

Министерство науки и высшего образования Российской Федерации  
Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа программной инженерии.

Работа допущена к защите

Директор ВШПИ

\_\_\_\_\_ П.Д.Дробинцев

«\_\_» \_\_\_\_\_ 2021 г.

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

работа бакалавра

### **ПРОГНОЗИРОВАНИЕ ТРЕНДОВ АКЦИЙ НА ОСНОВЕ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ТЕКСТОВ НОВОСТЕЙ**

по направлению подготовки (специальности) 09.03.04 –  
Программная инженерия

Направленность (профиль) 09.03.04\_01 – Технология  
разработки и сопровождения качественного программного  
продукта

Выполнил студент гр.  
3530904/70104

Д.О.Митусов

Руководитель  
доцент, к.т.н

О.Г.Малеев

Консультант  
по нормоконтролю

Е.Г.Локшина

Санкт-Петербург  
2021

Министерство науки и высшего образования Российской Федерации  
Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа программной инженерии.

УТВЕРЖДАЮ

Директор ВШПИ

\_\_\_\_\_ П.Д.Дробинцев

«\_\_» \_\_\_\_\_ 20\_\_ г.

## ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Митусову Дмитрию Олеговичу, 3530904/70104

1. Тема работы: Прогнозирование тренда акций на основе интеллектуального анализа текстов новостей

2. Срок сдачи студентом законченной работы:

3. Исходные данные по работе: документация Tensorflow, документация PyTorch

4. Содержание работы (перечень подлежащих разработке вопросов):

- 1.1 Сбор наборов данных
  - 1.1.1 Данные новостей
  - 1.1.2 Данные рынка
- 1.2 Обучение модели анализа новостей

Санкт-Петербург  
2021

- 1.3 Обучение моделей для торговли
- 1.4 Поточковая передача и хранение данных
  - 1.4.1 Данные новостей
  - 1.4.2 Данные рынка
  - 1.4.3 Формат хранения
- 2.1 Описание алгоритма работы с системой
- 2.2 Полученные результаты

5. Перечень графического материала (с указанием обязательных чертежей):

-

6. Консультанты по работе:

-

7. Дата выдачи задания

25.01.2021

Руководитель ВКР \_\_\_\_\_

О.Г.Малеев

Задание принял к исполнению

25.01.2021

Студент \_\_\_\_\_

Д.О.Митусов

## РЕФЕРАТ

На 35 с., 7 рисунков, 4 таблицы, 1 приложение

КЛЮЧЕВЫЕ СЛОВА: BERT, ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА, Q-ОБУЧЕНИЕ, ГЛУБОКОЕ ОБУЧЕНИЕ

Тема выпускной квалификационной работы: «Прогнозирование тренда акций на основе интеллектуального анализа текстов новостей».

Данная работа посвящена реализации алгоритмов машинного обучения для торговли на фондовой бирже.

Задачи, которые решались в ходе работы:

1. Реализация алгоритма классификации новостей для нахождения новостей, способных повлиять на цены акций компании.
2. Реализация алгоритма осуществления сделок на фондовой бирже, использующего в своей стратегии новостной фон.
3. Анализ результатов совместного использования реализованных алгоритмов.

Работа была проведена с использованием методов обработки естественного языка и алгоритмов глубокого обучения с подкреплением. Полученное решение позволяет делать прогноз тренда акций на небольшой промежуток времени для осуществления внутрисдневных сделок. На основании полученных результатов был сделан вывод о возможности применения реализованного решения для торговли на фондовой бирже.

## **ABSTRACT**

35 pages, 7 figures, 4 tables, 1 appendix

**KEYWORDS:** BERT, NATURAL LANGUAGE PROCESSING, Q-LEARNING, DEEP LEARNING

The topic of the final qualifying work: "Forecasting the trend of stocks based on the intellectual analysis of news texts."

This work is devoted to the implementation of machine learning algorithms for trading on the stock.

Tasks that were solved in the course of work:

1. Implementation of the news classification algorithm for finding news that can affect the prices of the company's shares.
2. Implementation of an algorithm for carrying out transactions on the stock exchange, using the news background in its strategy.
3. Analysis of the results of joint use of the implemented algorithms.

The work was carried out using natural language processing techniques and deep reinforcement learning algorithms. The resulting solution allows you to forecast the trend of stocks for a short period of time to carry out intraday transactions. Based on the results obtained, it was concluded that it is possible to use the implemented solution for trading on the stock exchange.

## СОДЕРЖАНИЕ

Введение.....	8
Глава 1. Описание реализованной системы .....	10
1.1. Сбор наборов данных.....	10
1.1.1. Данные новостей .....	10
1.1.2. Данные рынка.....	11
1.2. Обучение модели анализа новостей .....	12
1.3. Обучение моделей для торговли.....	18
1.4. Потоксовая передача и хранение данных .....	20
1.4.1. Данные новостей .....	20
1.4.2. Данные рынка.....	22
1.4.3. Формат хранения.....	22
Глава 2. Работа системы .....	24
2.1. Описание алгоритма работы с системой .....	24
2.2. Полученные результаты.....	27
Заключение .....	31
Список использованных источников .....	33
Приложение 1.....	35

## **ВВЕДЕНИЕ**

### **Актуальность и практическая значимость**

Предсказание движений фондового рынка – известная проблема. Одним из факторов, влияющим на цены акций, являются новости. Социальные сети являются одним из главных источников новостей, а также отражают общественное мнение о текущих событиях.

