

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование)

02.03.03 Математическое обеспечение и администрирование
информационных систем

(код и наименование направления подготовки, специальности)

Мобильные и сетевые технологии

(направленность (профиль)/ специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка алгоритма для прогнозирования изменения цен акций на
фондовом рынке с использованием нейросетевых технологий»

Студент

К.А. Кутняшенко

(И.О. Фамилия)

(личная подпись)

Руководитель

к.тех.н., В.С. Климов

(ученая степень, звание, И.О. Фамилия)

Консультант

(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

АННОТАЦИЯ

Тема бакалаврской работы: «Разработка алгоритма для прогнозирования изменения цен акций на фондовом рынке с использованием нейросетевых технологий».

Бакалаврская работа посвящена разработке подходов по использованию нейросетевых технологий для анализа мирового рынка акций.

В ходе выполнения исследований по бакалаврской работе на языке Python был разработан веб-сервис, предоставляющий информацию по прогнозам полученных с помощью нейронной сети об изменении стоимости акций международных компаний в течение квартала. Для работы сервиса производится периодичное переобучение нейронной сети, при этом на ее вход подается информация об экономических показателях компаний за текущий квартал и торгующихся через Санкт-Петербургскую биржу.

Бакалаврской работа состоит из введения, трёх глав, заключения и списка литературы. Во введении описывается актуальность проводимого исследования, дается краткая характеристика проделанной работы.

В первой главе анализируется проблема прогнозирования стоимости акций на фондовом рынке.

Во второй главе описывается разработка алгоритма по прогнозированию стоимости акций с использованием нейросетевых технологий.

В третьей главе описывается программная реализация веб-сервиса с прогнозами изменения стоимости акций различных компаний.

В заключении представлены выводы по проделанной работе.

В работе присутствуют , 20 рисунков, 13 формул. Список литературы состоит из 20 литературных источников. Общий объем выпускной квалификационной работы составляет 44 страницы.

ABSTRACT

The topic of the present graduation work is *Development of an algorithm for predicting changes in stock prices on the stock market using neural network technologies*.

The research is devoted to the development of approaches to the use of neural network technologies for the analysis of the global stock market.

The course of conducting research on bachelor's work in Python, a web service was developed that provides information on forecasts obtained using a neural network about changes in the value of shares of international companies during the quarter. For the service to work, the neural network is periodically retrained, and its input is provided with information about the economic indicators of companies for the current quarter and traded through the St. Petersburg Stock Exchange.

The bachelor's thesis consists of an introduction, three chapters, a conclusion, and a list of references. The introduction describes the relevance of the research, gives a brief description of the work done.

The first chapter analyzes the problem of predicting the value of shares in the stock market.

The second chapter describes the development of an algorithm for predicting the value of shares using neural network technologies.

The third chapter describes the software implementation of a web service with forecasts of changes in the value of shares of various companies.

As a result of this work, the structure of the stock price forecasting service using a neural network was developed.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 АНАЛИЗ ПРОГНОЗНЫХ МОДЕЛЕЙ НА ФОНДОВОМ РЫНКЕ	7
1.1 Существующих подходов к прогнозированию стоимости акций	7
1.2 Выводы по главе.....	13
2 РАЗРАБОТКА НЕЙРОСЕТЕВОГО АЛГОРИТМА ДЛЯ ПРОГНОЗИРОВАНИЯ СТОИМОСТИ АКЦИЙ	14
2.1 Прогнозирование стоимости акций при фундаментальном анализе как задача регрессии.....	14
2.2 Обучающая выборка и структура нейронной сети.....	19
2.5 Выводы по главе.....	24
3 РАЗРАБОТКА СЕРВИСА НЕЙРОСЕТЕВОГО ПРОГНОЗИРОВАНИЯ	25
3.1 Архитектура сервиса.....	25
3.2 Программные решения, применяемые при реализации сервиса	27
3.3 Обсуждение результатов прогнозирования	36
3.4 Выводы по главе.....	38
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	41

ВВЕДЕНИЕ

Системы искусственного интеллекта все активней применяются в различных сферах науки и техники для автоматизации различных процессов и снижения доли участия человека при принятии решений. Так, например, в бизнесе искусственный интеллект используется для определения уровня лояльности клиента, повышения уровня эффективности таргетированной рекламы, создание автономных ассистентов, сортировки обращений клиентов по содержанию вопроса.

Одновременно с этим граждане Российской Федерации стремятся повысить свой уровень дохода за счет инвестирования в фондовый рынок. Так, на XI Биржевом форуме Московской биржи представитель Центрального Банка заявил, что в течение нескольких лет ожидается увеличение количества граждан с брокерскими счетами до 74 миллионов человек. При инвестировании средств на фондовом рынке актуальной проблемой является отбор наиболее выгодных для покупки акций. Стандартной стратегией получения дохода за счет акций является покупка акций с высоким потенциалом роста цены для последующей их продажи через несколько месяцев.

В этом случае, оптимизация дохода, получаемого с фондового рынка, достигается путем использования различных средств для прогнозирования изменения цен акций.

Технологии искусственного интеллекта позволяют проводить построение прогнозных моделей.

Существуют подходы по применению алгоритмов машинного обучения для прогнозирования изменения цен на акции. Они основаны на анализе исторических финансовых данных компаний и обучении на их основе прогнозных моделей. Недостатком таких подходов является снижение точности предсказаний выдаваемых прогнозной моделью с течением времени. Это связано с тем, что параметры модели неизменны и ее прогнозы

в какой-то момент времени перестают соответствовать динамически изменяющейся обстановке на финансовых рынках.

Данная работа направлена на преодоление этого недостатка.

Таким образом, цель работы – разработка адаптивного алгоритма для прогнозирования изменения цен акций на фондовом рынке с использованием нейросетевых технологий.

Для достижения поставленной цели в работе решаются следующие задачи:

1. Анализ существующих подходов к прогнозированию цен акций на фондовом рынке.
2. Разработка адаптивного алгоритма для прогнозирования изменения цен акций на фондовом рынке с использованием нейросетевых технологий.
3. Разработка программной реализации предложенного алгоритма прогнозирования цен на акции.

1 АНАЛИЗ ПРОГНОЗНЫХ МОДЕЛЕЙ НА ФОНДОВОМ РЫНКЕ

1.1 Существующих подходов к прогнозированию стоимости акций

По данным ТАСС от 7 апреля 2021 года заместитель председателя Банка России Владимир Чистюхин в рамках XI Биржевого форума Московской биржи заявил, что через несколько лет прогнозируется увеличение брокерских счетов в России до 74 миллионов. Базовой возможностью брокерского счета, помимо обмена валюты, является покупка ценных бумаг. Инвестирование части заработка в ценные бумаги потенциально позволяет владельцу брокерского счета получать дополнительный доход за счет увеличения стоимости ценных бумаг. Суммарный доход владельца брокерского счета зависит, в том числе, от состава приобретенных акций. Это связано с тем, что скорость роста цены акции разных эмитентов различна и динамически меняется в течение времени.

С учетом этого актуальной задачей владельца брокерского счета является балансировка портфеля акций, которая заключается в продаже выросших в цене акций и покупке акций высоким потенциалом роста.

Для оценки потенциала роста стоимости акций применяются различные методы анализа, которые условно можно разделить на две группы:

- фундаментальный анализ финансовых показателей компании;
- технический анализ графика стоимости акции в течение времени.

Прогнозирование будущей стоимости акции на основе фундаментального анализа предполагает, что цена акции связана с набором финансовых показателей рассматриваемой компании. Финансовые показатели компаний не являются коммерческой тайной и публикуются в открытом доступе каждый квартал. Примерами финансовых показателей

являются чистая прибыль за квартал, доход компании на акцию, уровень долга компании и т.д.

Прогнозирование будущей стоимости акции на основе технического анализа предполагает, что стоимость акции зависит от глобального баланса заявок на покупку и продажу в биржевом стакане. Так как наряду с физическими лицами на рынке ценных бумаг участвуют крупные компании и правительства государств, то предполагается, что колебания цен акции происходит около тех уровней, где размещены заявки крупных участников рынка. Прогнозирование с использованием технического анализа основано на определении (на графике цены акций) уровней поддержки, сопротивления, каналов и других закономерностей.

Существуют исследования, направленные на попытку использования искусственного интеллекта для прогнозирования цены акций как с использованием фундаментального анализа, так и с использованием технического анализа. Однако в данной работе рассматривается прогнозирования стоимости акций только на основе фундаментального анализа.

Первый существующих подход по созданию интеллектуальных систем прогнозирования стоимости акций базируется на следующих предположениях:

- финансовые показатели различных компаний можно использовать для сравнения эффективности их функционирования друг с другом;
- рост стоимости акций происходит у тех компаний, которые функционируют эффективней остальных;
- на основе огромного массива исторических данных можно построить общую регрессионную модель связывающую стоимость акции компании с ее финансовыми показателями.

Данный подход можно визуализировать в виде рисунка 1.1.

