МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)
Институт математики и компьютерных технологий
Департамент программной инженерии и искусственного интеллекта

Денисенко Семен Денисович

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

бакалаврская работа
вид ВКР

РАЗРАБОТКА ПРОГРАММНОЙ СИСТЕМЫ ДЛЯ ТРЕЙДИНГА НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ. ПОДСИСТЕМА ГЕНЕРАЦИИ ОБУЧАЮЩИХ ВЫБОРОК И ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

по направлению подготовки (специальности) 09.03.04 «Программная инженерия»
профиль «Программная инженерия»

Владивосток
2023

В материалах данной выпускной квалификационной работы не содержатся сведения, составляющие государственную тайну, и сведения, подлежащие экспортному контролю	Автор работы подпись группа Б9119-09.03.04прогин « » 2023 г.
Уполномоченный по экспортному контролю И.Л. Артемьева подпись И.О. Фамилия « » 2023 г.	Руководитель ВКР доцент, к.ф.-м.н. должность, ученое звание В.Н. Лиховидов подпись И.О. Фамилия « » 2023 г.
Защищена в ГЭК с оценкой Секретарь ГЭК О.А. Крестникова подпись И.О. Фамилия « » 2023 г.	**«Допустить к защите»** и.о. директора департамента ученая степень, ученое звание О.А. Крестникова подпись И.О. Фамилия « » 2023 г.

Оглавление

Введение

Прогнозирование валютных рынков – актуальная задача, стоящая перед многими участниками мировой экономики. В частности, валютный рынок сегодня является сферой интересов многочисленных трейдеров и международных компаний, основная цель деятельности которых получить прибыль. Трейдером называют участника биржевой торговли, который вкладывает свои средства в биржевой актив и может впоследствии его продать. В качестве биржевого актива могут выступать валюты государств, акции, облигации, криптовалюты. Сектор биржи, на котором торгуют валютой, называется валютной биржей. Основополагающим фактором успеха в торговле на валютной бирже является предсказание будущих котировок. Чтобы составить торговую стратегию необходимо проанализировать множество экономических показателей. При этом анализируются данные за большой отрезок времени. Более того, объем данных, который необходимо обработать, увеличивается со временем, поэтому обработка поступающего объема информации становится всё более трудоёмким процессом. Также следует отметить постоянную изменчивость валютных рынков. Таким образом, решение задачи прогнозирования валютных рынков остаётся востребованным.

На данный момент машинное обучение представляет собой перспективный метод прогнозирования событий на рынке, широко применяемый в сфере бизнеса. Особый интерес вызывают искусственные нейронные сети, которые возникли в результате исследования мозговых процессов, связанных с мышлением, и попытки их моделирования. Впоследствии эти модели стали использоваться для решения практических задач, особенно в области классификации. Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Технически обучение заключается в определении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна обнаруживать сложные зависимости между входными и выходными данными, а также проводить обобщение. Ее

способность к классификации непосредственно следует из способности обобщать и выявлять скрытые зависимости между входными и выходными данными. После завершения обучения нейронная сеть может классифицировать элементы последовательности на основе предыдущих значений или текущих факторов [11].

Торговый робот, также известный как автоматическая торговая система, представляет собой программное обеспечение, которое автоматически выполняет торговые операции на финансовых рынках без прямого участия человека. Он использует заранее заданные правила и алгоритмы для принятия решений о входе и выходе с рынка, определения размеров позиций и управления рисками. Торговый робот работает на основе программирования и анализа данных рынка. Он может использовать различные технические индикаторы, статистические модели и другие алгоритмы для определения торговых сигналов. Когда определенные условия соблюдаются, робот может автоматически размещать ордера на покупку или продажу активов [10].

Использование торговых роботов в финансовой торговле становится все более распространенным. Они могут быть полезны в различных аспектах торговли, включая автоматизацию, точность, скорость и управление рисками. Торговые роботы могут автоматизировать процесс принятия решений и выполнения торговых операций. Они могут следить за рыночной ситуацией, анализировать данные и исполнять заранее заданные торговые стратегии без необходимости постоянного присутствия трейдера. Роботы могут обрабатывать большие объемы данных и выполнить сложные расчеты за кратчайшее время. Это позволяет им принимать решения на основе точных и актуальных данных и реагировать на изменения рыночной ситуации мгновенно. Такая точность и скорость обычно выше, чем у человеческих трейдеров. С постоянным ростом доступных данных и развитием алгоритмов машинного обучения, торговые роботы могут использовать более сложные модели и анализировать большие объемы информации для принятия решений. Это может улучшить качество прогнозирования и повысить эффективность торговли.

