ДОДАТОК А

Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії

1. Leshchynskyi Volodymyr. Principles of explanation in e-commerce system based on sales dynamics. COMPUTER AND INFORMATION SYSTEMS AND TECHNOLOGIES KHARKIV, APRIL 2020, p.76-77.

2. Назаров О.С. Теорія прогнозування: навч. посіб. – Харків: ХНУРЕ, 2017. – 300 с.

3. Smelyakov K., Smelyakov S., Chupryna A. Advances in Spatio-Temporal Segmentation of Visual Data. Chapter 1. Adaptive Edge Detection Models and Algorithms. Series Studies in Computational Intelligence (SCI), Vol. 876. – Publisher Springer, Cham, 2020. – P. 1-51. DOI https://doi.org/10.1007/978-3-030-35480-0.

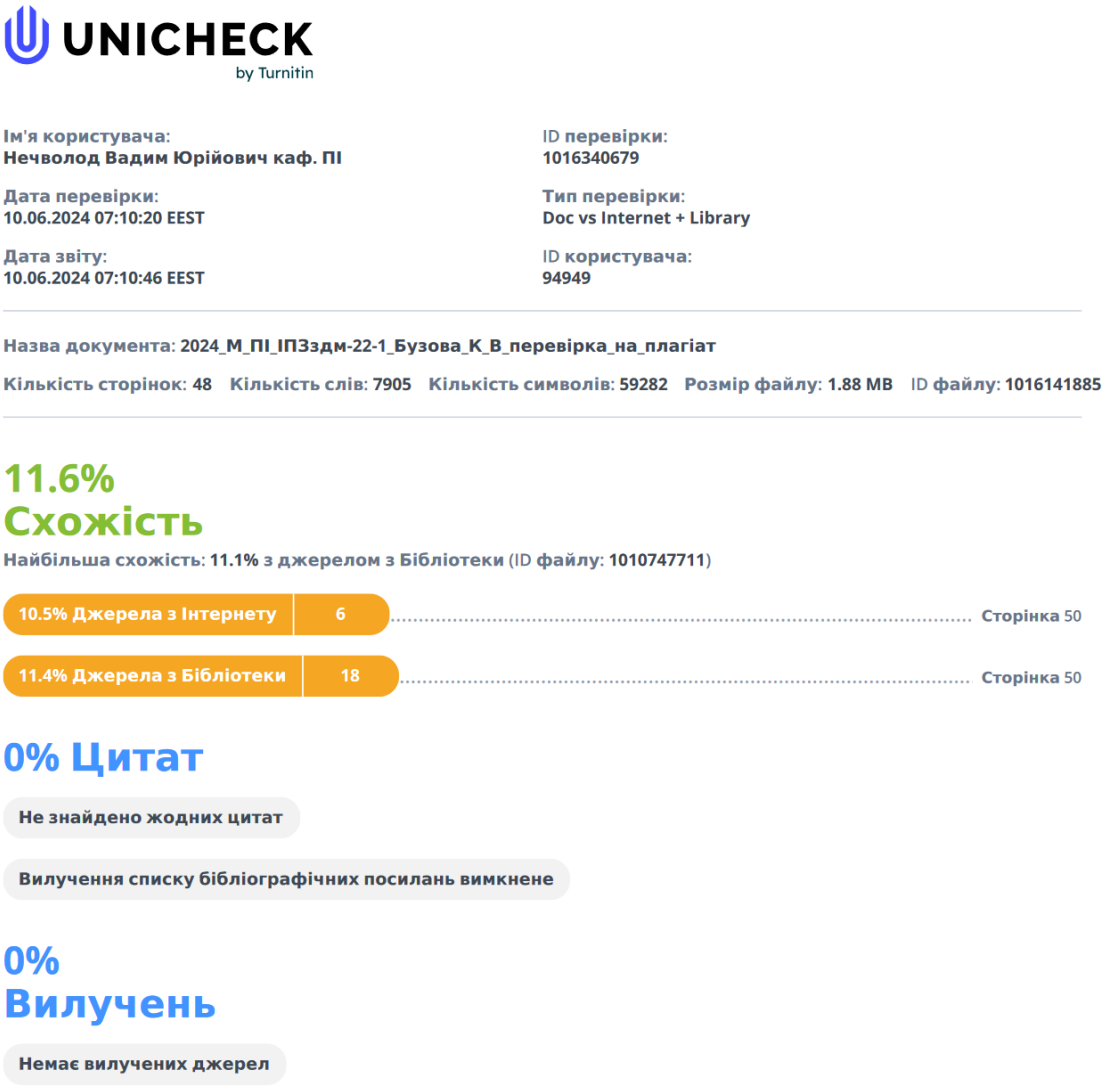
4. Korneliuk Olga, Khirova Viktoria. Features of forecasting modern stock market crises // Випуск 38. 2020 - Науковий вісник Херсонського державного університету, Херсон.- 2020 - С. 12.