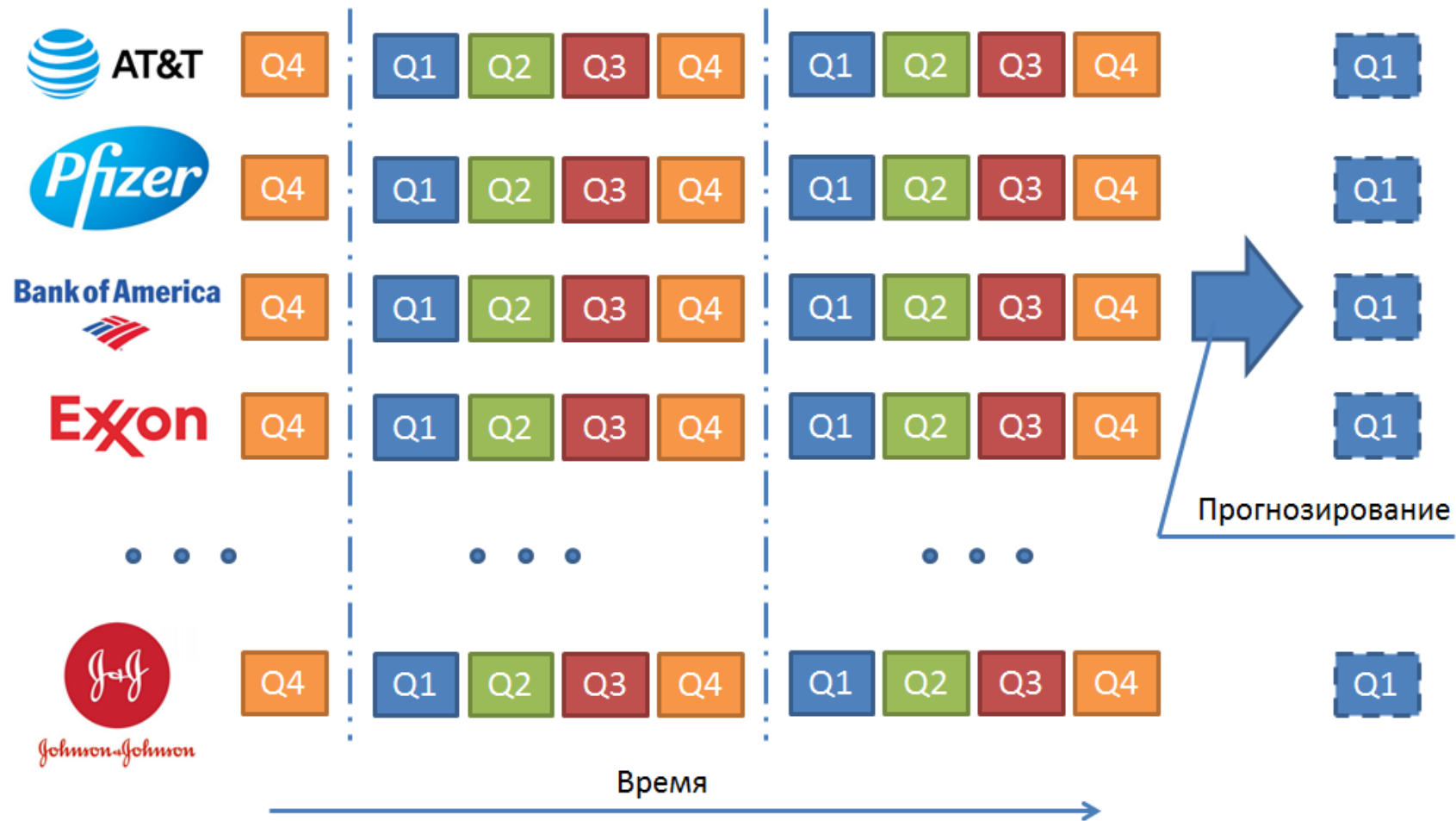


Рисунок 1.1 – Схема фундаментального анализа, при котором прогнозная модель обучается по всем доступным историческим данным выбранных эмитентов

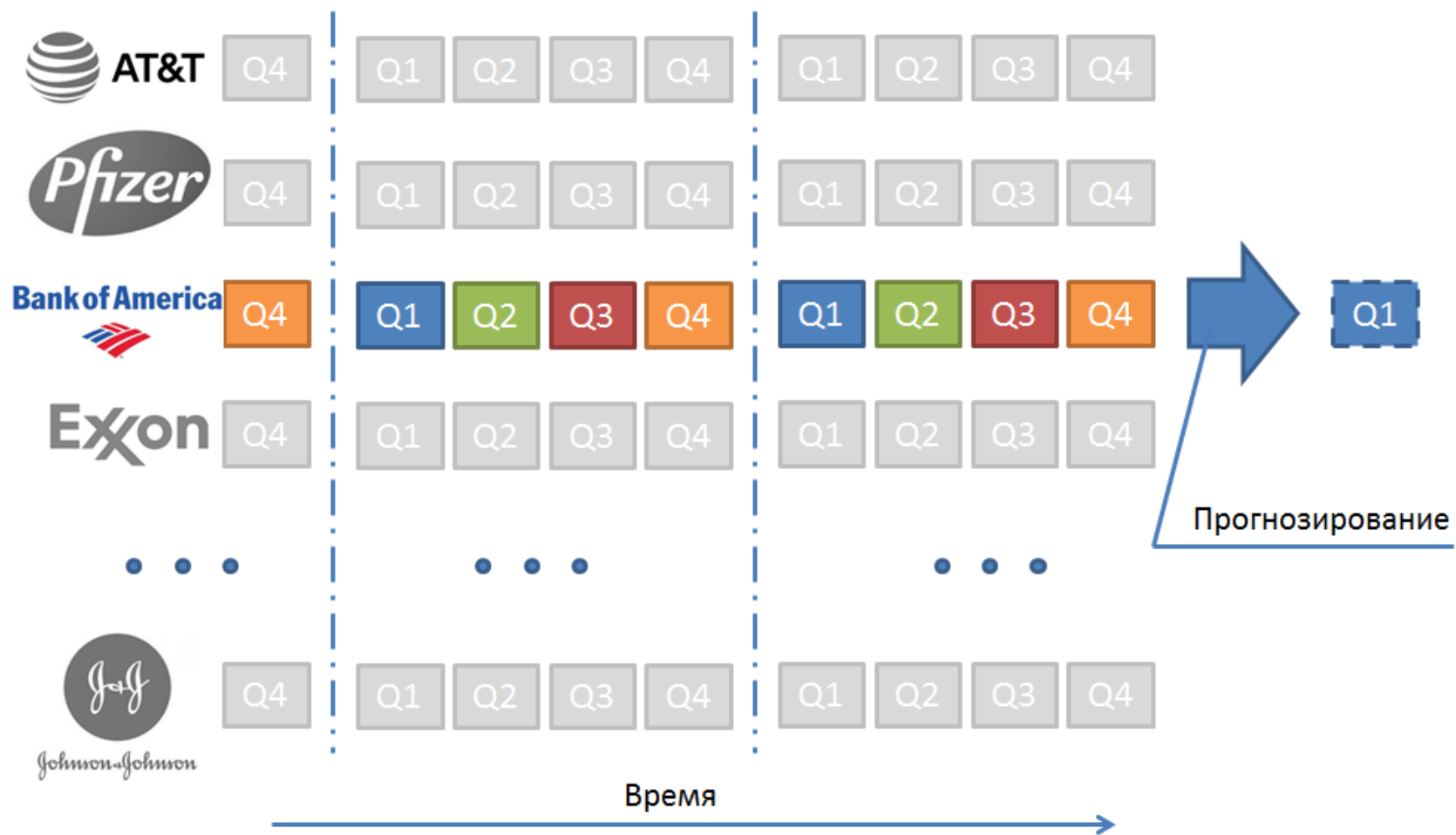


Рисунок 1.2 – Схема фундаментального анализа, при котором прогнозная модель обучается на исторических данных одного выбранного эмитента

На рисунке 1.1 (и далее на рисунках 1.2, 1,3) в виде прямоугольников Q1, Q2, Q3, Q4 показаны наборы финансовых показателей компаний за первый, второй, третий и четвертый квартал года.

На основе такого подхода с использованием алгоритмов машинного обучения предполагается получение общей модели для прогнозирования цены акций любой компаний.

Достоинством данного подхода является большое количество исторических данных, позволяющих использовать алгоритмы машинного обучения, чувствительные к размеру обучающей выборки, например глубокие нейронные сети.

Недостатком такого подхода является использование данных, потерявших актуальность при обучении прогнозной модели. Дело в том, что со временем исторические финансовые данные теряют актуальность применительно к текущей обстановке на рынке. Например, раньше считалось, что наличие высокой долговой нагрузки у компании является негативным фактором, а в настоящее время долговая нагрузка обосновывается расширением бизнеса и использованием заемных средства для его цифровизации (модернизации).

Другой подход использования фундаментального анализа при прогнозировании цен акций основан на составлении более частной модели, направленной или на определенный сектор или на набор компаний со схожей моделью бизнеса. Визуально такой подход можно представить в виде следующей схемы (рисунок 1.2). Достоинством такого подхода основанного на персонализации является получение более точных прогнозов по ценам акции, чем при использовании предыдущего подхода.

Недостатком является быстрое устаревание прогнозной модели, т.е. с течением времени прогнозная модель дает все менее и менее точные прогнозы. Это связано с изменениями внешнего экономического мира которые не учитываются жестко заданной прогнозной моделью с фиксированными внутренними параметрами.

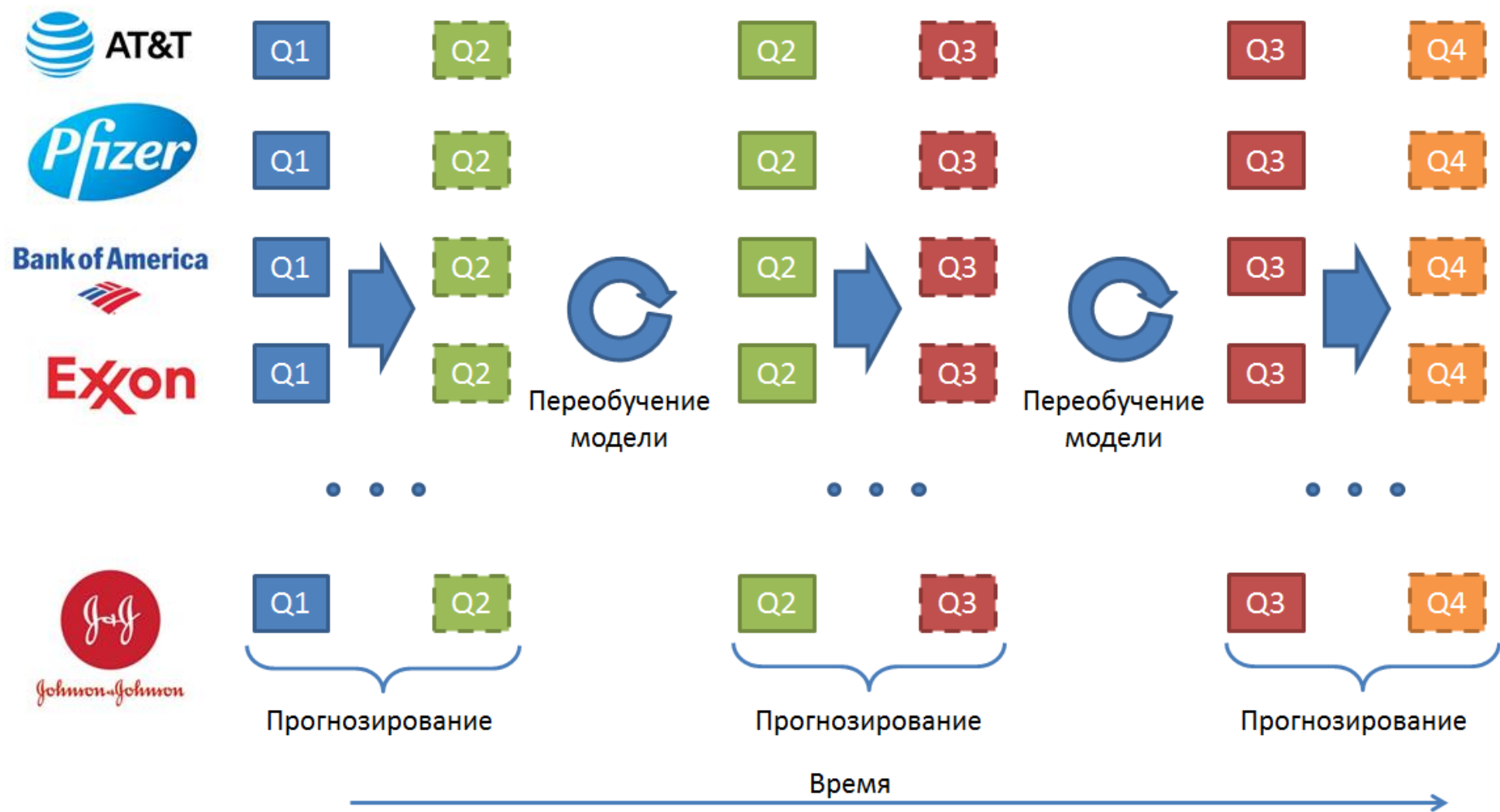


Рисунок 1.3 – Схема фундаментального анализа, при котором прогнозная модель переобучается ежеквартально (на квартальных отчетах эмитентов)

Для преодоления этого недостатка (отсутствие адаптируемости прогнозной модели к изменениям) предложена следующая схема фундаментального анализа (рисунок 1.3). При построении прогнозной модели используются финансовые данные всех компаний за один предыдущий квартал. Полученная модель используется для прогнозирования изменения цен акций в следующем квартале. Для обеспечения адаптируемости прогнозная модель строится заново каждый квартал с использованием новых квартальных данных. В качестве прогнозной модели предложено использовать нейронную сеть.