В работе рассматривается внутридневная торговля, которая связана с большими рисками. При внутридневной торговле потери могут составлять в 5-10 раз больше планируемых.[1] Но в то же время, при выборе правильной стратегии она может приносить значительную прибыль. Поэтому важно найти подход, минимизирующий риски. В данной работе рассматривается подход, при котором сначала ведется предварительный поиск компаний, акции которых в скором времени могут возрасти в цене.

### **Степень научной разработанности проблемы**

Работа охватывает две области применения нейронных сетей – для классификации оттенка текста (новости) и для биржевой торговли.

Для первого случая есть публикация в схожей области – Twitter mood predicts the stock market [2]. В статье исследовали, коррелируют ли коллективные состояния настроения публики (счастливое, спокойное, тревожное), полученные из сообщений пользователей Твиттера, со значением промышленного индекса Dow Jones. Для своих прогнозов они использовали нейро-нечёткие системы. Их результаты показывают,

что состояние общественного настроения в Твиттере сильно коррелирует с промышленным индексом Dow Jones.

Второй случай рассматривается в статье «Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting».[3] Для достижения цели используется агент Q-обучения, обученный несколько раз с одними и теми же данными. Экспериментальные результаты внутридневной торговли указывают на лучшую производительность, чем у традиционной стратегии «покупай и удерживай».

В этой работе совмещаются оба подхода для создания системы для биржевой торговли, требующей минимального контроля со стороны трейдера.

### **Цель работы**

Цель работы - реализация алгоритмов машинного обучения для торговли на фондовой бирже и их возможному применению.

### **Задачи работы**

1. Реализация алгоритма классификации новостей для нахождения новостей, способных повлиять на цены акций компании.
2. Реализация алгоритма осуществления сделок на фондовой бирже, использующего в своей стратегии новостной фон.
3. Нахождение и настройка источников для новостей и данных фондового рынка
4. Анализ результатов совместного использования реализованных алгоритмов.



# ГЛАВА 1. ОПИСАНИЕ РЕАЛИЗОВАННОЙ СИСТЕМЫ

## 1.1. Сбор наборов данных

### *1.1.1. Данные новостей*

Были собраны данные финансовых новостей для обучения модели.

Источник: Thomson Reuters - содержит информацию об опубликованных новостных статьях / оповещениях, такую как сведения о статьях, настроения и другие комментарии. Данные собраны за 2016 год, их общее количество – 952396.

Наиболее полезной информацией, которая будет использована для данной задачи классификации будет:

- `time(datetime64 [ns, UTC])` - метка времени в формате UTC, показывающая, когда данные стали доступны (вторая точность)
- `headline(object)` - заголовок новости
- `urgency(int8)` - типы новости (1: оповещение, 3: статья). Для решаемой задачи важно классифицировать именно оповещения
- `sentimentClass(int8)` - указывает преобладающий класс настроения для данной новости по отношению к активу. (-1: негативный, 0: нейтральный, 1: позитивный)

### 1.1.2. Данные рынка

Исторические данные рынка, которые были использованы – цена (в долларах США), объем, сопоставленные с соответствующими им датой и временем. Охватываемое время – с марта 2020 года по март 2021 года. Был выбран промежуток в пять минут, так как модель должна быстро реагировать на изменения цены.

Для сбора использовался tiingo API. В табл. 1 приведен пример данных для тикера \$TSLA.

Таблица 1. Пример собранных данных для тикера \$TSLA

date	close	volume
2020-03-27 12:10:00-04:00	500.665	479
2020-03-27 12:15:00-04:00	502.3	710
2020-03-27 12:20:00-04:00	502.36	103
2020-03-27 12:25:00-04:00	501.33	300
2020-03-27 12:30:00-04:00	501.735	484
2020-03-27 12:35:00-04:00	503.455	2755
2020-03-27 12:40:00-04:00	505.46	800
2020-03-27 12:45:00-04:00	504.97	1668

## 1.2. Обучение модели анализа новостей

Использованные библиотеки: tensorflow, transformers(huggingface), sklearn, pandas, gdown

Обучение модели производится на платформе Kaggle Kernels с GPU NVidia K80 (включение графического процессора приводит к ускорению в 12,5 раза во время обучения модели.[4])

Для импортирования набора данных для обучения и работы с ним используются библиотеки gdown + pandas.

«headline» (заголовок) будет служить в качестве исходных данных для модели, а «sentimentClass» (класс настроений) и «urgency» (срочность) - в качестве выходных данных.

Данные разделены на данные для обучения – 80% и данные для проверки – 20%.