5. Ponomarenko O.A. Time series forecasting with neural networks //Тези доповідей міжнародної науково-технічної конференції «Інженерія програмного забезпечення 2018» 04 – 08 червня 2018 р. – Київ, С.47

URL: https://ej.journal.kspu.edu/index.php/ej/article/view/642/633 (дата звернення 05.05.2024).

ДОДАТОК Б

Звіт результатів перевірки кваліфікаційної роботи на унікальність тексту



UNICHECK
by Turnitin

Ім'я користувача:
Нечволод Вадим Юрійович каф. ПІ

ID перевірки:
1016340679

Дата перевірки:
10.06.2024 07:10:20 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
10.06.2024 07:10:46 EEST

ID користувача:
94949

Назва документа: 2024_М_ПІ_ІПЗздм-22-1_Бузова_К_В_перевірка_на_плагіат

Кількість сторінок: 48 Кількість слів: 7905 Кількість символів: 59282 Розмір файлу: 1.88 MB ID файлу: 1016141885

11.6%
Схожість

Найбільша схожість: 11.1% з джерелом з Бібліотеки (ID файлу: 1010747711)

10.5% Джерела з Інтернету 6 ..... Сторінка 50

11.4% Джерела з Бібліотеки 18 ..... Сторінка 50

0% Цитат

Не знайдено жодних цитат

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

ДОДАТОК В

Програмний код

Код тренування моделі

```
import os
import uuid
import math
import json
from datetime import datetime, timedelta
import dateutil.parser
from pathlib import Path
from loguru import logger
import numpy as np
import pandas as pd
from sqlitedict import SqliteDict
from sklearn.preprocessing import MinMaxScaler
from sklearn.utils import class_weight
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import Dense, Flatten, Conv1D, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import utils
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
import optuna
from defaults import *

def train_model(self):
        logger.info('Training started.')

        features = list(self._df.columns)
        features.remove(self._timestamp_label)
        features.remove(self._dependent_variable)

        storage_name = os
.path
.join(self._temp_path, 'storage_{}'.format(self._thread_id))
        self._trial_storage = SqliteDict(storage_name, autocommit=True)

        train_start = 0
```

```
        validation_end = len(self._df) - 1
        validation_start = validation_end - self._config['validation_size']
        train_end = validation_start - 1

        self._df_train = self._df.loc[train_start:train_end].copy()
        self._df_validation                                              =
self._df.loc[validation_start:validation_end].copy()

        # Get rid of intermediate results
        if self._config['verbose_optuna'] == 0:
            optuna.logging.set_verbosity(optuna.logging.WARNING)

        # Initiate study
        study = optuna.create_study(directions=['maximize', 'maximize'])
        study.optimize(self._objective,
n_trials=self._config['max_trials'])

        # Process the result
        scores = [sum(item.values) for item in study.best_trials]
        best_trial_number = scores.index(max(scores))
        best_score = study.best_trials[best_trial_number].number
        best_params = study.best_trials[best_trial_number].params
        best_values = study.best_trials[best_trial_number].values

        # Log best results
        logger.info('Best score: {}'.format(best_values))
        logger.info('Best parameters: {}'.format(json.dumps(best_params)))

        # Save trained model and metadata
        with open(self._model_name, 'wb') as model_file:
            model_file.write(self._trial_storage[best_score]['model'])
        metadata = {SEQUENCE_SIZE: best_params[SEQUENCE_SIZE],
                    TRAINED: datetime.utcnow().isoformat(),
                    START_DAY: str(self._first_date),
                    END_DAY: str(self._last_date),
                    MODEL_EXPIRATION: self._model_expiration}
        with open(self._metadata_name, 'w') as meta_file:
            json.dump(metadata, meta_file)
```

```
# Clear the storage
self._trial_storage.close()
os.remove(storage_name)

logger.info('Training completed.')
```