1.2 Выводы по главе

Проведен анализ подходов к построению прогнозных моделей на основе фундаментального анализа финансовых показателей компаний, представленных на фондовом рынке. Выявлен недостаток существующих подходов связанный с отсутствием адаптируемости прогнозных моделей к внешним экономическим изменениям.

Для преодоления данного недостатка предложена схема построения прогнозной модели, основанной на использовании финансовых показателей за предыдущий квартал для прогнозирования цен акций в следующем квартале. При этом адаптируемость прогнозной модели обеспечивается наличием процедуры ее ежеквартальном переобучении исключительно на актуальных квартальных данных. В качестве прогнозной модели предложено использовать нейронную сеть

2 РАЗРАБОТКА НЕЙРОСЕТЕВОГО АЛГОРИТМА ДЛЯ ПРОГНОЗИРОВАНИЯ СТОИМОСТИ АКЦИЙ

2.1 Прогнозирование стоимости акций при фундаментальном анализе как задача регрессии

Для прогнозирования стоимости акций необходимо построить регрессионную модель, связывающую текущие финансовые показатели компании и изменение цены компании в будущем.

Формально задача построения регрессионной модели можно описать следующим образом. Существует множество входных переменных X – описаний объектов, а также множество Y выходных переменных (значения которых зависят от входных переменных).

В задаче регрессии существует семейство функций, которое является отображением вида:

$$f : W \times X \rightarrow Y, \quad (2.1)$$

где $w \in W$ – пространство настраиваемых параметров, $x \in X$ – пространство входных переменных, $y \in Y$ – пространство выходных переменных, f – семейство функций вида $f(w, x)$.

Так как регрессионный анализ предполагает поиск зависимости математического ожидания случайной величины от свободных переменных:

$$E(y | x) = f(x), \quad (2.2)$$

то в её состав входит аддитивная случайная величина ε . Таким образом:

$$y = f(w, x) + \varepsilon. \quad (2.3)$$

Регрессионная модель является настроенной, когда зафиксированы её параметры, то есть модель задаёт отображение для фиксированных значений W :

$$f : X \rightarrow Y \quad (2.4)$$

Исходными данными для построения регрессионной модели является обучающая выборка D^m вида:

$$D^m = \{(x_1, y_1), \dots, (x_m, y_m)\}, \quad (2.5)$$

где m – размер обучающей выборки, x_i – вектор значений входных переменных для i -го объекта из обучающей выборки, y_i – значение выходного параметра для i -го объекта из обучающей выборки. При этом вектор x_i описывается набором значений числовых атрибутов P , а y_i – описывается единственным числовым значением:

$$x_i = (P_1, P_2, \dots, P_n), \quad (2.6)$$

где, n – количество атрибутов, описывающих объект из обучающей выборки.

Одним из возможных способов построения регрессионных моделей является использование нейронных сетей прямого распространения. Такая нейронная сеть состоит из однотипных вычислительных элементов – нейронов.

Математическая модель нейрона была предложена американскими учеными У. Маккалоком и У. Питтсом. Модель нейрона представлена на рисунке 2.1.

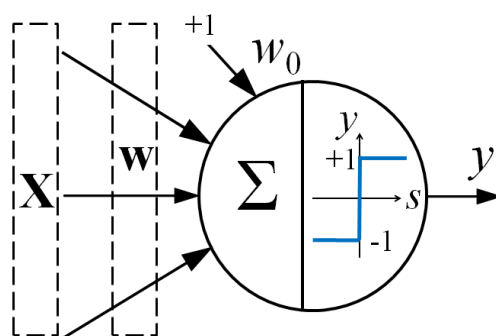


Рисунок 2.1 – Модель нейрона со ступенчатой функцией активации

Такой нейрон работает следующим образом:

1. На его входа подается набор (вектор) входных сигналов:

$$\mathbf{X} = (x_1, x_2, \dots, x_n)$$

2. Каждый входной сигнал умножается на соответствующий весовой коэффициент из вектора $\mathbf{w} = (w_1, w_2, \dots, w_n)$

3. Затем взвешенные сигналы суммируются со смещением w_0 нейрона:

$$s = w_0 + \sum_{i=1}^n x_i w_i, \quad (2.7)$$

где n – количество входных сигналов нейрона.

4. Затем к полученной сумме сигналов применяется функция активации, чтобы определить выходной сигнал нейрона:

$$y = f(s) \quad (2.8)$$

Поздней в место применения ступенчатой функции активации (как на рисунке 2.1) исследователями было предложено использование гладких функций активации, таких как сигмоида, представленная на рисунке 2.2:

$$f(x) := \frac{2}{1 + e^{-2x}} - 1$$

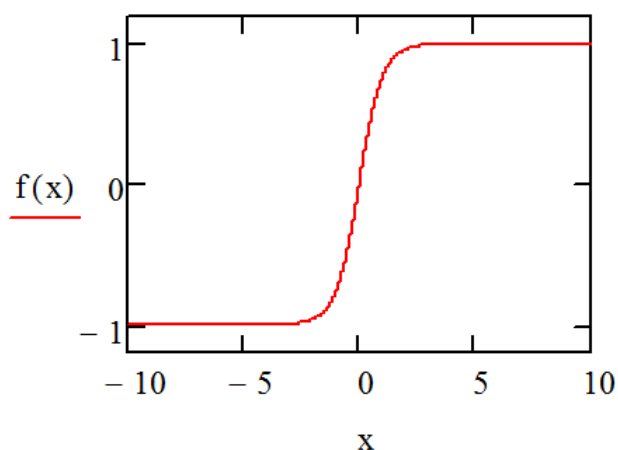


Рисунок 2.2 – Сигмоидная функция

При использовании с увеличением количества слоев нейронов и количества нейронов увеличивается вычислительные возможности нейронной сети. Пример архитектуры с использованием нескольких слоев представлен на рисунке 2.5.

Построение регрессионной модели с помощью нейронной сети заключается в поиске таких значений набора всех весовых коэффициентов \mathbf{W} , при которых нейронная сеть на своем выходе выдавала

максимально точный сигнал зависимой переменной y_i на всей обучающей выборке $D^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$, $i = 1, 2, \dots, m$.

Способов настройки весовых коэффициентов существует несколько, среди которых наиболее известные метод Левенберга-Маркара, метод Байесовской регуляризации и метод масштабируемых сопряженных градиентов.

Метод Левенберга-Маркара работает следующим образом. Сначала все весовые коэффициенты задаются случайным образом. Затем на каждой итерации производится корректировка весовых коэффициентов на величину Δw . Изменения в регрессионной модели (функции $f(w, x_n)$) на каждой итерации будут выглядеть так:

$$f(w + \Delta w, x) \approx f(w, x) + J \Delta w, \quad (2.9)$$

где J – якобиан функции $f(w, x_n)$ в точке w , представленная матрицей $N \times R$:

$$J = \begin{bmatrix} \frac{\partial f(w, x_1)}{\partial w_1} & \dots & \frac{\partial f(w, x_1)}{\partial w_R} \\ \dots & \dots & \dots \\ \frac{\partial f(w, x_N)}{\partial w_1} & \dots & \frac{\partial f(w, x_N)}{\partial w_R} \end{bmatrix} \quad (2.10)$$

В данном случае вектор параметров $w = [w_1, \dots, w_R]^T$.

От метода требуется на каждой итерации определение таких значений Δw , которые обеспечивают наименьшую E сумму квадратов ошибок на обучающей выборке.

В данном методе это достигается путем решения системы линейных уравнений:

$$\Delta w = (J^T J + \lambda I)^{-1} J^T (y - f(w)), \quad (2.11)$$

где $\lambda \geq 0$ – параметр регуляризации, который изменяется на каждой итерации метода, а I – единичная матрица

Итерации по уточнению весовых коэффициентов продолжается до тех пор пока Δw выше заданного значения.

В методе Байесовской регуляризации задача нахождения весовых коэффициентов формулируется как минимизация среднеквадратичной ошибки по параметрам нейронной сети. Для обеспечения решения вводится функция вида $\Omega(w)$. Она отвечает за предпочтительность выбора решения $h(x)$. Таким образом задача рассматривается как минимизация ошибки:

$$F = \sum_{i=1}^L (y_i - h(x_i))^2 + \lambda \Omega(h), \quad (2.12)$$

где λ – множитель отвечающий за масштабирование значимости $\Omega(h)$ относительно F

При этом выбор решения h ведется в рамках модели H , задающей ограничение на его вид. Вероятность выбора решения h для обучающей выборки D рассчитывается по формуле Байеса:

$$P(h | D, H) = \frac{P(D | h, H)P(h | H)}{P(D | H)} \quad (2.13)$$

Алгоритм работы метода масштабируемых сопряженных градиентов следующий:

1. $K=0$. Инициализация матрицы весовых коэффициентов W . Расчет градиента $G = \text{grad } E(W)$ и вектора направления $p_n = -\frac{G}{\|G\|}$. E – ошибка работы нейронной сети на обучающей выборке.

2. Поиск значения α , обеспечивающего минимизацию ошибки $E(W_0 + \alpha p)$. Корректировка значений W : $W(K + 1) = W(K) + \alpha p(K)$

3. Если достигнута требуемое точность работы нейронной сети, то остановка алгоритма и иначе рассчитать $G(k + 1) = \text{grad } E(W(k + 1))$

4. Вычислить β и вектор направления P_{k+1} :

$$\beta = \frac{G(K+1)^T G(K+1)}{G(K)^T G(K)}; P_{k+1} = -\frac{G(k+1) + \beta p(k)}{\| -G(k+1) + \beta p(k) \|}.$$

5. Присвоить $p(k)$ значением $p(k+1)$. Присвоить $G(k)$ значением $G(k+1)$. Вернуться на первый пункт.

Стоит отметить, что заранее нельзя сказать, какой метод настройки нейронной сети даст наилучшие результаты.

2.2 Обучающая выборка и структура нейронной сети

В качестве источника финансовых показателей различных компаний можно использовать один из общедоступных веб-сервисов, например «Yahoo Finance», расположенный по ссылке <https://finance.yahoo.com/>. Для доступа к финансовым данным у данного сервиса есть программный интерфейс Yahoo Finance API. Однако доступ к этому API предоставляется по платной подписке, поэтому от его использования было принято решение отказаться.

