) і *epochs*. Епохою називається одна ітерація в процесі навчання мережі, тобто весь датасета один раз пройшов через мережу в прямому і зворотному напрямку. Так як неможливо пропустити весь датасет разом, то його ділять на *batches* - маленькі партії. Дуже важливо коректно оцінити необхідну кількість епох, адже їх мала кількість призводить до недонавчання, а надто велика - до перенавчання.

Тепер можна приступити безпосередньо до прогнозування, скориставшись $predict()$. Задіявши $inverse\_transform()$, результати прогнозування були отримані в абсолютних одиницях (долар США).

Для спрощення є сенс привести в таблицю параметри роботи нейронної мережі:

Кількість шарів LSTM	1
Кількість блоків шару LSTM (units)	4
Кількість повнозв'язних шарів (Dense)	1
Функція активації	Sigmoid
Оптимізатор	Adam
Функція втрат	Mean Squared Error
Batch-size	5
Кількість епох	5

Табл. 4.1. Параметри моделі

4.4. Результати, отримані за допомогою LSTM

Розглянемо набір даних з 1-хвилинним інтервалом. Припустимо, ми можемо робити прогноз на крок вперед. Хочемо спрогнозувати ціну на 15 хвилин вперед. Спочатку поекспериментуємо з прогнозом, коли беремо похвилинну ціну за місяць, два тижні, тиждень, день, 6 годин, 3 години.