ДОДАТОК Г

Апробація результатів робот

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ

МАТЕРІАЛИ 28-го МІЖНАРОДНОГО МОЛОДІЖНОГО ФОРУМУ

«РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ»

16-18 квітня 2024 р.
том 7

КОНФЕРЕНЦІЯ «КОМП'ЮТЕРНИЙ ЗІР, СИСТЕМНИЙ АНАЛІЗ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ»

Харків 2024

АЛФАВІТНИЙ ПОКАЖЧИК

УДК 004.94:004.02:336.7 DOI: https://doi.org/10.30837/ITF.CVSAMM.2024.166

ПРОГРАМНІ РІШЕННЯ ДЛЯ ДОПОВНЕННЯ ТРЕЙДИНГОВОЇ ТОРГІВЛІ

Бузова К.В.
Науковий керівник – канд. техн. наук, доц. Назаров О.С.
Харківський національний університет радіоелектроніки, каф. ПІ,
м. Харків, Україна
e-mail: kristina.buzova.cpe@nure.ua

The purpose of this work is to study and evaluate various methods and technologies applicable for the development of trading forecasting systems in the stock market. It examines the different types of neural networks suitable for solving such problems, existing implementations and strategies for improving software solutions, and the possibility of combining their use with conventional algorithmic trading methods. By combining different approaches, it is possible to improve the efficiency of the system as a whole, since it is not always possible to ensure optimal operation of systems using neural networks due to the specifics of time series and factors that may not be taken into account during modeling, in this case, solutions using trading algorithms can save the system from excess losses.

Дослідження методів доповнення алгоритмічної акційної торгівлі з застосуванням машинного навчання є дуже актуальним напрямком досліджень у фінансовій сфері. Машинне навчання може бути використане для прогнозування зміни цін, виявлення патернів на ринку, розробки торгових стратегій та оптимізації управління портфелем [1].

Основні напрямки досліджень у цій області включають:

– прогнозування цінових рухів, використання алгоритмів машинного навчання, таких як нейронні мережі, дерева рішень, ансамблеві методи та інші, для аналізу історичних даних цін та ринкових показників для прогнозування майбутніх цінових рухів;

– виявлення патернів на ринку, машинне навчання може допомогти виявити складні патерни та зв'язки в цінових динаміках, які можуть бути важко виявити людиною;

– розробка торгових стратегій, на основі прогнозів зміни цін та виявлених патернів, можна розробляти торгові стратегії, які автоматично виконують угоди на ринку;

– оптимізація управління портфелем, машинне навчання може бути застосоване для оптимізації алгоритмів управління портфелем, зокрема для розподілу активів, стратегій ребалансування та управління ризиками.

Для проведення досліджень у цій області можуть використовуватися різні методи машинного навчання [2], а також статистичні та фінансові моделі. Важливою частиною дослідження є валідація моделей на історич-

них даних та їх тестування на реальних ринках для перевірки ефективності та придатності до використання в реальному середовищі акційної торгівлі.