Одновременно с этим, все финансовые показатели различных компаний доступны через сайт сервиса (рисунок 2.3). Поэтому для получения показателей компании будет применяться технология парсинга страниц Yahoo Finance. Под парсингом понимается сбор данных со страницы и перевод их в структурированный вид.

Как видно из рисунка 2.3 для каждой компании мы можем получить доступ к следующим данным, которые будут использованы в качестве входных параметров нейронной сети:

- Price – цена одной акции на момент загрузки всех финансовых показателей;
- 52 Week min – минимальное значение цены акции за последние 52 недели;
- 52 Week max – максимальное значение цены акции за последние 52 недели;
- Average volume – средний объем ежедневных сделок по ценной бумаге;

- Market Capitalization – рыночная капитализация (произведение количества акций на текущую стоимость акций);

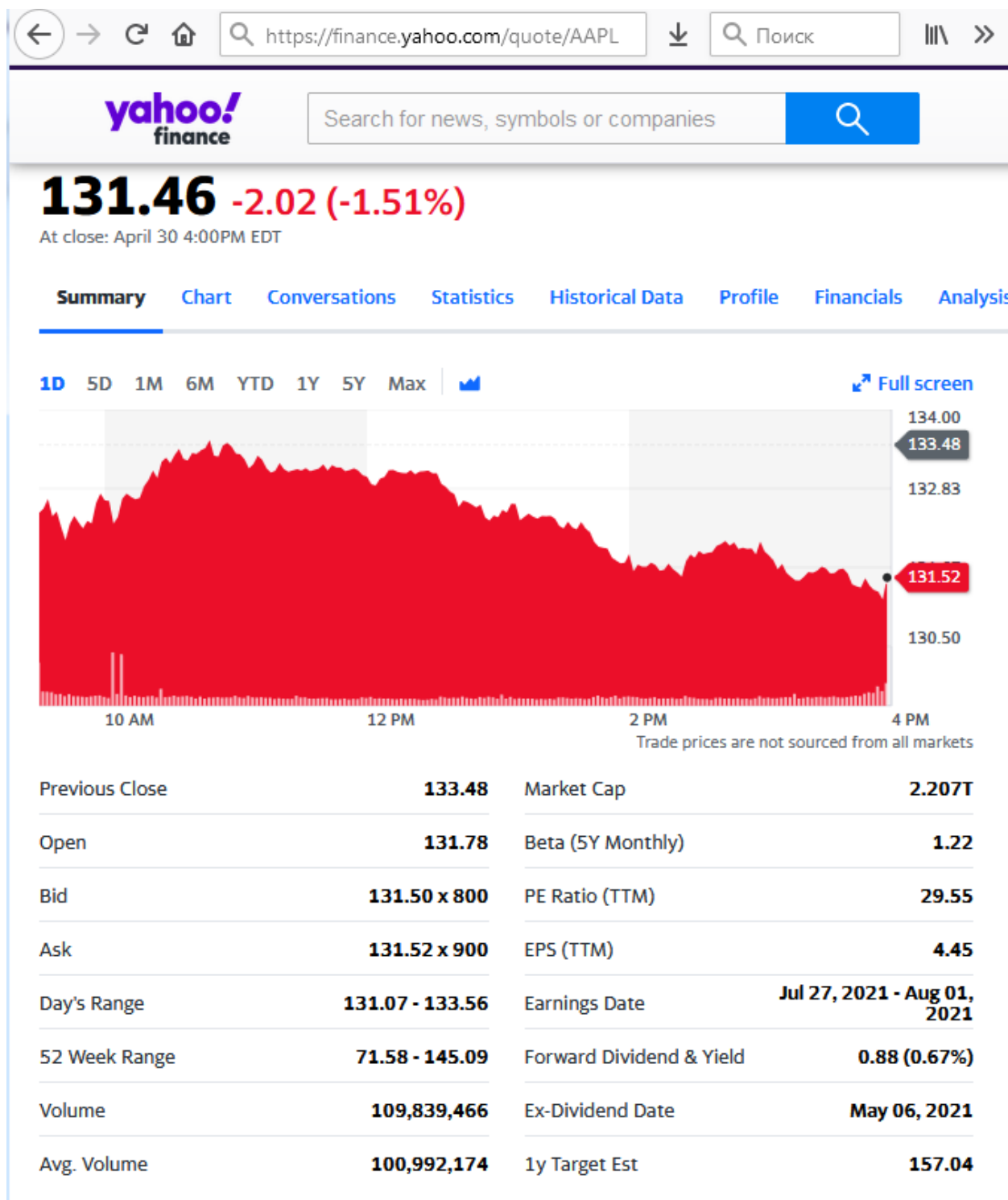


Рисунок 2.3 – Внешний вид страницы с финансовыми показателями компании Apple в Yahoo Finance, подвизраемой парсингу данных

- Beta – бета-коэффициент, показывающий изменчивость доходности ценной бумаги;
- PE Ratio (Price-to-Earnings Ratio) – отношению рыночной стоимости акции к годовой прибыли, полученной на акцию;
- EPS (Earnings per share) – отношение чистой прибыли к количеству акций на биржевом рынке;
- Forward Dividend – планируемые годовые дивиденды за год в долларах США;
- Yield – отношение планируемых годовых дивидендов к стоимости акции в процентах (другими словами – годовая доходность);
- 1y Target Est – ожидаемая цена на конец года, рассчитанная по жестко заданной формуле;
- Recommendation Rating – средняя оценка акции финансовых аналитиков, выраженная в виде числа от 1 до 5. Значение 5 означает рекомендацию к продаже акций, значение 1 рекомендацию к покупке.

Прогнозируемый параметр (выходной параметр) – процентное изменение стоимости акции в течение последующих 4 месяцев.

Ограничим набор рассматриваемых для анализа компаний. Так, как для граждан РФ покупка акций иностранных эмитентов обеспечивается посредством ПАО «Санкт-Петербургская Биржа», поэтому анализу будем подвергать все акции иностранных компаний, доступные через данную площадку. Санкт-Петербургская Биржа периодически расширяет список ценных бумаг, доступных к обороту, и на сегодняшний день доступны акции 1626 компаний.

$X = (\text{Price}, \text{52_Week_min}, \text{52_Week_max}, \text{Avg_volume}, \text{Market_Cap}, \text{Beta}, \text{PE}, \text{EPS}, \text{Forward_div}, \text{Yield}, \text{1y_Target_Est}, \text{Rec_Rating})$

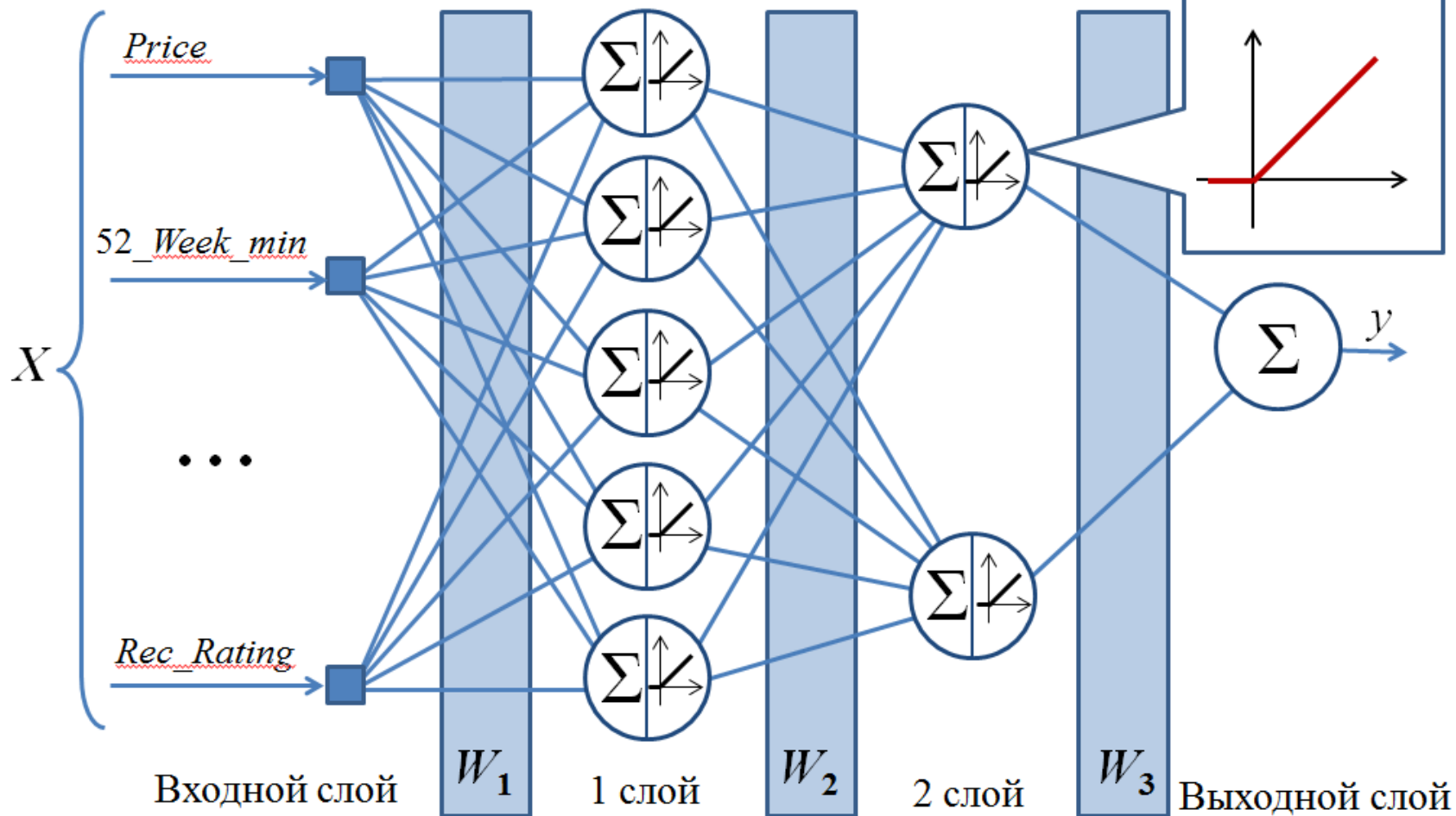


Рисунок 2.4 – Выбранная архитектура нейронной сети (двухслойная нейронная сеть прямого распространения)

Проведём анализ факторов, влияющих на выбор структуры нейронной сети:

- Размер обучающей выборки. Так как обучающая выборка для нейронной сети состоит из 1626 записей, то использование нейронных сетей с большим количеством слоев затруднительно. Это связано с тем, что в таких сетях большое количество весовых коэффициентов, которые должны быть подобраны в процессе обучения нейронной сети, а чем больше настраиваемых параметров, тем большего размера требуется обучающая выборка. Имеющийся размер обучающей выборки подходит для обучения двухслойной сети прямого распространения.