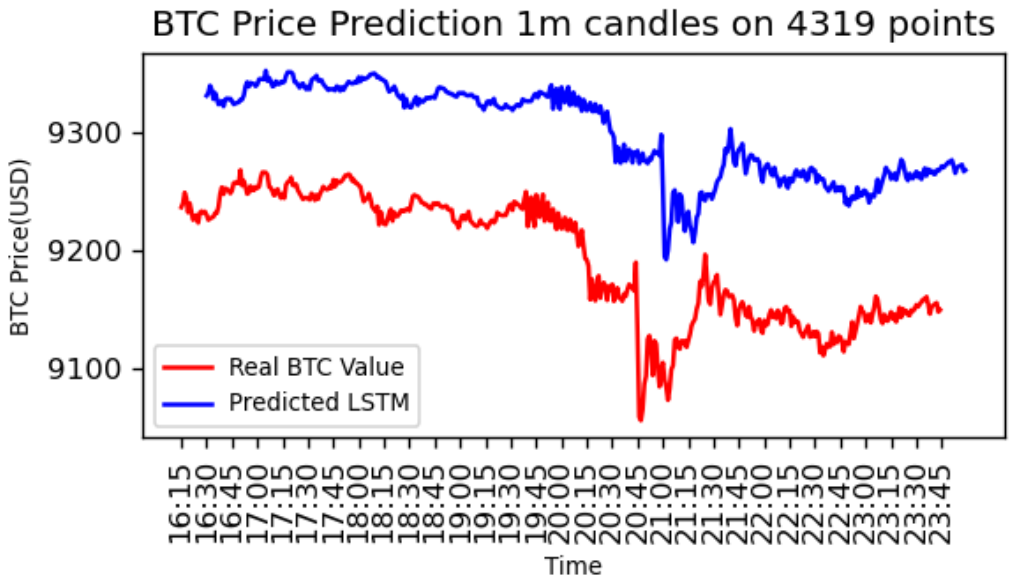

Рис. 4.5. В якості вхідних даних – похвилинна ціна за місяць

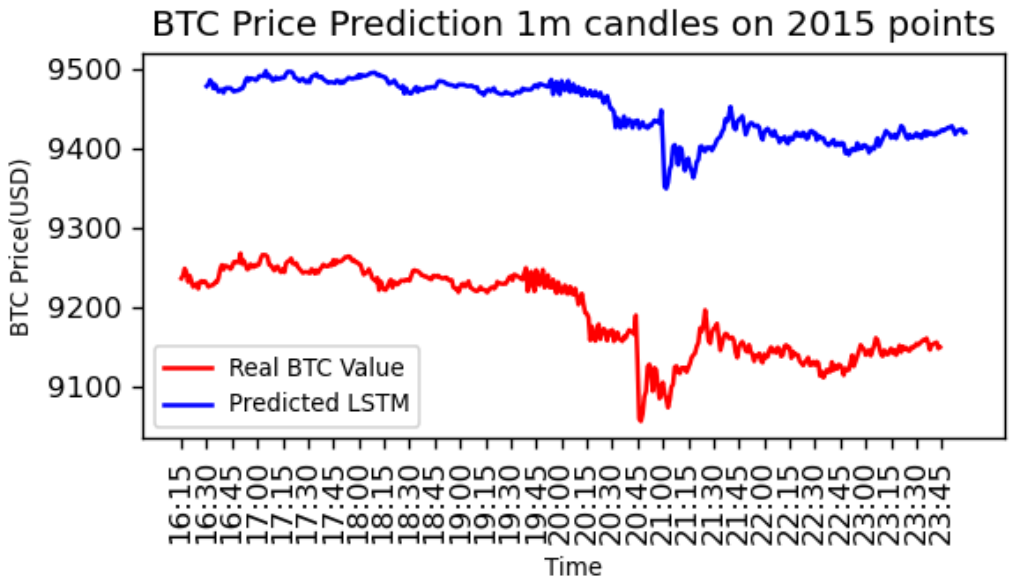

Рис. 4.6. Вхідні дані – похвилинна ціна за 2 тижні

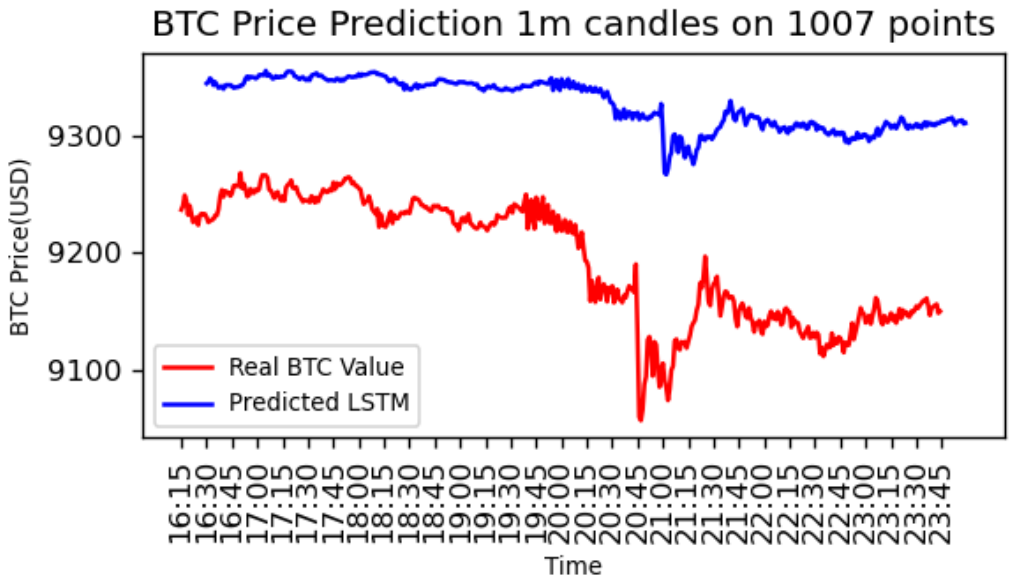

Рис. 4.7. Вхідні дані – похвилинна ціна за тиждень

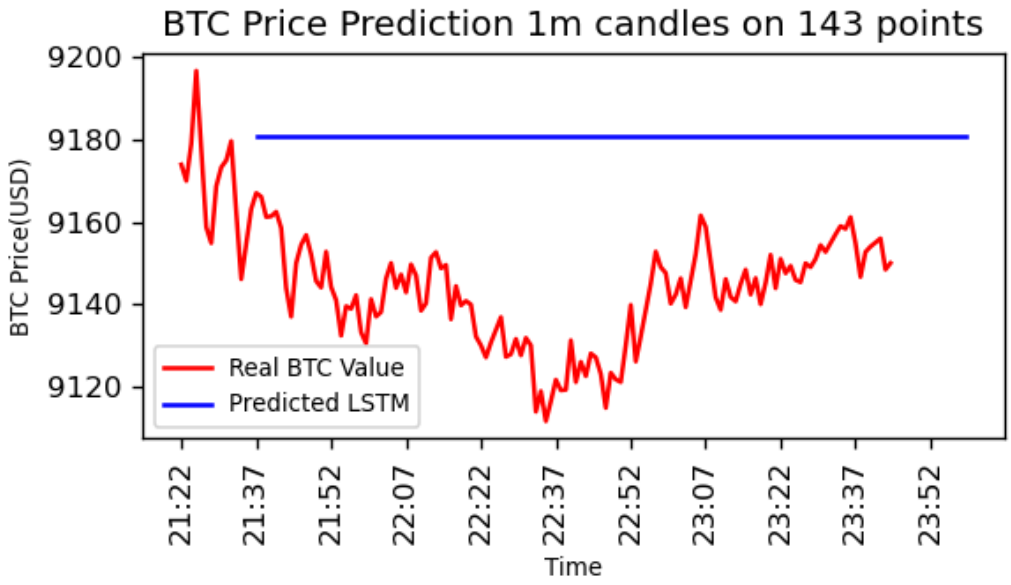

Рис. 4.8. Вхідні дані – похвилинна ціна за день

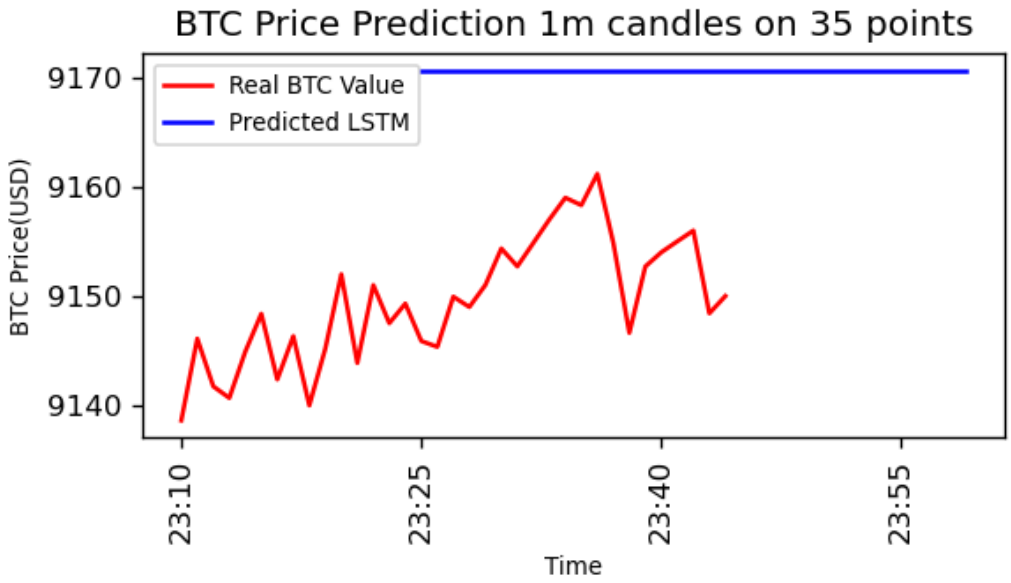

Рис. 4.9. Вхідні дані – похвилинна ціна за 6 годин

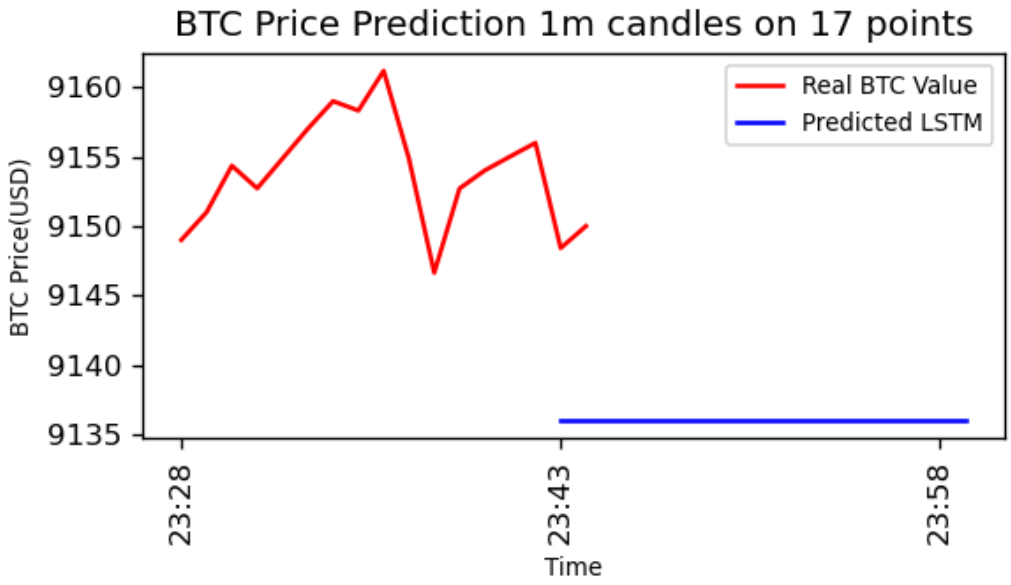