Найбільш популярні типи нейронних мереж у цьому контексті включають:

– мережі зворотного поширення помилки;

– рекурентні нейронні мережі (RNN);

– нейронні мережі з тимчасовою затримкою (TDNN).

Рекурентні нейронні мережі часто застосовуються в прогнозуванні цін на акції, оскільки вони здатні враховувати попередні значення у часовому ряді.

Оскільки ціни на акції представляють собою часові ряди, для їх прогнозування застосовується так званий "ковзне вікно". Для цього потрібна нейронна мережа з довгою короткостроковою пам'яттю (LSTM).

Звісно, не обов'язково реалізувати такі моделі власноруч, оскільки існують готові рішення на базі різноманітних бібліотек штучного інтелекту. Однією з найбільш помітних інструментальних засобів у цій сфері є бібліотека TensorFlow [3], яка дозволяє зосередитися на розробці бізнес-логіки та архітектури нейронних мереж. До переваг цієї бібліотеки можна також віднести можливість перегляду графу навченої нейронної мережі, а також можливість використовувати мову програмування Python.

Для полегшення інтеграції коду, що використовує TensorFlow, з рішеннями, що використовують алгоритмічні підходи варто зосередитись на бібліотеках, що також базуються на мові програмування Python. Одними з найпопулярніших в цій сфері можна виділити наступні продукти:

– Backtrader. Ця бібліотека для розробки, тестування та автоматизації торгівельних стратегій, має вбудовану підтримку для оптимізації параметрів стратегій.

– PyAlgoTrade. Ця бібліотека має простий у використанні API та підтримує різні типи індикаторів.

– Zipline. Це бібліотека Python, розроблена для алгоритмічної торгівлі та фінансового дослідження. Вона має вбудовану підтримку для історичних даних та може бути використана для розробки та тестування різних торгівельних стратегій.

Комбінація використання різник підходів може дозволити отримати більш стабільні результати.

Список використаних джерел:

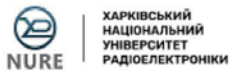
1. Marcos Lopez de Prado. Advances in Financial Machine Learning. Wiley, 2018. 393 с.

2. Назаров О.С. Теорія прогнозування: навч. посіб. Харків, 2017. 300 с.

3. Нишант Ш. Машинное обучение и TensorFlow. Питер, 2019. 336 с.

ДОДАТОК Д

Слайди презентації



Дослідження методів доповнення алгоритмічної акційної торгівлі з застосуванням машинного навчання

Бузова К.В., ІПЗздм-22-1
Науковий керівник:
доцент кафедри ПІ, Колесников Д.О.

20 червня 2024

1

Дослідження

Зараз набувають популярність автоматичні рішення на основі штучного інтелекту

На даний час існує доволі багато програмних рішень для підвищення ефективності акційної торгівлі

Практичний інтерес становить можливість застосування різних підходів в автоматизації акційної торгівлі

2

Огляд літератури

1. Leshchynskyi Volodymyr. Principles of explanation in e-commerce system based on sales dynamics. COMPUTER AND INFORMATION SYSTEMS AND TECHNOLOGIES KHARKIV, APRIL 2020, p.76-77.
2. Назаров О.С. Теорія прогнозування: навч. посіб. – Харків: ХНУРЕ, 2017. – 300 с.
3. Korneliuk Olga, Khirova Viktoria. Features of forecasting modern stock market crises // Випуск 38. 2020 - Науковий вісник Херсонського державного університету, Херсон.- 2020 - С. 12.
4. Marcos Lopez de Prado – Advances in Financial Machine Learning. – Wiley, 2018. – 393 с.
5. Zura Kakushadze and Juan Andr Serur – 151 Trading Strategies, 2018. 361 с.
6. Stefan Jansen – Machine Learning for Algorithmic Trading – Packt, 2020. – 800 с.
7. Ponomarenko O.A. Time series forecasting with neural networks //Тези доповідей міжнародної науково-технічної конференції «Інженерія програмного забезпечення 2018» 04 – 08 червня 2018 р. – Київ, С.47 URL: https://ej.journal.kspu.edu/index.php/ej/article/view/642/633 (дата звернення 05.05.2024).