Для классификации используется BERT - двунаправленная модель, основанная на архитектуре трансформера, которая заменяет последовательные рекуррентные нейронные сети гораздо более быстрым подходом, основанным на механизме внимания. Модель также предварительно обучена двум задачам обучения без учителя: предсказанию скрытых слов и предсказанию следующего предложения.[5] Это позволяет использовать предварительно обученную модель BERT, после ее точной настройки для выполнения конкретных задач, таких как классификация настроений, обнаружение намерений, ответы на вопросы и многое другое. Модели BERT обычно предварительно обучаются на большом объеме текста, а затем настраиваются для конкретных задач.

После предварительной обработки данных загружается объект конфигурации BERT, который управляет моделью и токенизатором. Затем токенизатор, который будет использоваться позже в скрипте, чтобы преобразовать вводимый текст в токены BERT, а затем дополнить и усесть их до максимально возможной длины. После этого загружается сама модель BERT.

Используемый вариант модели – BERT-base-uncased, так как все новости класса alert написаны прописными, а все новости класса article – строчными, и это не должно учитываться при обучении модели.

BERT-base-uncased использует словарь из 30 522 слов. Процесс токенизации включает в себя разбиение входного текста на список токенов, доступных в словаре. Для того, чтобы иметь дело со словами, не доступными в словаре, BERT использует технику, называемую BPE (кодирование пар байтов) основы WordPiece токенизацией. В этом подходе слово вне словарного запаса постепенно разбивается на подслова, а затем слово представляется группой подслов. При решении данной задачи максимальная длина токенов была взята, равной 68.

В библиотеке Transformers есть несколько различных моделей классификации BERT. Основная - это модель, которую просто называют «BertModel» (PyTorch) или «TFBertModel» (TensorFlow). Эта модель и будет использована.

Библиотека Transformers также поставляется с предварительно созданной моделью BERT для классификации последовательностей, называемой `TFBertForSequenceClassification`. В ней после загрузки чистой модели BERT, затем просто добавляют к ней слои `dropout` и `dense`. В случае классификации двух последовательностей просто добавляется два `dense` слоя вместо одного. На рис. 1 изображен общий вид используемой модели.

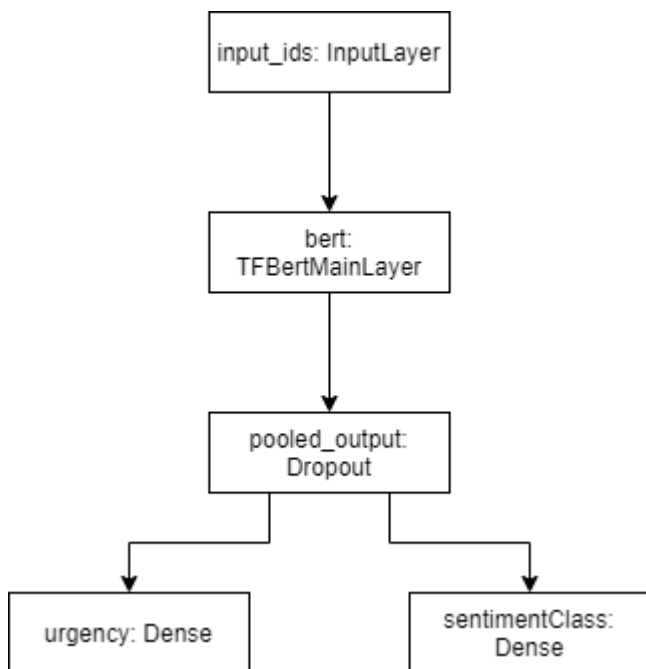


Рис. 1. Общая схема модели

На рис. 2 - более подробное описание модели.

Model: "BERT\_model19"

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 68)]	0	
bert (TFBertMainLayer)	TFBaseModelOutputWith	109482240	input_ids[0][0]
pooled_output (Dropout)	(None, 768)	0	bert[0][1]
sentimentClass (Dense)	(None, 3)	2307	pooled_output[0][0]
urgency (Dense)	(None, 3)	2307	pooled_output[0][0]

Total params: 109,486,854  
Trainable params: 109,486,854  
Non-trainable params: 0

Рис. 2. Подробное описание используемой модели

Затем все, что остается сделать, - это скомпилировать модель и проверить ее работу на тестовых данных.

Установка функции потерь и метрики:

```
loss = {'urgency': CategoricalCrossentropy(from_logits =
True), 'sentimentClass': CategoricalCrossentropy(from_logits
= True)}
```

```
metric = {'urgency': CategoricalAccuracy('accuracy'), 'senti-
mentClass': CategoricalAccuracy('accuracy')}
```

Основным изменением здесь является использование функции потерь категориальной кросс-энтропии вместо обычной, которая используется для мультиклассовой классификации. Функция потерь категориальной кросс-энтропии позволяет модели назначать меткам независимые вероятности.

Цикл обучения идентичен циклу, предоставленному в исходной реализации BERT.[6] Значения epoch и batch\_size были взяты 6 и 32 соответственно.