Рис. 4.10. Вхідні дані – похвилинна ціна за 3 години

Дивлячись на графіки, можемо зробити висновок, що для якісного та точного прогнозування, треба брати вибірки більшої довжини. Якщо є обмеженність даних, як у нашому випадку, то краще за все брати вибірку за 30 днів. Оскільки так ми отримаємо більш-менш прийнятний результат. Нагадаємо, що у цих випадках наша модель працювала за 5 ітерацій.

Спробуємо змінити кількість ітерацій на 100, отримаємо наступне:

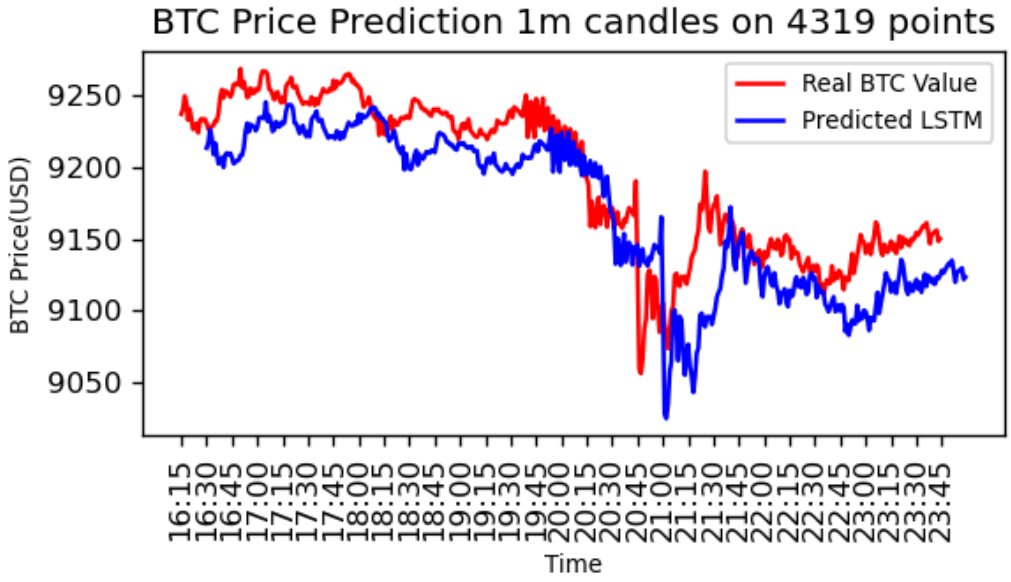

Рис. 4.11. Вхідні дані – похвилинна ціна за місяць:

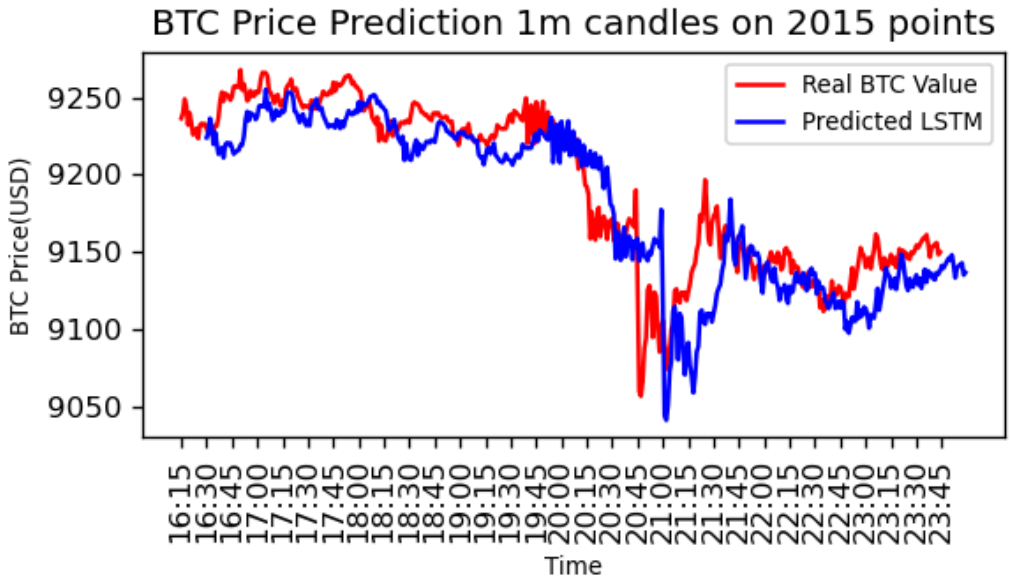

Рис. 4.12. Вхідні дані – похвилинна ціна за 2 тижні

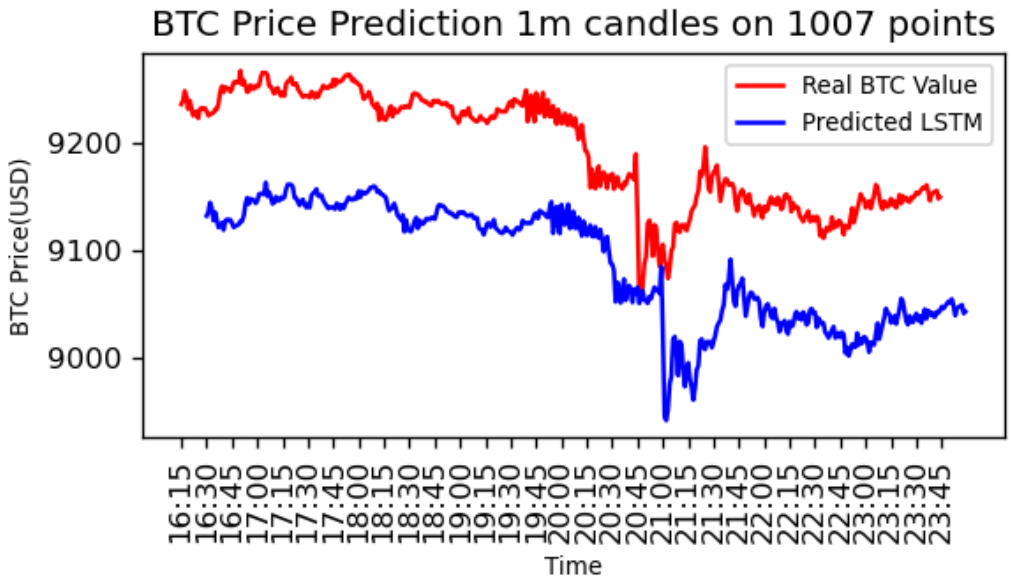

Рис. 4.13. Вхідні дані – похвилинна ціна за тиждень

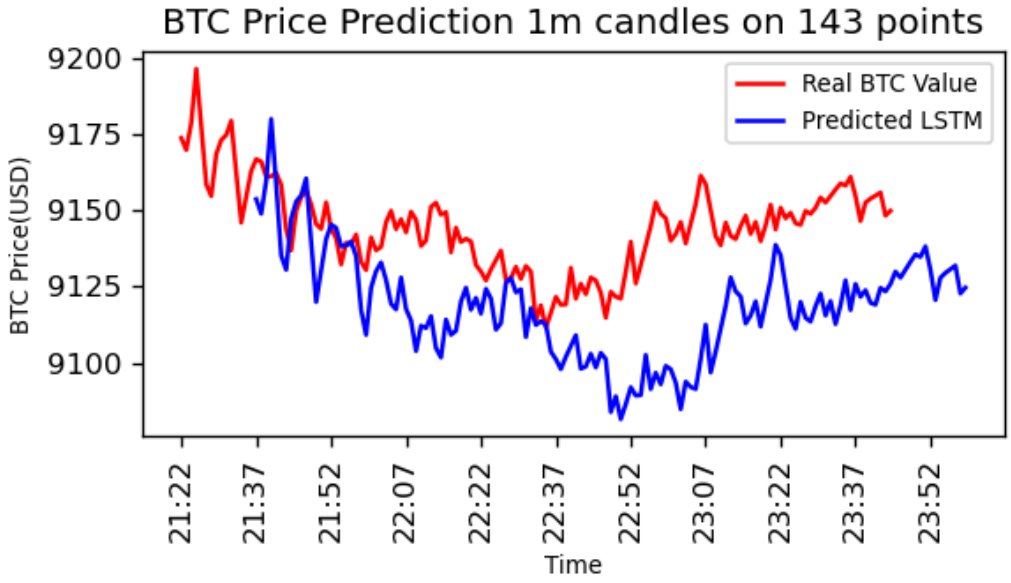

Рис. 4.14. Вхідні дані – похвилинна ціна за день

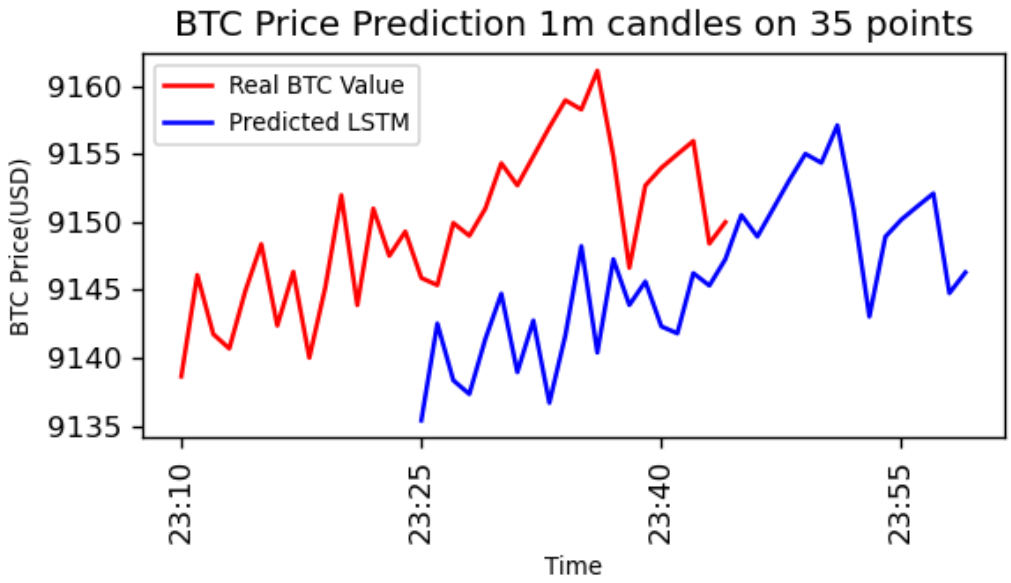

Рис. 4.15. Вхідні дані – похвилинна ціна за 6 годин

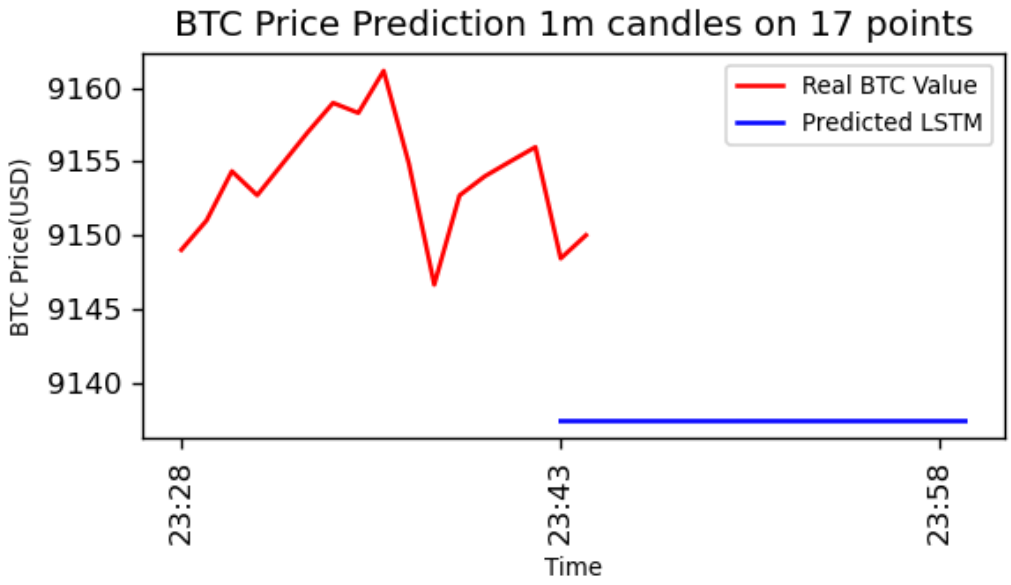

Рис. 4.16. Вхідні дані – похвилинна ціна за 3 години