Постановка задачі

- Прогнозування фондового ринку має значення не тільки для отримання прибутку окремими учасниками, але й для підвищення ефективності та стабільності фінансового сектору та економіки в цілому через збільшення ліквідності. Це створює більше можливостей як для тих, хто потребує капіталу, так і для тих, хто хоче примножити свої інвестиції. Метою цієї роботи є порівняння існуючих методів та інструментів для прогнозування фінансових трендів і цінових рухів.
- Кінцевим результатом буде порівняльний аналіз програмних комплексів для прогнозування цін на фінансових ринках, оцінка їх швидкості роботи та рекомендації щодо їх використання.

Методологія

- Рішення задачі складається з кількох етапів:

– аналіз різних методів глибинного навчання, бібліотек та інструментів, які доступні у відкритому доступі, для можливого застосування
– пошук даних для навчання і тестування моделей
– програмна реалізація моделі
– візуальне представлення результатів

Архітектура система для проведення експериментального дослідження

Основними компонетами розробленої системи є джерело даних та Jupiter в якості рушія для запуску різних моделей.

Окремим застосунком з більш практичним підходом є докеризований застосунок, що має апі для отримання «свіжих даних» для перенавчення і створення нової моделі, та апі виклику прогнозу.

Опис програмного забезпечення, що було використано у дослідженні

Для побудови експериментів була використана мова Python і рущій Jupiter.

У якості основної бібліотеки машинного навчання була використана бібліотека TensorFlow.
У якості рушія запуску та розповсюдження готового рішення використовується docker.

Для побудови веб застосунку та інтеграції готового рішення використовувався ферймворк groovy and grails.

Зміст проведеного експерименту

В якості джереа данних були взяті історичні дані по комнанії Tesla за останній рік з сайту https://finance.yahoo.com/quote/TSLA/history?p=TSLA.

Основними критеріями роботи розроблених рішень була прибутковість та час виконання

У кожному експерименті виконуються наступні етапи:

- форматування даних відповідно до вимог моделі;
- розподіл даних на тренувальні, валідаційні й тестувальні набори у співвідношенні 8:1:1. Історичні дані поділяються по часу таким чином: перші 80% для тренування, наступні 10% для валідації та останні 10% для тестування;
- виконання валідаційного тренування. На кожній епосі перевіряється, чи натренувалася модель, шляхом оцінки фінансової ефективності на валідаційній частині датасету. Тренування зупиняється, якщо протягом певної кількості епох не досягнуто кращої моделі;
- тестування отриманої моделі на тестувальній частині датасету та в реальних умовах.

Зміст проведеного експерименту

В якості критерію ефективності моделей дослідження була використана фінансова ефективність, що розраховується за формулою:

$$\varphi = \frac{\max(0, b)}{a} - 1$$

Щоб уникнути ситуацій коли коефіцієнт може приймати значення -1, що значить повну втрату капіталу, був введений ефективності з урахуванням величини капіталу протягом усього періоду.

$$\varphi_{acc} = \frac{\int_s^f max(0, b(t))dt}{a \cdot (f - s)} - 1$$

- де $b(t)$ – функція, що відображає величину капіталу в часі;
- s – час початку торгів;
- f – час завершення торгів;
- a – початковий капітал.

Опис шарів результуючої моделі

```
self.model = Sequential()
self.model.add(Conv1D(filters=filters,
                      kernel_size=kernel_size,
                      activation='relu',
                      input_shape=(sequence_size, n_features)))
self.model.add(Flatten())
self.model.add(Dropout(dropout))
self.model.add(Dense(dense_layer_size, activation='relu'))
self.model.add(Dense(2, activation='softmax'))
self.model.compile(optimizer=Adam(learning_rate=learning_rate),
                   loss='binary_crossentropy',
                   metrics=['accuracy'])
```

Створюється об'єкт моделі Sequential з бібліотеки Keras

Додається згортковий шар (Conv1D) з заданою кількістю фільтрів і розміром ядра згортки. Функція активації — ReLU.
Вхідна форма даних — це послідовності заданої довжини (sequence_size) з певною кількістю ознак (n_features).