Для тонкой настройки используется тот же оптимизатор, с которым изначально обучался BERT: «Adaptive Moments»

(Адам). Скорость обучения была взята на уровне 5e-5, как рекомендовано в исходной статье.[5]

После того, как модель обучена, ее можно оценить на данных для проверки, чтобы увидеть, как она работает:

loss: 0.3628

sentimentClass\_loss: 0.3288

urgency\_loss: 0.0341

sentimentClass\_accuracy: 0.8031

urgency\_accuracy: 0.9874

Как оказалось, модель неплохо определяет срочность новости, но может ошибаться при анализе настроений. Но учитывая, что:

А) классов настроений – 3, в отличие от вариантов срочности новости;

Б) финансовые новости, как правило, не имеют ярко выраженных оттенков настроения;

В) в одной новости может идти речь сразу о нескольких компаниях, для каждой из которых оттенки настроения могут отличаться;

результаты являются удовлетворительными, и модель можно применять для анализа новостей из Твиттера.

Пример работы на наборе данных из Твиттера, (тикер: AAPL), приведен в табл. 2.

Таблица 2. Пример работы на наборе данных из Твиттера, (тикер: AAPL)

	date	time	urgency	sentimentClass	tweet
0	20.03.2021	18:56:54	2	2	Apple's stock, in the last two decades, has had many rough patches. It is very hard to hold through them.. But i always emphasize the same point, own it don't trade it, precisely because of how well it has done! \$AAPL
1	20.03.2021	17:21:59	2	0	BTW @TMobile is also now collecting and selling your data and you need to opt out of it. Not easy to figure out. Why do I have to opt out. It should be, I have to opt in. These devices need to be controlled by the user @tim_cook - enough of this spying. \$aapl \$tmus
2	20.03.2021	17:20:01	2	0	We need a setting to turn off all data collection from apps on the iPhone @tim_cook - just release it, apple. ItsB™s my phone. Its my info and I should be able to control it without being duped by apps. \$aapl #DeleteFacebook
4	20.03.2021	4:00:32	0	1	Mark Zuckerberg had a change of heart about the privacy changes expected with Apple's ( \$AAPL ) upcoming update to iOS. <a href="https://t.co/kR8VCziQ3a">https://t.co/kR8VCziQ3a</a>

В данном наборе данных было 1000 новостей за 2020-2021 годы. Для класса срочности получено количество оповещений – 539. Для классов настроения разбиение выглядит следующим образом:

- 561 - нейтральных
- 245 - положительных
- 195 – отрицательных



### 1.3. Обучение моделей для торговли

Так как в этой работе используется только одна стратегия – игра на повышение, при обучении достаточно учитывать лишь общий тренд – повышение цены через короткое время после покупки акций. Таким образом нет необходимости использовать данные большого количества компаний, достаточно лишь использовать исторические данные об акциях, которые обладали высокой волатильностью.

Была реализована модель для торговли акциями, обученная с использованием глубокого обучения с подкреплением, в частности глубокого Q-обучения.

Обучение с подкреплением – это семейство методов машинного обучения, которые позволяют создавать агентов, которые учатся у окружения, взаимодействуя с ним, по мере того, как они изучают оптимальную линию поведения методом проб и ошибок. Важная идея здесь заключается в том, что этот метод может быть применен к любой реальной задаче, которую можно условно описать как марковский процесс.

При Q-обучении на основе получаемого от среды вознаграждения агент формирует функцию полезности  $Q$ , что впоследствии дает ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой. Одно из преимуществ Q-обучения — то, что оно в состоянии сравнить ожидаемую полезность доступных действий, не формируя модели окружающей среды.

Была использована Q-сеть с двойной глубиной(DDQN), так как для стандартной Q-сети(DQN) есть несколько статистических проблем, которые решены в DDQN:

- 1) DQN аппроксимирует набор значений, которые сильно взаимосвязаны;
- 2) DQN склонна быть чрезмерно оптимистичной. Она будет переоценивать пребывание в одном состоянии, хотя это произошло только из-за статистической ошибки;[7]

В задаче торговли на фондовой бирже Q-обучение применяется следующим образом:

- 1) В определенный момент времени (эпизод) агент сохраняет текущее представление цены акций во временном окне
- 2) Агент выбирает и выполняет действие – покупка, продажа или удерживание.
- 3) Агент наблюдает за последующим состоянием, получает сигнал вознаграждения, измеряя разницу в инвестиционном портфеле
- 4) Агент регулирует свои параметры на основе вычисленного градиента потерь.

В начале работы алгоритма совершается покупка фиксированного количества акций или покупка акций на фиксированную сумму. В конце совершается продажа всех непроданных акций.

Было выбрано количество эпизодов – 100. После каждых 10 эпизодов полученная модель сохранялась. После этого работа моделей была проверена на различных тестовых данных. Результаты изображены на графике (рис. 3).