Отримані результати свідчать про те, що збільшена кількість епох сприяла більш точному прогнозуванню, але все одно похибка існує.

Тепер порівняємо прогноз, коли беремо похвилинні відліки і робимо на них прогноз на 15 хвилин вперед - попередній результат прогнозу використовується на наступному кроці у вхідних даних, тобто буде 15 циклів.

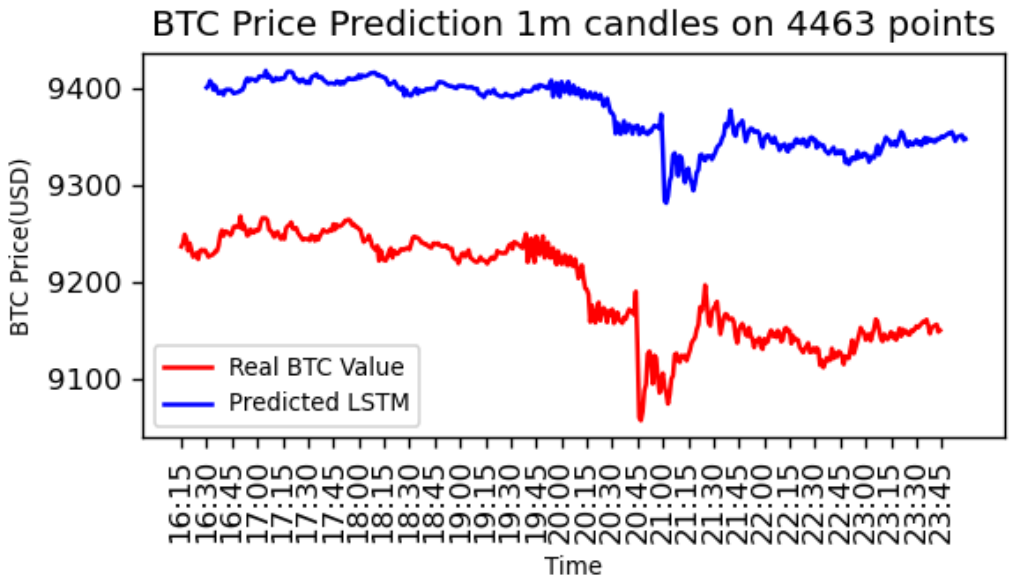

Рис. 4.17. Прогнозування на 15хв з похвилинними відліками

Потім беремо 5-хвилинні відліки і робимо прогноз на 15-хвилин (3 цикли):