- Количество входов нейронной сети. По установленным эмпирически правилам Джеффом Хитом количество нейронов первого скрытого слоя должно быть меньше количеству входов нейронной сети. А количество нейронов последующих скрытых слоев должно быть меньше, чем в предыдущих слоях. Пользуясь этими правилами опытным путем было установлено, что оптимальная архитектура сети для имеющихся данных – 5 нейронов в первом слое, 2 нейрона во втором слое.

- Диапазон изменения входных значений нейронной сети. Функция активации нейронов выбирается в зависимости от диапазона изменения входных параметров нейронной сети, при необходимости используется нормировка значений путем линейного преобразования. Так как все 12 входных параметров могут принимать только положительные значения, то в качестве функции активации во всех нейронах будет использоваться функция $ReLU(S)$. В нашем случае нормировка данных не требуется.

Архитектура полученной нейронной сети, ее входные параметры и применяемая функция активации представлены на рисунке 2.4.

При таком сочетании параметров нейронной сети для ее обучения оптимальным будет использование алгоритма основанный на стохастическом градиентном спуске (алгоритм Adam).

2.5 Выводы по главе

Обосновано, что построение модели для прогнозирования роста стоимости акций компаний на основе их фундаментальных показателей можно свести к задаче построения регрессионной модели. В качестве регрессионной модели можно использовать нейронную сеть прямого распространения. Для выбора архитектуры нейронной сети был определен набор финансовых показателей, подаваемых на ее входной слой; проанализирован диапазон значений этих параметров, на основе которого выбрана функция активации (ReLU); определено количества и структура скрытых слоев сети (в первом слое – 5 нейронов, во втором слое – 2 нейрона).

В ходе исследований был определен оптимальный алгоритм обучения нейронной сети прямого распространения с выбранной архитектурой – алгоритм Adam, основанный на стохастическом градиентном спуске.

В качестве источника данных для формирования обучающей выборки был выбран сервис Yahoo Finance, а в качестве метода извлечения данных – парсинг страниц сервиса.

Теперь необходимо определить архитектуру разрабатываемого сервиса по прогнозированию изменения цен на акции.

3 РАЗРАБОТКА СЕРВИСА НЕЙРОСЕТЕВОГО ПРОГНОЗИРОВАНИЯ

3.1 Архитектура сервиса

Для нейросетевого сервиса прогнозирования стоимости акций выбрана следующая архитектура, которая представлена на рисунке 3.1. Основные элементы сервиса:

- GPU облако Google Colab;
- финансовый сервис Yahoo Finance;
- WEB-сервер (хостинг);
- клиент сервиса, получающий прогнозы с помощью браузера;
- разработчик, способный менять логику работы сервиса, посредством изменения python-кода в облаке Google Colab.

Рассмотрим детально работу всех элементов сервиса. Основной программный код, отвечающий за логику работы сервиса прогнозирования цен акций, выполняется в GPU облако Google Colab. Google Colab поддерживает автономное выполнение программного кода на языке Python. Помимо программного кода, Google Colab позволяет хранить данные в облаке, размер которого составляет несколько Гбайт.

Программный код, размещенный на Google Colab, выполняет следующие задачи сервиса прогнозирования цен на акции:

- парсинг данных финансовых данных из сервиса Yahoo Finance;
- хранение результатов парсинга в xls-файлах;
- обучение нейронной сети прямого распространения на финансовых показателях компаний;
- генерирование с помощью нейронной сети прогнозов по акция на ближайшие 4 месяца;
- визуализация прогнозов в виде изображений с водяным знаком (для исключения возможности парсинга данных сторонними сервисами);

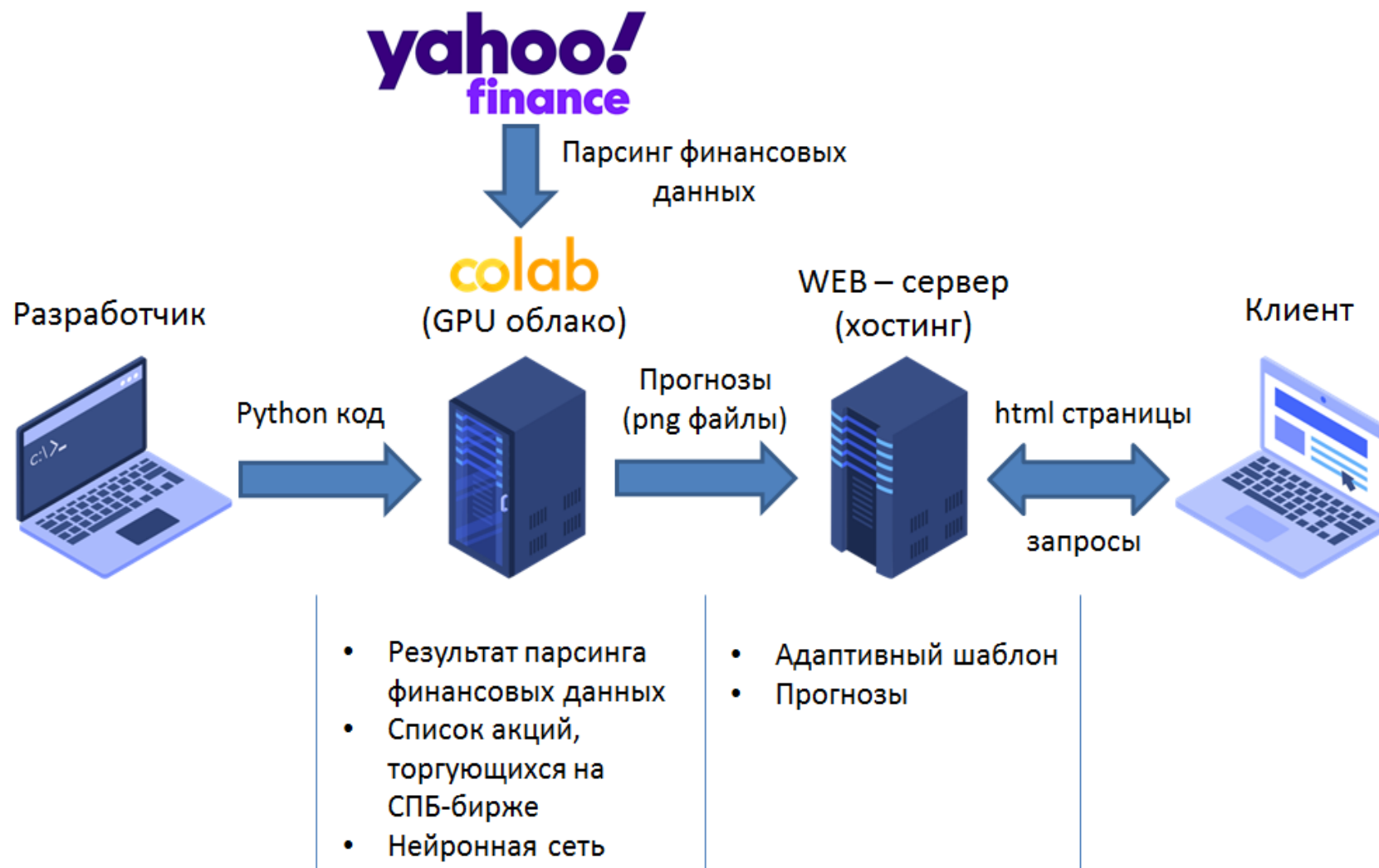


Рисунок 3.1 – Схема взаимодействия элементов системы для прогнозирования стоимости акций

Впоследствии прогнозы в виде png-файлов передаются на WEB-сервер (хостинг), где отображаются пользователю посредством html страниц с использованием технологий адаптивной верстке. На WEB-сервере для экономии средств услуга использования базы данных не подключена.

3.2 Программные решения, применяемые при реализации сервиса

Парсинг финансовых данных компаний организован посредством использования языка запросов к элементам XML-документа (XPath). Зная заранее путь XPath до требуемого элемента на веб-странице можно считывать хранящееся в нем значение. Путь XPath можно определить с помощью любого браузера с поддержкой консоли разработчика. Для этого требуется кликнуть правой кнопкой мыши по нужному элементу в консоли разработчика и выбрать пункт «Скопировать XPath» (рисунок 3.2).

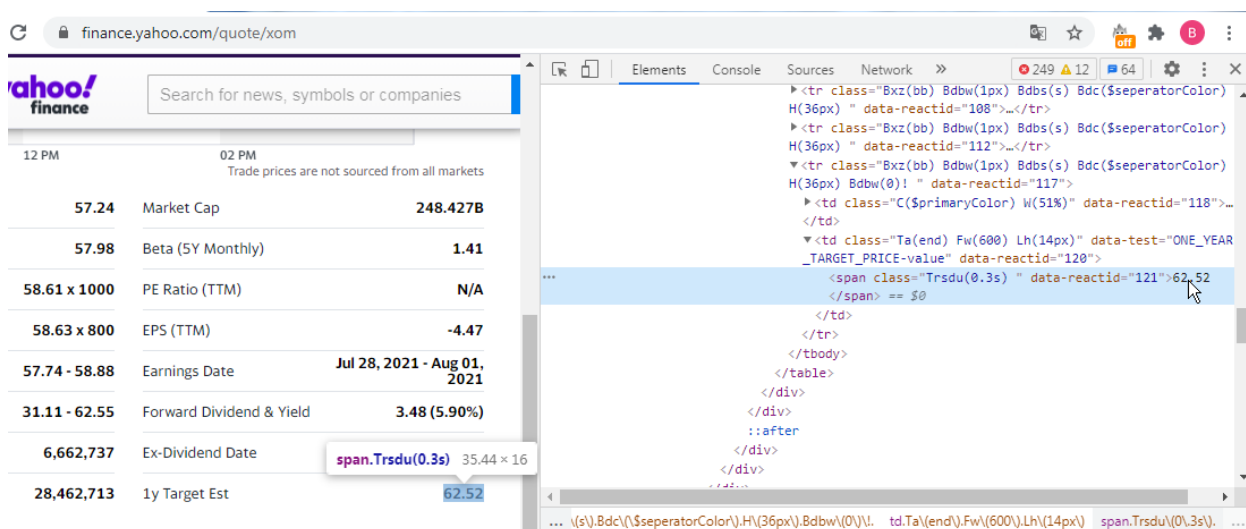


Рисунок 3.2 – Выбор элемента страницы, до которого надо определить XPath через консоль разработки браузера Google Chrome

Так как в сервисе Yahoo Finance структура страниц с финансовыми показателями для различных компаний повторяется, то пути XPath будут идентичными.

Для парсинга данных с Yahoo Finance воспользуемся следующими библиотеками python:

- pandas – данная библиотека предназначена для работы с данными, которые можно представить в табличном виде и содержит в себе удобные методы для сохранения данных в xls файлы;
- lxml – данная библиотека содержит методы обработки html файлов;
- requests – данная библиотека содержит методы для выполнения HTTP запросов;
- datetime – данная библиотека содержит методы для формирования временных меток (определения текущей даты);
- google.colab – данная библиотека содержит методы для взаимодействия с облачным хранилищем файлов Google drive.