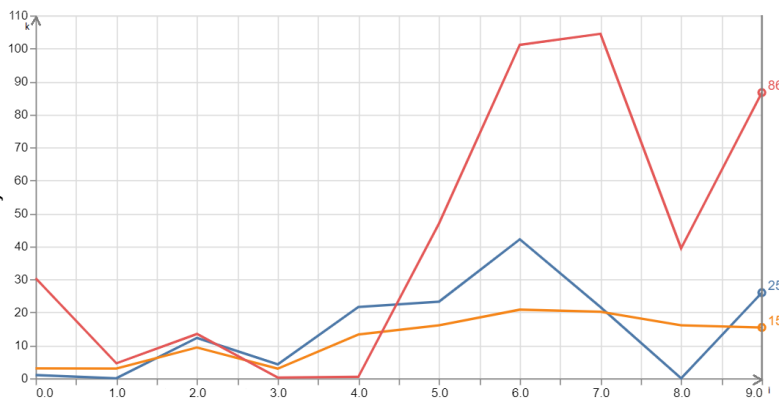


Рис. 3. Результаты проверки работы моделей на различных тестовых данных

Ось абсцисс – номер модели, ось ординат – изменения в инвестиционном портфеле.

По полученным результатам можно сделать вывод, что при стратегии игры на повышение при внутридневной торговле модель model\_50\_20, обучаемая в течение 70 эпизодов, наиболее эффективна.

## 1.4. Потокковая передача и хранение данных

### 1.4.1. Данные новостей

Возможность получения данных из Twitter в реальном времени поддерживается только официальным Twitter API, бесплатная версия которого имеет ограничения по количеству

запросов и по времени. Поэтому для данной задачи использовалась библиотека `twint`, имеющая инструмент парсинга Twitter, написанная на Python. `Twint` позволяет извлекать сообщения из профилей Twitter без использования Twitter API.

Для ускорения работы по сбору данных используется `multiprocessing`. При таком подходе средняя скорость составляет примерно 600 тикеров в минуту.

Варианты работы:

- Сообщения, содержащие “\$<имя\_тикера>” или имя компании от всех пользователей – наибольшее количество сообщений, большое количество спама и ненужной информации.
- Сообщения, содержащие “\$<имя\_тикера>” или имя компании от верифицированных пользователей – значительно ограничен круг пользователей, отсутствие спама, но не все посты являются новостями.
- Сообщения, содержащие “\$<имя\_тикера>” или имя компании и имеющие метку “новость” от всех пользователей – значительно ограничено количество постов, практически отсутствует спам и ненужная информация, но не все посты, являющиеся новостями, имеют метку “новость”, некоторая информация может быть пропущена.
- Сообщения, содержащие “\$<имя\_тикера>” или имя компании и имеющие метку “новость” от верифицированных пользователей – каждое сообщение имеет ценность, но информации очень мало.

Из-за избытка ненужной информации в варианте 1, недостатка информации в варианте 4, было решено использовать вариант 3.

Список тикеров Nasdaq проходится примерно за 7 минут, если также включать в поиск полные названия компаний. Это минимальная периодичность запуска скрипта. Но при такой периодичности при вариантах поиска 2, 3, 4 среднее количество найденных новостей меньше одной. Поэтому более оптимальным будет поиск раз в 10-15 минут. За это время новость будет оставаться релевантной.[8]

#### *1.4.2. Данные рынка*

Использованные библиотеки: alpaca-trading-api, websocket, sqlite3.

Для получения информации о текущей цене акций используется Alpaca Data API. На вход поступает тикер, полученный при работе с новостями. При каждом обновлении цены поступает сообщение, и вызывается функция `on_message` которая записывает полученное сообщение с тикером, ценой и объемом в базу данных.

#### *1.4.3. Формат хранения*

Полученные данные хранятся в базе данных sqlite. На рис. 4 изображена визуализация базы.

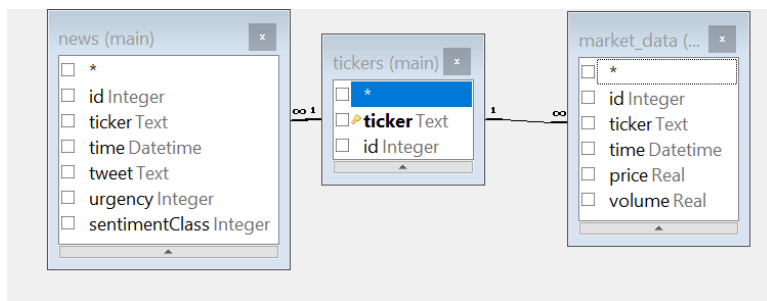


Рисунок 4. Формат хранения в базе данных

Список тикеров - таблица tickers:

- «Ticker» - тикер

Новости и их классы - таблица news

- «Ticker» - тикер
- «Time» - текущие дата и время
- «Tweet» - новость
- «Urgency» - новость
- «SentimentClass» - объем на данный момент

Данные рынка - таблица market\_data:

- «Ticker» - тикер
- «Time» - текущие дата и время
- «Price» - Цена на данный момент
- «Volume» - объем на данный момент

После завершения работы системы, данные об этом тикере удаляются из базы данных.