Додається шар вирівнювання (Flatten), який перетворює багатовимірні дані в одномірний масив.

Додається шар Dropout із заданою ймовірністю виключення нейронів (dropout), що допомагає запобігти перенавчанню.

Додається полносвязний шар (Dense) з заданою кількістю нейронів (dense_layer_size) і функцією активації ReLU.

Додається вихідний шар з двома нейронами і функцією активації softmax, що підходить для задачі бінарної класифікації.

Модель компілюється з використанням оптимізатора Adam і функції втрат binary_crossentropy, а також метрики точності.

Опис шарів результуючої моделі

```
self.model.fit(x_train,
               y_train,
               validation_data=(x_validation, y_validation),
               epochs=self._config['max_epochs'],
               class_weight=class_weights,
               batch_size=batch_size,
               verbose=self._config['verbose_nn'],
               callbacks=[EarlyStopping(monitor='val_loss',
                                        patience=self._config['patience_earlystopping'],
                                        restore_best_weights=True,
                                        verbose=self._config['verbose_nn']),
                          ReduceLROnPlateau(monitor='val_loss',
                                            factor=self._config['factor_reduce_lr'],
                                            patience=self._config['patience_reduce_lr'],
                                            min_lr=self._config['learning_rate_min'],
                                            verbose=self._config['verbose_nn'])])
```

Модель навчається на даних x_train та y_train. Використовуються дані валідації (x_validation, y_validation) для оцінки моделі під час навчання. Вказуються кількість епох, ваги класів, розмір батчу та рівень подробиць виводу.

Додаються два зворотні виклики (callbacks):
EarlyStopping: Зупиняє навчання, якщо значення функції втрат на валідації не покращується протягом заданої кількості епох. Також відновлюються найкращі ваги моделі.
ReduceLROnPlateau: Зменшує швидкість навчання, якщо значення функції втрат на валідації не покращується протягом заданої кількості епох.

Таким чином, цей код створює, компілює та навчає модель нейронної мережі для задачі бінарної класифікації часових рядів, використовуючи декілька корисних зворотних викликів для покращення процесу навчання.