Подключение требуемых библиотек показано на рисунке 3.3.

```
[54] import pandas as pd
      from lxml import html
      import requests
      import datetime

[55] from google.colab import drive
      drive.mount('/content/gdrive')
```

Drive already mounted at /content/gdrive; to attempt to forcibly remount, call d

Рисунок 3.3 – Код для подключения библиотек используемых при парсинге данных с Yahoo Finance

Определив через консоль разработчика пути до всех элементов страницы, содержащие необходимые финансовые показатели сформируем функцию readYF(), осуществляющую сбор данных. Единственный входной параметр функции – тикер компании. Тикер краткое символьное обозначение компании, например тикер ХОМ обозначает компанию Exxon mobil.

Внутри функции тикер храниться в переменной ticker. На первом этапе работы функции осуществляется построчная загрузка страницы как текстового файла, находящейся по адресу <https://finance.yahoo.com/quote/> с добавлением в конце тикера компании. Затем с помощью метода fromstring() осуществляется форматирование полученного текста в разметку страницы. И последовательно с помощью метода xpath() осуществляется загрузка текстовых данных находящихся в указанных элементах страницы. Полученные данные сохраняются в переменные price, week_range, avg_volume, market_cap, beta, PE, EPS, target_Est, dividends (рисунок 3.4).

```
def readYF(ticker):
    page = requests.get('https://finance.yahoo.com/quote/'+ticker)
    webpage = html.fromstring(page.content)
    price = float(webpage.xpath('//*[@id="quote-header-info"]/div[3]/div[1]/div/span[1]/text()')[0])
    week_range = webpage.xpath('//*[@id="quote-summary"]/div[1]/table/tbody/tr[6]/td[2]/text()')[0]
    avg_volume = webpage.xpath('//*[@id="quote-summary"]/div[1]/table/tbody/tr[8]/td[2]/span/text()')[0]
    market_cap = webpage.xpath('//*[@id="quote-summary"]/div[2]/table/tbody/tr[1]/td[2]/span/text()')[0]
    beta = webpage.xpath('//*[@id="quote-summary"]/div[2]/table/tbody/tr[2]/td[2]/span/text()')[0]
    PE = webpage.xpath('//*[@id="quote-summary"]/div[2]/table/tbody/tr[3]/td[2]/span/text()')[0]
    EPS = webpage.xpath('//*[@id="quote-summary"]/div[2]/table/tbody/tr[4]/td[2]/span/text()')[0]
    target_Est = webpage.xpath('//*[@id="quote-summary"]/div[2]/table/tbody/tr[8]/td[2]/span/text()')[0]
    dividends = webpage.xpath('//*[@id="quote-summary"]/div[2]/table/tbody/tr[6]/td[2]/text()')[0]
    return (price, week_range, avg_volume, market_cap, beta, PE, EPS, target_Est, dividends)
```

Рисунок 3.4 – Код основной функции для парсинга данных, основанной на использовании XPath

Для того, чтобы получить данные обо 1626 компаний торгующихся на СПБ бирже необходимо в переменную tickers (тип - list) загрузить список тикеров анализируемых компаний. Затем с помощью метода DataFrame() создать переменную df_dataset. В цикле for производится постепенный перебор всех тикеров из списка tickers. Для этого на каждой итерации вызывается функция readYF() и результат работы функции помещается в переменную df_dataset. Для контроля результата работы функции с помощью метода head() на экран выводится несколько первых элементов полученных с помощью парсинга данных (рисунок 3.5).

```
df_dataset = pd.DataFrame(columns=['Ticker', 'Price', '52 week range', 'Avg. Volume',
                                  'Market Cap', 'beta', 'PE', 'EPS', 'Target_st',
                                  'Dividend & Yield'])

for i in range(len(tickers)):
    data = readYF(tickers[i])
    df_dataset.loc[i] = [tickers[i], data[0], data[1], data[2], data[3], data[4],
                        data[5], data[6], data[7], data[8]]
```

```
df_dataset.head()
```

	Ticker	Price	52 week range	Avg. Volume	Market Cap	beta	PE	EPS	Target_st	Dividend & Yield
0	MMM	196.98	131.12 - 203.16	2,413,655	114.183B	0.96	20.16	9.77	194.05	5.92 (3.00%)
1	ABBV	112.68	79.11 - 113.41	7,000,978	198.861B	0.81	38.68	2.91	122.91	5.20 (4.66%)
2	AFL	54.22	30.32 - 54.87	3,600,501	36.905B	0.99	6.97	7.78	53.91	1.32 (2.46%)
3	ADM	64.34	33.01 - 64.53	2,580,334	35.947B	0.87	17.48	3.68	66.67	1.48 (2.34%)

Рисунок 3.5 – Формирование таблицы с результатами парсинга

Для того, чтобы сохранить хронологию загружаемых данных, результаты парсинга сохраняются в `xlsx` файл с названием, содержащим текущую дату. Для получения текущей даты используется метод `today()`. Для перевода даты в указанный строковый формат используется метод `strftime()`. С помощью инструкции `print` полученная дата выводится на экран. Для сохранения данных, содержащихся в переменной `df_dataset`, используется метод `to_excel()` (рисунок 3.6). В результат в облачном хранилище будет создан `xlsx`-файл с финансовыми данными 1626 компаний на указанную в имени файла дату.

```
[60] today = datetime.datetime.today()
      day = today.strftime("%d-%m-%Y")
      print(day)
```

```
[03-05-2021]
```

```
[61] df_dataset.to_excel('gdrive/My Drive/Colab Notebooks/fond/nn_'+day+'.xlsx')
```

Рисунок 3.6 – Сохранение результатов парсинга в `xls` файл

Для моделирования работы нейронной сети используется набор методов объекта MLPRegressor из библиотеки sklearn. Объект MLPRegressor представляет собой нейронную сеть прямого распространения и содержит в себе различные методы для работы с ней.

Сначала именем model создается экземпляр объекта MLPRegressor. При создании model конструктору передаются параметры hidden_layer_sizes (количество нейронов в каждом скрытом слое), activation (функция активации у нейронов скрытых слоев), solver (алгоритм обучения нейронной сети). Теперь с помощью метода fit() производится обучение нейронной сети на переданных в нее наборах данных. X_train – финансовые данные 1626 компаний полученные в предыдущем квартале, Y_train – текущее процентное изменение цен акций по сравнению с предыдущим кварталом. Обученная нейронная сеть используется для прогнозирования изменения цен акций в следующем квартале. Для этого через нейронную сеть обрабатываются данные этих 1626 компаний, но уже текущий квартал (эти данные содержатся в X_new). Результат работы нейронной сети сохраняется в переменную Y_pred (рисунок 3.7).

```
from sklearn.neural_network import MLPRegressor
model = MLPRegressor(hidden_layer_sizes=(5,2,), activation='relu', solver='adam')

%%time
model.fit(X_train, Y_train)

CPU times: user 933 ms, sys: 6.5 ms, total: 940 ms
Wall time: 949 ms
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:
  % self.max_iter, ConvergenceWarning)
MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
             beta_2=0.999, early_stopping=False, epsilon=1e-08,
             hidden_layer_sizes=(5, 2), learning_rate='constant',
             learning_rate_init=0.001, max_fun=15000, max_iter=200,
             momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
             power_t=0.5, random_state=None, shuffle=True, solver='adam',
             tol=0.0001, validation_fraction=0.1, verbose=False,
             warm_start=False)
```

```
Y_pred = model.predict(X_new)
```

Рисунок 3.7 – Задание и обучение нейронной сети

Функция `frame_transform` преобразует данные `X_new` и `Y_pred` в переменную типа `DataFrame`, оставляя только тикер компании (`Ticker`), цену акции (`Price`), размер дивидендов (`Dividend`), отношение планируемых годовых дивидендов к стоимости акции в процентах (`Yield`) и прогнозируемое изменение цены в акций (`Target`) (рисунок 3.8).

Для визуализации полученных от нейронной сети прогнозов разработана функция `render_mpl_table()`, которая с помощью методов библиотеки `matplotlib` генерирует изображение с прогнозами в формате `png` (рисунок 3.8). В функции также предусмотрена возможность наложения водяных знаков на изображение, путем передачи строковых переменных `copyr_str` и `labelf_str`. Подписи изображения сверху в функции предусмотрено получения текстовых данных посредством переменной `label_str`. Также с помощью входных переменных `header_color` и `row_color` обеспечена возможность задания цветовой палитры изображения (рисунок 3.8).

```
dfm = frame_transform_nn(X_new, Y_pred)

def render_mpl_table(copyr_str, labelf_str, label_str, data, col_width=3.0, row_height=0.625, font_size=14,
                    header_color='#800000', row_colors=['#f1f1f2', 'w'], edge_color='w',
                    bbox=[0, 0, 1, 1], header_columns=0,
                    ax=None, **kwargs):
    if ax is None:
        size = (np.array(data.shape[::-1]) + np.array([0, 1])) * np.array([col_width, row_height])
        fig, ax = plt.subplots(figsize=size)
        ax.axis('off')
    mpl_table = ax.table(cellText=data.values, bbox=bbox, colLabels=data.columns, **kwargs)
    mpl_table.auto_set_font_size(False)
    mpl_table.set_fontsize(font_size)
    ax.text(0, 1.05, label_str, rotation = 0, fontsize = 15, alpha=1)
    ax.text(0.1, 0.1, copyr_str, rotation = 30, fontsize = 50, alpha=0.15)
    ax.text(0.6, 0.5, copyr_str, rotation = 30, fontsize = 50, alpha=0.15)
    ax.text(0.1, 0.8, copyr_str, rotation = 30, fontsize = 50, alpha=0.15)
    ax.text(0.05, 0, labelf_str, rotation = 0, fontsize = 14, alpha=1)
    for k, cell in mpl_table._cells.items():
        cell.set_edgecolor(edge_color)
        if k[0] == 0 or k[1] < header_columns:
            cell.set_text_props(weight='bold', color='w')
            cell.set_facecolor(header_color)
        else:
            cell.set_facecolor(row_colors[k[0]%len(row_colors) ])
    return ax.get_figure(), ax
```

Рисунок 3.8 – Функция для генерирования изображения с прогнозами

С помощью переменных `copyr_str` задается текст отображаемый на водяных знаках, переменная `labelf_str` отвечает за текст отображаемый под рисунком, а переменная `label_str` отвечает за заголовок отображаемый над рисунком заголовков. В переменную `label_str` посредством переменной `nowtime_str` интегрирована отображение даты в которую составлен представленный на рисунке прогноз (рисунок 3.9).