## **ГЛАВА 2. РАБОТА СИСТЕМЫ**

### **2.1. Описание алгоритма работы с системой**

Готовое решение представляет собой пакет прикладных программ для анализа торговых стратегий. После завершения задач по обучению, можно начать тестировать работу системы. Общая схема системы приведена на рис. 5.

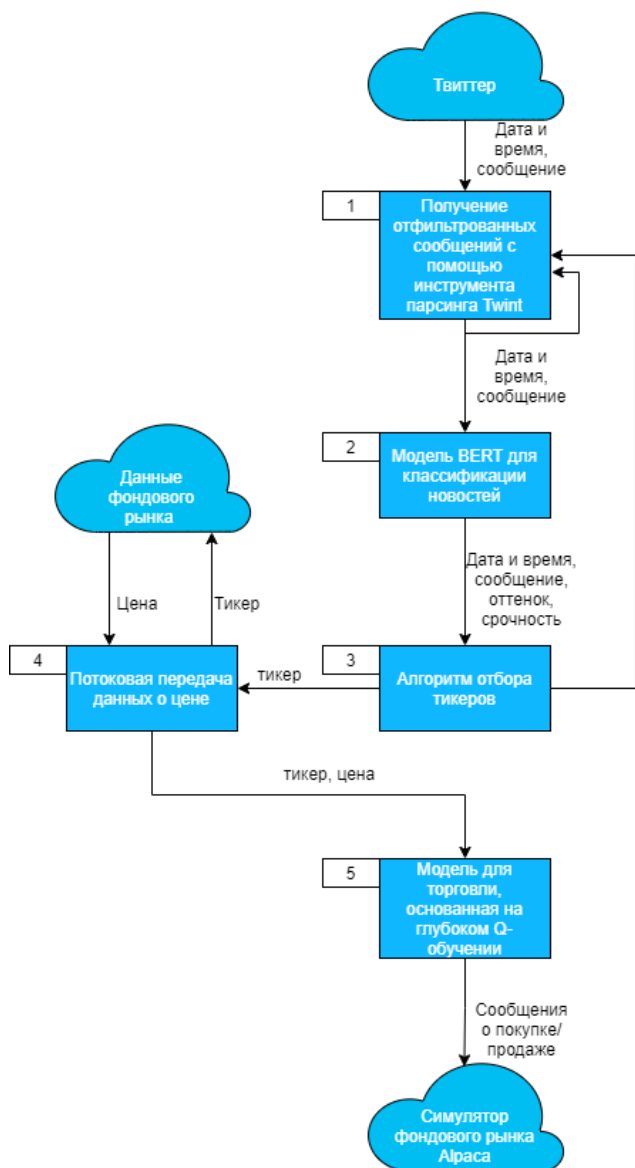


Рис. 5. Общая схема работы реализованной системы



Весь проект разделен на несколько скриптов, которые запускаются помощью bat-файла. Более подробное описание алгоритма работы:

1 - Запускается поиск сообщений с заданным фильтром за последние 15 минут, занося найденные результаты в базу данных, в случае если сообщения по заданному фильтру были найдены. Если сообщений по заданному фильтру не было найдено, через 15 минут поиск запускается повторно.

2 - После завершения поиска сообщений начинает работу скрипт с работающей моделью BERT, предварительно обученной на собранном наборе новостей. Получаемые результаты – класс оттенка настроения и класс срочности новости. Результаты заносятся в базу данных.

3- Производится поиск по алгоритму отбора тикеров. Выбирается тикер, для которого новости о соответствующей ему компании имеют положительный оттенок и класс срочности – оповещение. Количество новостей с положительным оттенком должно быть больше, чем с отрицательным (нейтральные новости не учитываются). В случае нахождения более одного тикера, подходящего под заданные требования, выбирается тикер с наибольшим количеством положительных и нейтральных новостей. В случае, если тикер не был найден, происходит ожидание 15 минут, и поиск новостей запускается повторно.

4 - Когда подходящий тикер найден, начинается потоковая передача данных о цене акций с помощью Alpaca Data API.

Полученная информация начинает заносится в базу данных.

5 – Из базы данных информация поступает на вход модели для торговли на фондовой бирже, основанной на глубоком Q-обучении. Выходные данные модели – действия, которые нужно совершить в данное время, для получения наибольшей прибыли. Возможные действия: покупка, продажа, удерживание. Завершение работы происходит в 0:00 по североамериканскому восточному времени. Все вхождения в базе данных об этом тикере удаляются.

Регулируемые параметры:

- 1) Время начала поиска новостей, то есть время запуска скрипта
- 2) Фильтр для сообщений (варианты – активен фильтр новостей, активен фильтр верифицированных пользователей, активны оба фильтра)
- 3) Выбор модели для торговли (модель, подходящая для компаний с более высокой волатильностью акций и модель, подходящая для компаний с низкой волатильностью акций)

## **2.2. Полученные результаты**

Результаты получены на основе исторических данных. Для этого сначала были собраны новости (табл. 3).