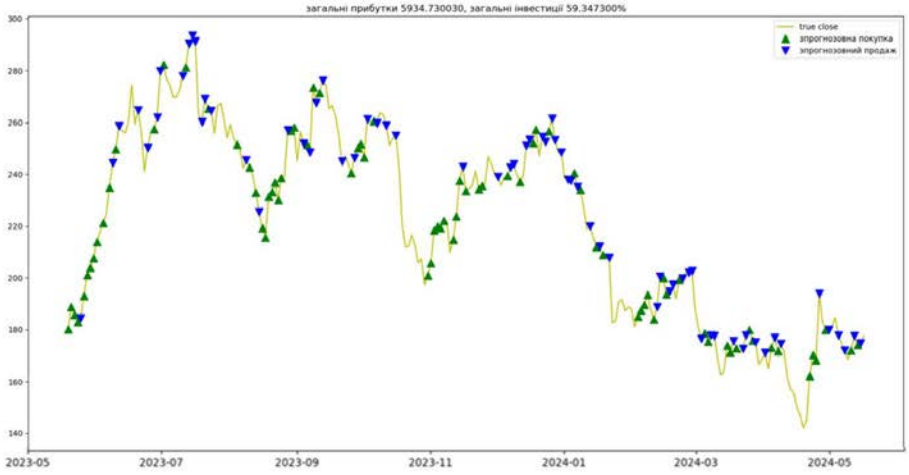
Результати експерименту

Результат застосування стратегії ABCD на річному наборі даних

Результати експерименту

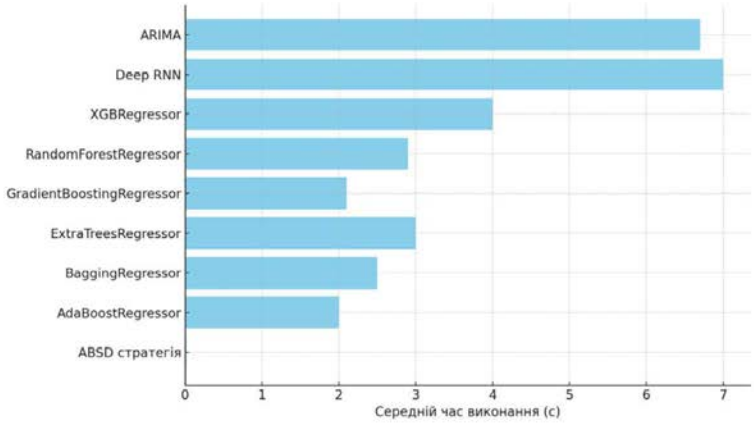
Результат застосування ABCD стратегії на даних за 3 місяці

Результати експерименту



Результат застосування стратегії з глибоким навчанням

Результати експерименту



Час виконання різних програмних рішень прогнозування

Аналіз отриманих результатів

При використанні моделей штучного інтелекту зі застарілими даними (більше одного дня) спостерігається значне відхилення від прогнозування цін (більше 5%).

Для поліпшення роботи програмних застосунків, доцільно запускати перенавчання моделей кожного дня після закінчення торгових сесій, оскільни в реальному часі перенавчання та запуск моделі займає більше часу, ніж відводиться на прийняття рішення стосовно торгівлі акціями.

Публікація результатів

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

СЕРТИФІКАТ

засвідчує, що

БУЗОВА КРІСТІНА ВАЛЕРІЇВНА

(Харківський національний університет радіоелектроніки)
взяла участь у роботі
секції **«Системний аналіз, математичне та комп'ютерне моделювання»**
конференції **«КОМП'ЮТЕРНИЙ ЗІР, СИСТЕМНИЙ АНАЛІЗ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ»**
в рамках **28-го** Міжнародного молодіжного форуму
«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ В XXI СТОЛІТТІ»

Голова Оргкомітету В.О. Дорошенко
«17» квітня 2024 р.

Підсумки

- Для ефективної торгівлі в реальному часі, використання програмного забезпечення, заснованого виключно на нейронних мережах, не завжди є доцільним. Це обумовлено тим, що такі системи потребують постійного перенавчання та коригування на основі нових даних, що займає значний час. Цей процес часто перевищує бажані часові рамки для здійснення оперативних купівлі або продажу активів. Натомість, програмні рішення, які базуються на стандартних торгових стратегіях, забезпечують вищу швидкість та передбачуваність результатів. Водночас, ці рішення можуть бути значно покращені завдяки використанню нейронних мереж для оптимізації значень коефіцієнтів, що дозволяє досягти ще більшої точності та ефективності у прогнозуванні.
- Таким чином, для успішного прогнозування майбутніх цін акцій необхідно поєднувати різні підходи та моделі, враховуючи їхні сильні сторони та обмеження. Використання сучасних технологій, таких як нейронні мережі, у комбінації з традиційними торговими стратегіями може значно підвищити швидкість прийняття рішень та точність прогнозів, що є ключовим фактором у сучасній динамічній фінансовій сфері.

ДОДАТОК Е

Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015

Експертний висновок результатів перевірки кваліфікаційної роботи

студент (посада) | програмної інженерії (кафедра) | ІПЗздм-22-1 (група)

Бузова Крістіна Валеріївна

(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні познаки	
	7.15 Додатки	

зауважень немає

Експерт _____ (підпис) Олена ОЛІЙНИК (прізвище, ініціали)

12.06.2024