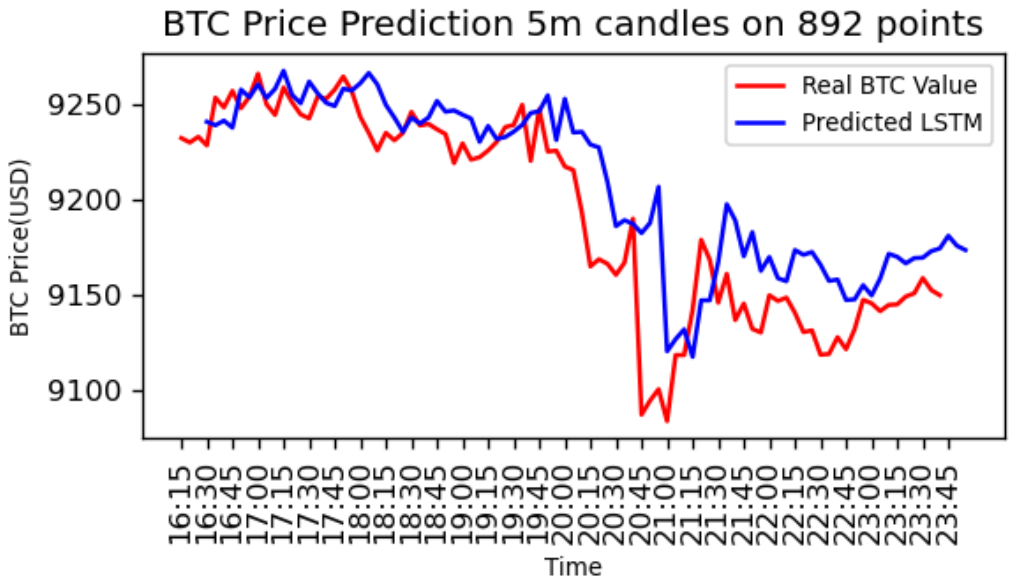

Рис. 4.18. Прогнозування на 15хв з 5-хвилинними відліками

15-хвилинні - 1 прогноз:

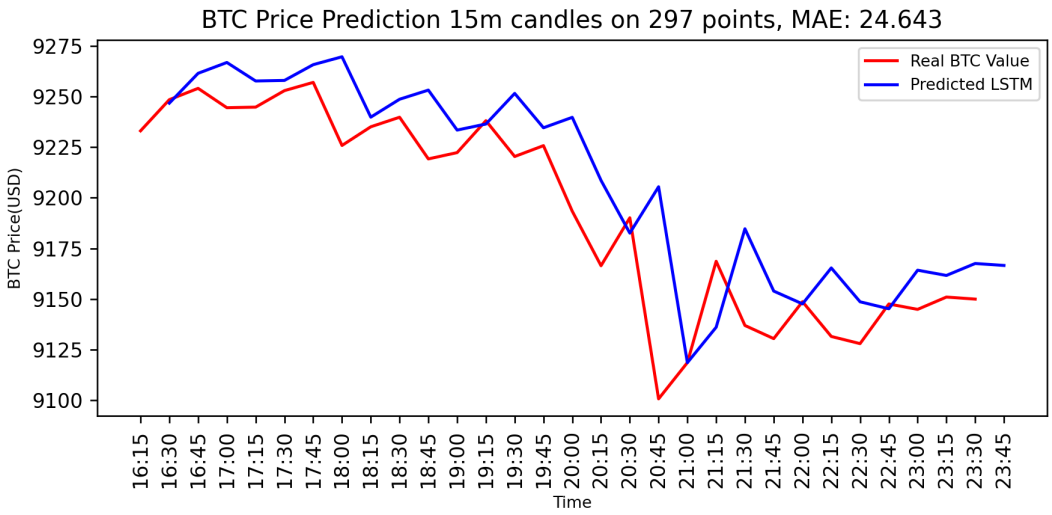

Рис. 4.19. Прогнозування на 15хв з 15-хвилинними відліками

Для оцінки ефективності моделі використовуються такі показники: середня абсолютна похибка (MAE), середньоквадратична похибка (MSE) та корінь із середньоквадратичної похибки (RMSE). Зазначимо, що значення MAE та RMSE розраховуються в тих самих одиницях, в яких представлені вхідні дані (у нашому випадку, USD). Найкращим вважається випадок з найнижчими значеннями MAE, MSE та RMSE. Наприклад, у контексті ціни Bitcoin, MAE 25 означає, що прогнозована ціна становить ± 25 доларів США від фактичної ціни. Модель, яка іноді передбачає помилкові значення, буде мати більш високе значення RMSE, хоча у неї все ж може бути більш низьке MAE. Таким чином, моделі слід оцінювати за всіма трьома показниками.

Дивлячись на таблицю, наведену нижче, не викликає жодних сумнівів, що, чим більше раз попередній результат прогнозу використовується на наступному кроці у вхідних даних, тим більше накопичується помилка. Можемо дійти висновку, що краще за все прогнозувати на 1 точку вперед.