Результатом выполнения функции `render_mpl_table()` является изображение с прогнозом, которое сохраняется в переменную `fig`. С помощью метода `savefig()` производится сохранение изображения в облачное хранилище Google. Сгенерированное изображение для удобства контроля выводится на экран (рисунок 3.10).

```
copyr_str = '2inv.ru'
labelf_str = '2inv.ru'
label_str = nowtime_str + ' Predicting changes in stock prices '
```

```
fig,ax = render_mpl_table(copyr_str, labelf_str, label_str, dfm, header_columns=0, col_width=1.8)
fig.savefig("gdrive/My Drive/Colab Notebooks/fond/table_nn.png", dpi=120, bbox_inches='tight')
fig.clear
ax.clear
```

```
<bound method _AxesBase.clear of <matplotlib.axes._subplots.AxesSubplot object at 0x7fa79accf510>>
[01-05-2021] Predicting changes in stock prices
```

Ticker	Price, \$	Dividend, \$	Yield, %	TARGET, %
ARCC	19.25	1.6	7.23	17.55
SWM	45.67	1.76	3.35	12.52
WMB	24.36	1.64	5.86	10.93
MO	47.75	3.44	6.27	10.53
VLO	73.96	3.92	4.61	10.33
ET	8.61	0.61	4.46	9.16
GLPI	46.49	2.6	3.91	8.77
OHI	38.0	2.68	4.94	8.29
PM	95.0	4.8	4.4	8.21

Рисунок 3.9 – Изображение, сгенерированное на основе прогнозов

Для передачи изображения с прогнозами из GPU облака Google на WEB-сервер с сайтом сервиса используется связка криптографического протокола TLS и протокола передачи файлов FTP.

В этой части кода использовались следующие библиотеки:

- `ftplib` – библиотека, реализующая методы для работы с протоколом ftp;
- `ssl` – библиотека, обеспечивающая работу с криптографическими методами;
- `os` – библиотека, содержащая методы для работы с файлами в облачном хранилище.

С помощью переменной `send` осуществляется проверка на наличие файлов предназначенных для отправки в WEB-сервис. Если такие файлы есть, то используется метод `FTP_TLS()`, осуществляющий подключение к WEB-серверу с использованием криптографического протокола TLS. Затем с помощью метода `login()` производится авторизация на сервере. С помощью метода `cwd()` производится выбор категории для копирования сгенерированного изображения с прогнозами. Затем с помощью `storbinary()` осуществляется отправка файла на WEB-сервер.

```
import ftplib
import os
import ssl

if send == 1:
    ftp_session = ftplib.FTP_TLS(ftp_server)
    ftp_session.login(my_userid, my_passwd)

f = open('gdrive/My Drive/Colab Notebooks/fond/table_nn.png', 'rb')
ftp_session.cwd("/http/images")
ftp_session.storbinary("STOR "+ 'table_nn.png', f, 1024)

'226 Transfer complete.'

ftp_session.quit()
```

Рисунок 3.10 – Код, отвечающий за от отправку сгенерированного изображения на WEB-сервер

Как только сервер пришлет ответ “226 Transfer complete” с помощью метода quit() производится разрыв соединения с сервером (рисунок 3.10).

The screenshot shows a web browser window with the URL `2inv.ru/neuralnet.html`. The page content includes a header with the 2inv.ru logo and navigation links. The main heading is "Нейросетевой анализ" (Neural Network Analysis), followed by a subtitle: "Прогнозирование стоимости акции с использованием нейронной сети. Правила пользования." (Forecasting stock price using a neural network. Rules of use). A teal button labeled "НА ГЛАВНУЮ СТРАНИЦУ" (Back to Home) is present.

The main content area is titled "ПРОГРАММНАЯ АНАЛИТИКА" (Programmatic Analytics) and contains the following information:

[01-05-2021] Predicting changes in stock prices

Ticker	Price, \$	Dividend, \$	Yield, %	TARGET, %
ARCC	19.25	1.6	7.23	17.55
SWM	45.67	1.76	3.35	12.52
WMB	24.36	1.64	5.86	10.93
MO	47.75	3.44	6.27	10.53
VLO	73.96	3.92	4.61	10.33
ET	8.61	0.61	4.46	9.16
GLPI	46.49	2.6	3.91	8.77
OHI	38.0	2.68	4.94	8.29
PM	95.0	4.8	4.4	8.21
PSX	80.91	3.6	3.87	7.95
GNL	19.2	1.6	5.83	7.62
ABBV	111.5	5.2	4.06	7.57
MPC	55.65	2.32	3.63	7.45
NAVI	16.83	0.64	3.31	7.41
STRA	75.06	2.4	2.78	7.27

- TARGET, % - прогнозируемое изменение цены акции на интервале не более 4 месяцев.

Прогноз составлен на основе:

- Цены акции на момент сбора данных
- Минимального и максимального значений цена за последние 52 недели
- Среднего объема ежедневных сделок
- Рыночной капитализации
- Price-to-Earnings Ratio (PE)
- Earnings per share (EPS)
- Размер дивидендов
- Ожидаемая цена на конец года (рассчитывается по формуле)

Рисунок 3.11 – Страница сервиса 2inv.ru с прогнозами нейронной сети

По ссылке `2inv.ru/neuralnet.html` всегда доступен актуальный прогноз изменения цен акций, подготовленной с помощью нейронной сети. Вид страницы показан на рисунке 3.11.

3.3 Обсуждение результатов прогнозирования

Как показывает практика, при фундаментальном анализе нейронная сеть не может предсказать цену акции на точную цену. Но при инвестировании этого и не требуется. Дело в том, что при торговле на бирже любым брокером обеспечена возможность выставления заявок на продажу типа «Take-profit». Данная заявка на продажу акции срабатывает автоматически при достижении указанной цены акции.

Предполагается, что разработанный сервис прогнозирования будет использован для выставления заявок «Take-profit». По этой причине возможны следующие варианты исхода прогнозов (рисунок 3.12):

1. Цена в течение прогнозируемого периода (4 месяца) вырастет до прогнозного значения. В этом случае акция в соответствии с заявкой «Take-profit» будет автоматически продана, а инвестор заработает разницу между курсами. Прогноз в этом случае будет считаться сработавшим.

2. Цена в течение прогнозируемого периода вырастет до прогнозного значения, но рост цены продолжится. В этом случае акция в соответствии с заявкой «Take-profit» все равно будет продана, а инвестор заработает разницу между курсами. Прогноз в этом случае будет считаться сработавшим.

3. Цена в течение прогнозируемого периода не вырастет до прогнозного значения. В этом случае акция по заявке «Take-profit» продана не будет. Прогноз в этом случае будет считаться не сработавшим.

4. Цена на акцию достигнет прогнозного значения за пределами прогнозного периода. В этом случае акция в соответствии с заявкой «Take-profit» все будет продана, а инвестор заработает разницу между курсами. Но прогноз в этом случае будет считаться не сработавшим.

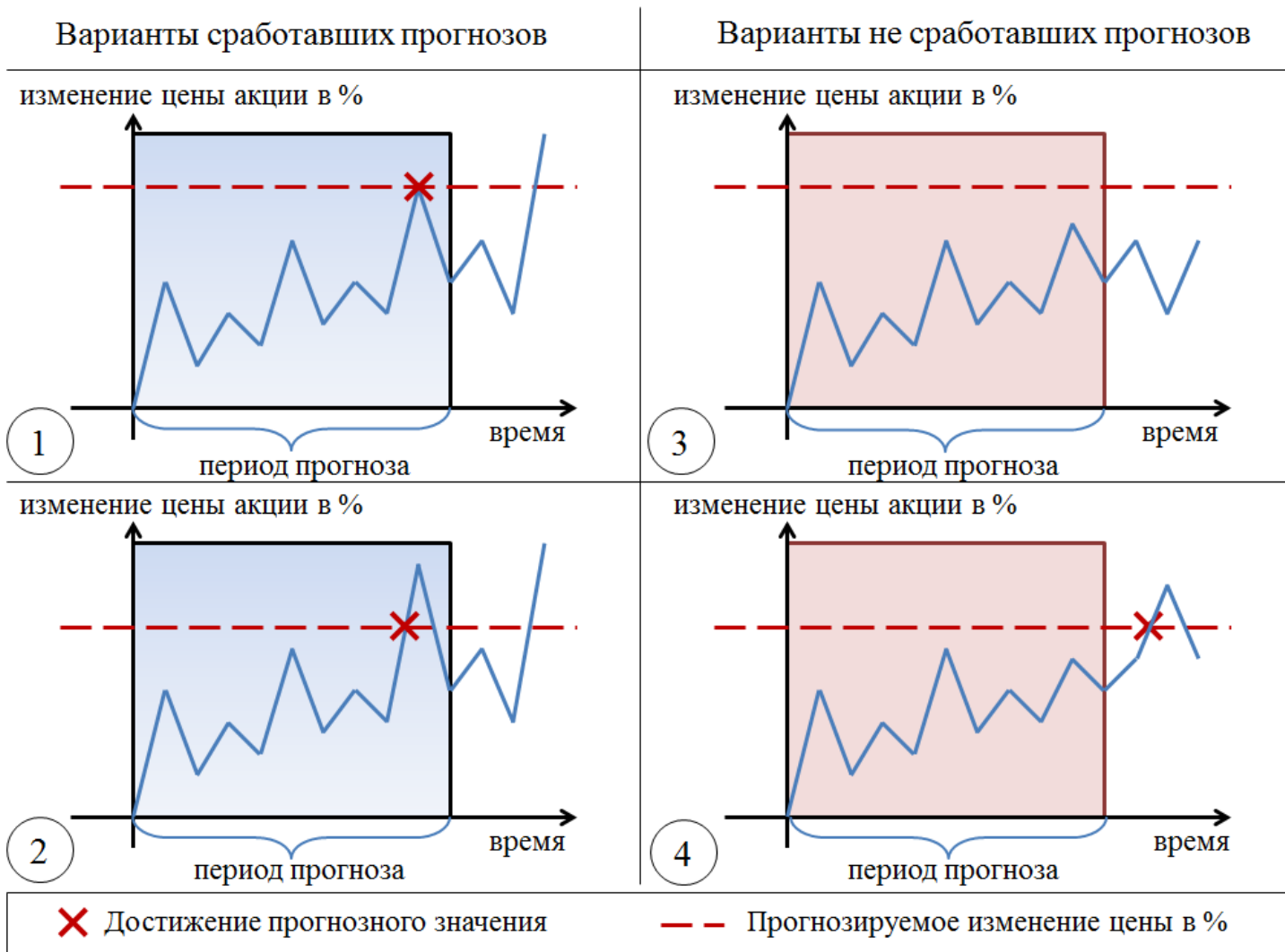


Рисунок 3.12 – 4 варианта исхода прогнозов

С применением данного подхода было установлено, что около 75% прогнозов, сгенерированных нейронной сетью сбываются. При этом следует учитывать, что данные результаты были получены во время «бычьего» цикла рынка. Поэтому в период «медвежьего» цикла рынка к прогнозам системы надо относиться с осторожностью.

3.4 Выводы по главе

Разработана структура сервиса прогнозирования цен на акции с использованием нейронной сети, включающей в себя, в том числе, GPU облако Google Colab и WEB-сервис. GPU облако используется для парсинга финансовых данных различных компаний с сервиса Yahoo Finance с их последующим использованием для обучения нейронной сети. Обученная нейронная сеть используется для прогнозирования изменения стоимости акций в течение 4 месяцев. GPU облако формирует изображение с текущими прогнозами, устанавливает соединение с WEB-сервером (хостингом) и отправляет туда полученное изображение. Пользователь, заходя на сайт проекта (2inv.ru), в последствии, может получить информацию с прогнозами

На языке Python разработан программный код, реализующий работу сервиса.