Таблица 3. Собранные новости, компания Nvidia

	date	tweet	urgency	sentimentClass
0	27.05.2021	NVIDIA Virtual GPU and @VMware empower enterprises with graphics-rich, secure, and cost-effective VDIs. Read this @Forbes article to learn why #vGPU solutions are essential for "anywhere workspaces" across industries. #VDI #VWS #vPC <a href="https://t.co/CzJBQuY5aY">https://t.co/CzJBQuY5aY</a> <a href="https://t.co/HvcnKdy8qC">https://t.co/HvcnKdy8qC</a>	2	2
1	27.05.2021	Nvidia made \$5.66 billion in revenue last quarter, despite RTX 30-Series shortages. <a href="https://t.co/q6XO6c7yIp">https://t.co/q6XO6c7yIp</a> <a href="https://t.co/t2p2l6NOx6">https://t.co/t2p2l6NOx6</a>	0	1
2	27.05.2021	Nvidia anticipates RTX 30-Series graphics card shortages will continue until the "second half of the year." <a href="https://t.co/Mbqua7Ofyx">https://t.co/Mbqua7Ofyx</a> <a href="https://t.co/MiqAegaZLt">https://t.co/MiqAegaZLt</a>	0	0
3	27.05.2021	Nvidia teases GeForce RTX 3080 Ti announcement for May 31st <a href="https://t.co/lRbN4pG4M0">https://t.co/lRbN4pG4M0</a> <a href="https://t.co/wWqmYhOq4q">https://t.co/wWqmYhOq4q</a>	0	1
4	27.05.2021	Nvidia gaming revenue spikes by 106%, data center by 79% <a href="https://t.co/xjocHanDDp">https://t.co/xjocHanDDp</a> by @DanielHowley <a href="https://t.co/SqBPkwBBia">https://t.co/SqBPkwBBia</a>	0	2
5	27.05.2021	Nvidia forecast beats expectations but crypto mining's role remains unclear <a href="https://t.co/BNrowthXxe">https://t.co/BNrowthXxe</a> <a href="https://t.co/QJqU9kdTG8">https://t.co/QJqU9kdTG8</a>	0	1

Модель не всегда корректно определяет оттенок новости, поэтому для нее требуется дальнейшая доработка.

В случае, если новостей с положительными оттенками больше, чем с отрицательными, тестировалась работа модели для биржевой торговли. В рассматриваемом случае количество новостей с положительным оттенком больше, поэтому была проведена проверка работы модели, и на ее основе был построен график (рис. 6).

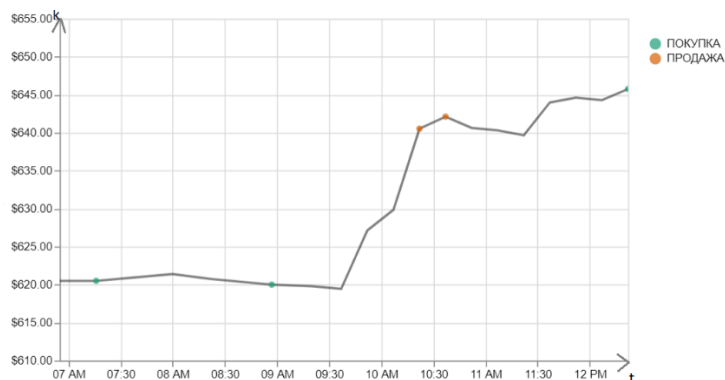


Рис. 6. Результаты работы системы для компании Nvidia

Ось абсцисс – это цена за акцию в дол. США, ось ординат – время.

Рассматриваемая компания – Nvidia, дата – 28.05.2021.

Был взят промежуток в 15 минут между возможностями совершения сделки. Производится покупка и продажа только одной акции за раз. В результате прибыль составила \$42,15 дол. США.

Второй рассматриваемый случай является примером, когда новости имеют более длительную актуальность. Акции компании дольше остаются в трендах. Для этого примера новости собирались и классифицировались в течение двух дней. Результат представлен в табл. 4.

Таблица 4. Собранные новости, компания DocuSign

	date	tweet	urgency	sentimentClass
0	4.6.21	\$DOCU is 'in a sweet spot to continue to receive sig	2	2
1	4.6.21	Premarket Movers \$CLB +9.6% \$CLNE +7.7% \$DOCU	0	1
2	4.6.21	More investors sign up for DocuSign shares, while	0	2
3	3.6.21	\$DOCU Q1 FY22 International revenue increased 8	0	2
4	3.6.21	\$DOCU reported record dollar net retention of 125	0	0
5	3.6.21	Wowza DocuSign with another heater of a quarter	2	2
6	3.6.21	The most accurate \$DOCU estimate is from @sPAC	2	1
7	3.6.21	\$DOCU beats the Estimize EPS Consensus by 8c and	0	2
8	3.6.21	\$DOCU reports FQ1 earnings of 44c EPS and \$469.1	0	1
9	3.6.21	\$DOCU reports first quarter results for FY22 with 5	0	2
10	3.6.21	\$DOCU * Q1 Adj. EPS \$0.44 Beats \$0.28 Estimate	0	1
11	3.6.21	\$DOCU Q1 Earnings: EPS: \$0.44 vs. \$0.27 Expected	0	1
12	3.6.21	Tune into a #live webcast of \$DOCU Q1 FY22 earni	2	0
13	3.6.21	\$DOCU reports after the close, Estimize Consensus	0	0

Сделки совершались также в течение двух дней. График для второго рассматриваемого случая изображен на рис.