Відліки	MAE	MSE	RMSE
1-хвилинні	193.02012	38400.02604	195.95925
5-хвилинні	25.18074	1267.73265	35.60523
15-хвилинні	24.64351	1176.45336	34.29947

Табл. 4.2. Оцінка точності прогнозування

4.5. Висновки до розділу

Четвертий розділ присвячений розробці програмного коду, що прогнозує ціни на криптовалюту.

У цьому розділі було проаналізовано інструменти, що підходять для вирішення поставленої задачи та було обрано мову програмування Python і бібліотеку машинного навчання Keras. Для прогнозування курсу криптовалюти біткоін було обрано метод глибинного навчання: рекурентну нейрону мережу з довгою короткостроковою пам'яттю (LSTM).

Було детально описано дані, які використовуються, проведено їх попередню обробку. Далі створено тренувальний та тестовий набори. Наступним кроком був вибір параметрів моделі нейронної мережі з комірками LSTM та проведене її навчання на створених наборах.

Проведено декілька експериментів, щоб дізнатися, чи залежить точніть прогнозу від довжини вибірки. Спочатку поекспериментували з прогнозом, коли беремо похвилинну ціну за місяць, два тижні, тиждень, день, 6 годин, 3 години. Зробили висновок, що для кращої роботи моделі потрібно брати вибірки більшої довжини.

Крім того, було порівняно прогноз, коли беремо похвилинні відліки і робимо на них прогноз на 15 хвилин вперед - попередній результат прогнозу використовується на наступному кроці у вхідних даних, потім беремо 5-хвилинні та 15-хвилинні відліки. Отримані результати дають змогу дійти висновку, що чим більше раз попередній результат прогнозу використовується на наступному кроці у вхідних даних, тим більше накопичується помилка, тому краще прогнозувати на 1 точку вперед.

ВИСНОВКИ

У результаті проведеної роботи була досягнута поставлена мета, а саме було написано програмний код для прогнозування курсу криптовалюти біткоін та виявлено залежність точності прогнозу від довжини часового ряду та частоти дискретизації.

Поставлена мета зумовила необхідність вирішення таких завдань: було розглянуто основні поняття в сфері криптовалют та виконано аналіз актуальності задачі прогнозування курсу криптовалют, яка підтвердила високу популярність з боку трейдерів. Виявлено основні фактори, що впливають на зміну курсу криптовалют. Проведено огляд поточного стану прогнозування на ринку криптовалют й огляд існуючих досліджень та рішень.

Після аналізу існуючих підходів прогнозування часових рядів щодо зручності, часозатрат та ефективності прогнозу, було виділено 5 основних з них:

- Метод використання інтегрованої моделі авторегресії (ARIMA)
- Метод використання штучної нейронної мережі (ANN)
- Метод використання рекурентної мережі з довгою короткостроковою пам'яттю (LSTM)
- Метод використання згорткової нейронної мережі (CNN)
- Метод використання комбінованих нейронних моделей

У результаті аналізу, для прогнозування було обрано підхід, який базується на використанні рекурентної мережі з довгою короткостроковою пам'яттю (LSTM). Такий вибір зумовлен тим, що така мережа здатна обробляти цілі послідовності даних, тому добре підходить для обробки і побудови прогнозів на основі часових рядів.

Проведені експерименти дають змогу дійти висновку, що для прогнозування цін за допомогою мережі LSTM, необхідно мати більші за довжиною вибірки. Якщо постає питання прогнозування на 15 хвилин вперед, краще використовувати дані з таким ж відліком, оскільки в іншому випадку потрібно буде використовувати попередній результат прогнозу на наступному кроці у вхідних даних, що погано впливає на роботу моделі.

Таким чином, прогнозування ціни криптовалют за допомогою LSTM не може бути достатньо якісним та точним для прийняття рішення інвестування, оскільки на фондовий ринок впливає багато факторів: пости зі згадуванням відповідної криптовалюти у соціальній мережі, розповсюдження самої цифрової валюти у всіх світових засобах масової інформації, вплив коментарів користувачів на онлайн-платформах.

Майбутня робота зосереджена саме на розробці точної та надійної системи для прогнозування курсу криптовалют, у якій використовуються дані соціальних мереж як зовнішній чинник, що впливає на курс. Для вирішення цієї проблеми буде проведено аналіз впливу кількості обговорень криптовалюти у соціальній мережі Twitter на її курс та запропоновано підхід до побудови системи прогнозування курсу криптовалют з використанням цих даних. У такому випадку, цю систему буде доречно використовувати для прийняття рішень у трейдерстві. Також буде використано стохастичний градієнтний спуск, один з прикладів онлайн-навчання, для порівняння з прогнозом, отриманим за допомогою LSTM. Більш детально буде проаналізовано використання комбінованих моделей для прогнозування, оскільки такий підхід дає можливість в одній моделі поєднати усі переваги та компенсувати недоліки одних за допомогою інших, через що точність прогнозування, як одного з головних критеріїв ефективності моделі, покращується.