Исследование результатов прогнозирования показывает, что в период «бычьего» рынка сбывается около 75% процентов сформулированных нейронной сетью прогнозов.

ЗАКЛЮЧЕНИЕ

В ходе написания балкарской работы были получены следующие результаты:

1. Проведен анализ походов к построению прогнозных моделей на основе фундаментального анализа финансовых показателей компаний, представленных на фондовом рынке. Выявлен недостаток существующих подходов связанный с отсутствием адаптируемости прогнозных моделей к внешним экономическим изменениям.

2. Для преодоления данного недостатка предложена схема построения прогнозной модели, основанной на использовании финансовых показателей за предыдущий квартал для прогнозирования цен акций в следующем квартале. При этом адаптируемость прогнозной модели обеспечивается наличием процедуры ее ежеквартальном переобучении исключительно на актуальных квартальных данных. В качестве прогнозной модели предложено использовать нейронную сеть.

3. Обосновано, что построение модели для прогнозирования роста стоимости акций компаний на основе их фундаментальных показателей можно свести к задаче построения регрессионной модели. В качестве регрессионной модели можно использовать нейронную сеть прямого распространения. Для выбора архитектуры нейронной сети был определен набор финансовых показателей, подаваемых на ее входной слой; проанализирован диапазон значений этих параметров, на основе которого выбрана функция активации (ReLU); определено количества и структура скрытых слоев сети (в первом слое – 5 нейронов, во втором слое – 2 нейрона).

4. В ходе исследований был определен оптимальный алгоритм обучения нейронной сети прямого распространения с выбранной

архитектурой – алгоритм Adam, основанный на стохастическом градиентном спуске.

5. В качестве источника данных для формирования обучающей выборки был выбран сервис Yahoo Finance, а в качестве метода извлечения данных – парсинг страниц сервиса.

6. Разработана структура сервиса прогнозирования цен на акции с использованием нейронной сети, включающей в себя GPU облако Google Colab и WEB-сервис. GPU облако используется для парсинга финансовых данных различных компаний с сервиса Yahoo Finance с их последующим использованием для обучения нейронной сети. Обученная нейронная сеть используется для прогнозирования изменения стоимости акций в течение 4 месяцев. GPU облако формирует изображение с текущими прогнозами, устанавливает соединение с WEB-сервером (хостингом) и отправляет туда полученное изображение. Пользователь, заходя на сайт проекта (2inv.ru), в последствии, может получить информацию с прогнозами

7. На языке Python разработан программный код, реализующий работу сервиса.

8. Исследование результатов прогнозирования показывает, что в период «бычьего» рынка сбывается около 75% процентов сформулированных нейронной сетью прогнозов.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Абрамов, Н.С. Нейросетевые технологии поиска целевых объектов на снимках дистанционного зондирования земли / Н.С. Абрамов, А.А. Талалаев, В.П. Фраленко, О.Г. Шишкин, В.М. Хачумов // V Международная конференция и молодежная школа "Информационные технологии и нанотехнологии" ИТНТ-2019, Самара, 21–24 мая 2019 года. – Издательство: Новая техника, 2019. – с. 26-33. – Текст : непосредственный.
2. Андриенко М.П. Понимание повторяющихся нейронных сетей: предпочтительная нейронная сеть для данных временных рядов / М.П. Андриенко, П.А. Юдин, Е.Ю. Вишневецкая // Актуальные вопросы современной науки и образования: сборник статей Международной научно-практической конференции : в 2 ч.. 2020. – Издательство: Наука и Просвещение (Пенза), 2020. – с. 96-98. – Текст : непосредственный.
3. Близко, М.В. Нейронные сети. Реализация нейронных сетей в MATLAB / М.В. Близко // Искусственный интеллект. Теоретические аспекты, практическое применение (ИИ-2020). Донецк, 27 мая 2020 года. – Издательство: Государственное учреждение Институт проблем искусственного интеллекта (Донецк), 2020. – с. 23-26. – Текст : непосредственный.
4. Голоскоков, К.П. Применение нейронных сетей в задачах прогнозирования и проблемы идентификации моделей прогнозирования на нейронных сетях / К.П. Голоскоков // Современные проблемы прикладной информатики: сборник научных трудов. – Издательство: Санкт-Петербургский государственный инженерно-экономический университет, 2006. – с. 116-120. – Текст : непосредственный.
5. Грушевская, А.Л. Сравнительный анализ решения одной задачи классификации четырьмя типами нейронных сетей / А.Л. Грушевская, А.Н. Покровский // Современные методы прикладной математики, теории управления и компьютерных технологий (ПМТУКТ-2013) : сборник трудов

VI международной конференции, 10–16 сентября 2013 года. – Издательство: Воронежский государственный университет (Воронеж), 2013. – с. 83-84. – Текст : непосредственный.

6. Дорогов, А.Ю. Нейронные сети глубокого обучения с управляемой коммутацией нейронных плоскостей / А.Ю. Дорогов // Дистанционные образовательные технологии: Материалы IV Всероссийской научно-практической конференции (с международным участием), 2019. – Издательство: Общество с ограниченной ответственностью «Издательство Типография «Ариал» (Симферополь), 2019. – с. 284-296. – Текст : непосредственный.

7. Ибрагимов Р.М. Влияние функций активации нейронных сетей на скорость обучения на примере нейронной сети с обратным распространением ошибки / Р.М. Ибрагимов // Актуальные проблемы физической и функциональной электроники: материалы 21-й Всероссийской молодежной научной школы-семинара. 2018. – Издательство: Ульяновский государственный технический университет (Ульяновск), 2018. – с. 125-126. – Текст : непосредственный.

8. Клячин В.Н. Использование агрегированных классификаторов при технической диагностике на базе машинного обучения / В.Н. Клячин, Ю.Е. Кувайскова, Д.А. Жуков // Информационные технологии и нанотехнологии (ИТНТ-2017) – сборник трудов III международной конференции и молодежной школы. Самарский национальный исследовательский университет имени академика С.П. Королева. 2017. – Предприятие "Новая техника" (Самара), 2017. – с. 1770-1773. – Текст : непосредственный.

9. Кононова, Н.В. Исследование подсистемы контентной фильтрации с использованием методов машинного обучения / Н.В. Кононова, Ю.А. Андрусенко, Т.А. Самокаева // Студенческая наука для развития информационного общества – сборник материалов VI Всероссийской научно-технической конференции. 22–26 мая 2017. – Северо-Кавказский

федеральный университет (Ставрополь), 2017. – с. 268-270. – Текст : непосредственный.

10. Лебеяднцев, В.В. Влияние архитектуры нейронной сети и исходных данных на работу нейронной сети для задач классификации / В.В. Лебеяднцев, М.И. Озерова // Информационные технологии в науке и производстве : материалы VII Всероссийской молодежной научно-технической конференции, 2020. – Издательство: Омский государственный технический университет (Омск), 2020. – с. 145-152. – Текст : непосредственный.

11. Малинин, П.В. Иерархический подход в задаче идентификации личности по голосу с помощью проекционных методов классификации многомерных данных / П.В. Малинин, В.В. Поляков // Доклады томского государственного университета систем управления и радиоэлектроники. №21, 2010. – Томский государственный университет систем управления и радиоэлектроники (Томск), 2010. – с. 128-130. – Текст : непосредственный.

12. Мелдебай, М.А. Анализ мнений покупателей на основе машинного обучения / М.А. Мелдебай, А.К. Сарбасова // Прикладная математика и информатика: современные исследования в области естественных и технических наук – материалы III научно-практической всероссийской конференции (школы-семинара) молодых ученых. 24–25 апреля 2017 года. – Издатель Качалин Александр Васильевич, 2017. – с. 360-363. – Текст : непосредственный.

13. Наумов, Д.П. Регулятор САР на основе машинного обучения / Д.П. Наумов, Д.П. Стариков // Информационные технологии в управлении, автоматизации и мехатронике – сборник научных трудов Международной научно-технической конференции. 06–07 апреля 2017 года. – ЗАО "Университетская книга" (Курск), 2017. – с. 106-114. – Текст : непосредственный.

14. Синягов, А.И. Реализация искусственной нейронной сети на базе нейронной сети Петри / А.И. Синягов, А.А. Суконщиков // Шаг в будущее:

искусственный интеллект и цифровая экономика. Материалы 1-й Международной научно-практической конференции. Государственный университет управления. 2017. – Издательство: Государственный университет управления (Москва), 2017. – с. 130-135. – Текст : непосредственный.

15. Урубкин, М.Ю. Нейронные сети Кохонена и нечеткие нейронные сети в интеллектуальном анализе данных / М.Ю. Урубкин, А.В. Авакьянц // Совершенствование методологии познания в целях развития науки: сборник статей по итогам Международной научно-практической конференции: в 2 частях. 2017. – Издательство: Общество с ограниченной ответственностью "Агентство международных исследований" (Уфа), 2017. – с. 36-39. – Текст : непосредственный.

16. Catak, F.O. Deep Neural Network Based Malicious Network Activity Detection Under Adversarial Machine Learning Attacks / Ferhat Ozgur Catak, Sule Yildirim Yayilgan // Intelligent Technologies and Applications: Third International Conference, INTAP 2020, Grimstad, Norway, September 28–30, 2020, Revised Selected Papers. – Springer Nature Switzerland AG, 2021. – pp. 280-291. – Текст : непосредственный.

17. Chandre, P.R. Intrusion Detection and Prevention Using Artificial Neural Network in Wireless Sensor Networks / Pankaj R. Chandre, Parikshit N. Mahalle, Gitanjali R. Shinde // Proceeding of First Doctoral Symposium on Natural Computing Research (RSNCR 2020). – Springer Nature Singapore Pte Ltd. 2021. – pp. 113-121. – Текст : непосредственный.

18. Liu, Zh. Human Activities Recognition from Videos Based on Compound Deep Neural Network / Zhijian Liu, Yi Han, Zhiwei Chen, Yuelong Fang, Huimin Qian, Jun Zhou // Advances in Intelligent Systems and Computing : The 10th International Conference on Computer Engineering and Networks (CENet 2020). – Springer Nature Singapore Pte Ltd. 2021. – pp. 314-326. – Текст : непосредственный.

19. Singh, M.M. A Technique to Detect Wormhole Attack in Wireless Sensor Network Using Artificial Neural Network / Moirangthem Marjit Singh, Nishigandha Dutta, Thounaojam Rupachandra Singh, Utpal Nandi // Evolutionary Computing and Mobile Sustainable Networks (Proceedings of ICECMSN 2020). – Springer Nature Singapore Pte Ltd. 2021. – pp. 297-307. – Текст : непосредственный.

20. Zhao, C. Recommendation Based Heterogeneous Information Network and Neural Network Model / Cong Zhao, Yan Wen, Ming Chen, Geng Chen // International Conference on Wireless and Satellite Systems: 11th EAI International Conference, WiSATS 2020, Nanjing, China, September 17-18, 2020, Proceedings, Part II. – Springer Nature Switzerland AG, 2021. – pp. 588-598. – Текст : непосредственный.