7.

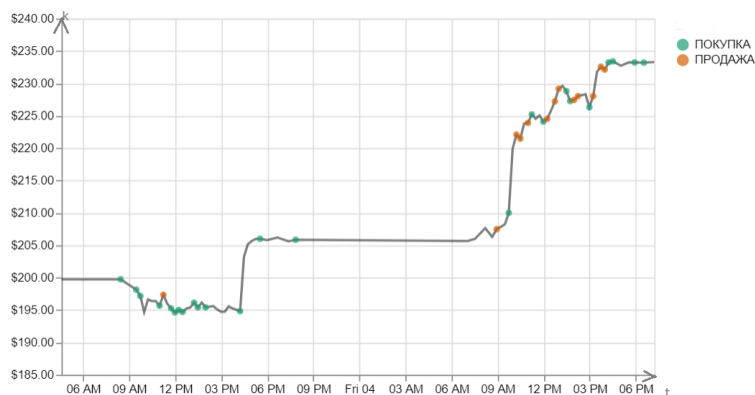


Рис. 7. Результаты двухдневной работы системы для компании DocuSign

Рассматриваемая компания – DocuSign, Inc., даты – 03.06.2021 - 04.06.2021.

Был взят промежуток в 15 минут между возможностями совершения сделки. Производится покупка и продажа только одной акции за раз. В результате прибыль составила \$343,28 дол. США.

В случае стандартной работы решения, при которой сделки совершались в пределах одного дня, прибыль составила \$18,45 дол. США (Приложение 1).

Таким образом, при дальнейшей возможной модификации созданного решения, можно будет добиться значительно большего увеличения капитала.

В результате был создан пакет прикладных программ для анализа торговых стратегий на основе исторических данных.

### **Заключение**

В ходе выполнения работы были решены следующие задачи:

1. Реализован алгоритм классификации новостей для нахождения новостей, способных повлиять на цены акций компании.
2. Реализован алгоритм осуществления сделок на фондовой бирже, который использует в своей стратегии новостной фон.
3. Найдены источники для новостей и данных фондового рынка.
4. Проведен анализ результатов совместного использования реализованных алгоритмов.

5. Создан пакет прикладных программ для анализа торговых стратегий на основе исторических данных.

На основе полученных результатов можно сделать вывод о полезности учета новостей при формировании торговых стратегий. При этом в дальнейшем необходимо совершенствование алгоритма отбора и классификации новостей, так как в некоторых случаях его работа может быть недостаточно точной при использовании в торговых стратегиях, в которых не требуется контроль со стороны трейдера.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Adam Grimes. The Art and Science of Technical Analysis / Adam Grimes. Hoboken: John Wiley & Sons, 2012. - 463 с.
2. Bollen, J. Twitter mood predicts the stock market / Bollen J., Mao H., Zeng X. // Journal of Computational Science. – 2011. - №2(1). - с. 1-8
3. Salvatore Carta. Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting / Salvatore Carta, Anselmo Ferreira, Alessandro Sebastian Podda // Expert Systems with Applications. – 2021. - №164. – с. 1-16
4. Running kaggle kernels with a gpu. URL: <https://www.kaggle.com/dansbecker/running-kaggle-kernels-with-a-gpu>
5. Jacob Devlin. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova // 2019. - с. 1-16
6. Репозиторий BERT. URL: <https://github.com/google-research/bert>
7. Hado van Hasselt. Deep Reinforcement Learning with Double Q-learning / Hado van Hasselt, Arthur Guez, David Silver // Google DeepMind. – 2015. – с. 1-13
8. Emanuele Tetia. The relationship between twitter and stock prices. Evidence from the US technology industry / Emanuele Tetia, Maurizio Dallochio, Alberto Aniasi //



Technological Forecasting & Social Change. - 2019. - №149

9. Hikmat Ullah Khan. Twitter trends: A ranking algorithm analysis on real time data / Hikmat Ullah Khan, Shumaila Nasir, Kishwar Nasim // Expert Systems with Applications. – 2021. - №164.
10. Документация Tensorflow API. URL: [https://www.tensorflow.org/api\\_docs](https://www.tensorflow.org/api_docs)
11. Документация Twitter API. URL: <https://developer.twitter.com/en/docs>
12. Twintproject. URL: <https://github.com/twintproject/twint>

## Приложение 1