СПИСОК ЛІТЕРАТУРИ

1. Alahmari, S. (2019). Using Machine Learning ARIMA to Predict the Price of Cryptocurrencies.
2. Bakar, N. A., & Rosbi, S. (2017). Autoregressive integrated moving average (ARIMA) model for forecasting cryptocurrency exchange rate in high volatility environment: A new insight of bitcoin transaction. International Journal of Advanced Engineering Research and Science, 4(11).
3. Beikverdi, A. (2015). Is China the Primary Driving Force in Bitcoin?. Cointelegraph. Retrieved from https://cointelegraph.com/news/is-china-the-primary-driving-force-in-bitcoin.
4. Bottou, L. & Bousquet, O. (2012). Optimization for Machine Learning. The Tradeoffs of Large Scale Learning. Cambridge: MIT Press, 351–368.
5. Box, G., Reinsel, G., & Jenkins, G. (1976). Time Series Analysis: Forecasting and Control.
6. Brownlee, J. (2018). Deep Learning for Time Series Forecasting. Predict the Future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.
7. Gusarov, V. M. (2001). Statistika: Uchebnoe posobie dlja vuzov. [Statistics: Textbook for Universities]. [in Russian]
8. Greenberg, A. (2011). Crypto Currency. Retrieved from https://www.forbes.com/forbes/2011/0509/technology-psilocybin-bitcoins-gavin-andresen-crypto-currency.html#6110e19c353e.
9. Hileman, G., & Rauchs, M. (2017). Global Cryptocurrency Benchmarking Study. United Kingdom, Cambridge: Cambridge Centre For Alternative Finance.
10. Hitam, N., & Ismail, A. (2018). Comparative Performance of Machine Learning Algorithms for Cryptocurrency Forecasting. Indonesian Journal Of Electrical Engineering And Computer Science, 11(3), 1121. https://doi.org/10.11591/ijeecs.v11.i3.pp1121-1128
11. Hochreiter, S. & Schmidhuber, J. (1997). Long Short-term Memory. Neural Computation, 9(8), 1735–1780.

12. Hyndman, R., & Athanasopoulos, G. (2018). Forecasting: principles and practice (2nd ed.). OTexts, Melbourne, Australia. https://otexts.com/fpp2/.

13. Ji, L., Zou, Y., He, K., & Zhu, B. (2019). Carbon futures price forecasting based with ARIMA-CNN-LSTM model. Procedia Computer Science, 162, 33-38. https://doi.org/10.1016/j.procs.2019.11.254

14. Kim, T., & Cho, S. (2019). Predicting residential energy consumption using CNN-LSTM neural networks. Energy, 182, 72-81. https://doi.org/10.1016/j.energy.2019.05.230

15. Kim, Y., Kim, J., Kim, W., Im, J., Kim, T., Kang, S., & Kim, C. (2016). Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies. PLOS ONE, 11(8), e0161197. https://doi.org/10.1371/journal.pone.0161197.

16. Livieris, I., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. Neural Computing And Applications. ttps://doi.org/10.1007/s00521-020-04867-x

17. Livieris, I., Pintelas, E., Stavroyiannis, S., & Pintelas, P. (2020). Ensemble Deep Learning Models for Forecasting Cryptocurrency Time-Series. Algorithms, 13(5), 121. https://doi.org/10.3390/a13050121

18. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. PLOS ONE, 13(3), e0194889. https://doi.org/10.1371/journal.pone.0194889

19. Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Networks, 16(5-6), 555-559. https://doi.org/10.1016/s0893-6080(03)00115-1

20. McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115–133 https://doi.org/10.1007/BF02478259

21. McNally, S. (2016). Predicting the price of Bitcoin using Machine Learning. National College of Ireland.

22. Mudassir, M., Bennbaia, S., Unal, D., & Hammoudeh, M. (2020). Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach. Neural Computing And Applications.

https://doi.org/10.1007/s00521-020-05129-6

23. Murphy, K., (2013). Machine Learning: A Probabilistic Perspective. Cambridge, Mass.: MIT Press, 261-262.
24. Nakamoto, S. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf.
25. Odesa I. I. Mechnikov National University. (2021). 77th reporting student scientific conference of Odesa I.I. Mechnikov National University. Odesa, 104.
26. Rawat, W., & Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Computation, 29(9), 2352-2449.
27. Robbins, H. & Monro, S. (1951). A Stochastic Approximation Method. The Annals of Mathematical Statistics, 22(3), 400-407.
28. Russell, S. & Norvig, P. (2015). Artificial Intelligence: A Modern Approach. 3rd ed. Harlow: Pearson, 753.
29. Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V., & Soman, K. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. 2017 International Conference On Advances In Computing, Communications And Informatics (ICACCI). https://doi.org/10.1109/ICACCI.2017.8126078
30. Schmidhuber, J. (2015) Deep Learning in Neural Networks: An Overview. Neural Networks, ISSN: 0893-6080, Vol: 61, pp. 85-117.
31. Shen, D., Urquhart, A., & Wang, P. (2019). Does twitter predict Bitcoin?. Economics Letters, 174, 118-122. https://doi.org/10.1016/j.econlet.2018.11.007
32. Shukla, S., & Dave, S. (2020). Bitcoin beats coronavirus blues. The Economic Times. Retrieved from https://economictimes.indiatimes.com/markets/stocks/news/bitcoin-beats-coronavirus-blues/articleshow/75049718.cms.
33. Soslovskij, V., & Kosovskij, I. (2016). Rinok kriptovaljut jak sistema. [Cryptocurrency market as a system]. Visnik Universitetu Bankivs'koï Spravi — Bulletin of the University of Banking, 236-246.
34. Taddy, M. (2019). Stochastic Gradient Descent. Business Data Science:

Combining Machine Learning and Economics to Optimize, Automate, and Accelerate Business Decisions. New York: McGraw-Hill, 303-307.

35. The National Technical University «Kharkiv Polytechnic Institute». (2020). "Problems of informatics and modeling Twentieth International Science and Technology Conference, Karolino-Bugaz, 52.
36. Van Rossum, G. & Drake, F. (2006). The Python Language Reference Manual. Bristol, United Kingdom: Network Theory Limited.