Цель выпускной квалификационной работы состоит в разработке программной системы для трейдинга на основе нейронных сетей. Под работкой системы для трейдинга подразумевается разработка программного обеспечения для автоматической торговли, а также разработка алгоритма генерации обучающих выборок для нейронных сетей.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1) провести обзор рассматриваемой предметной области;
2) провести анализ предметной области и построить ее модель;
3) сформировать технический проект программной системы;
4) реализовать и провести тестирование программной системы.

1 Обзор предметной области

В данной главе будут выделены основные объекты предметной области, описана деятельность экспертов в предметной области и обоснована актуальность разрабатываемого программного средства.

1.1 Основные понятия

Актив [16] – совокупность имущества, принадлежащего юридическому лицу или предпринимателю.

Паттерн [16] – это повторяющаяся структура на ценовом графике, которая может содержать информацию о будущем направлении цены.

Торговый робот [16] – реализованный алгоритм, автоматизирующий торговлю на бирже. Роботы могут отличаться по методам реализации и целям, однако в основе всех них лежит принцип минимизации участия человека в процессе.

1.2 Валютный рынок Forex

Forex (Foreign Exchange Market) – международный рынок по торговле валютой. Валютный рынок представляет собой систему экономических и организационно-правовых отношений по операциям купли-продажи иностранных валют и платежных документов в иностранных валютах.

Торговля на рынке Forex осуществляется с помощью специальных контрактов, именуемых валютными парами. Денежные знаки любой страны обозначаются тремя буквами. Пара состоит из двух валют: на первом месте стоит базовая, а на втором – котируемая величина, отображающая стоимость единицы товара.

Основными участниками валютного рынка являются коммерческие банки, инвестиционные фонды, международные компании, центральные банки и индивидуальные трейдеры. Они осуществляют операции на валютном рынке с

целью обеспечения своих потребностей в иностранной валюте для международной торговли, инвестиций или спекуляции.

Трейдер – это человек или организация, занимающиеся торговлей на финансовых рынках с целью получения прибыли на разнице валютных курсов. Трейдеры отслеживают динамику рынка и осуществляют операции купли-продажи валют.

Сделки, которые совершают трейдеры на рынке Forex, бывают двух видов: на покупку и на продажу. Покупка также называется «длинной позицией», поскольку в разрезе стратегического планирования, курсы валют всегда растут. Продажа, соответственно, – «короткая позиция».

Решение о покупке или продаже принимается исходя из торговой стратегии, которой придерживается трейдер. Торговая стратегия – это совокупность инструментов анализа и правил, которых придерживается трейдер в собственной работе на валютном рынке.

На курс валюты влияет множество факторов: политика центрального банка, состояние экономики конкретной страны и мира в целом, действия крупных игроков на валютном рынке, большую роль играют слухи и ожидания среди трейдеров. Для прогнозирования динамики валютного курса применяется как фундаментальный экономический анализ, так и используемый только на биржах способ – так называемый технический анализ.

Фундаментальный анализ – это метод анализа финансовых рынков, который основывается на изучении фундаментальных факторов, влияющих на цену активов. Он предназначен для определения «фундаментальной» стоимости актива, то есть его внутренней ценности, на основе основных экономических, финансовых и других факторов, связанных с активом. Цель фундаментального анализа состоит в том, чтобы определить, является ли актив переоцененным или недооцененным на рынке, и использовать эту информацию для принятия инвестиционных решений. Трейдеры и инвесторы, применяющие фундаментальный анализ, могут использовать его для выбора акций, оценки ценности валют, анализа товарных рынков и т. д.

Технический анализ – это метод анализа финансовых рынков, который основывается на изучении исторических ценовых данных и объемов торговли для прогнозирования будущих движений цен. Он основан на предположении, что ценовая информация и ее движение уже отражают все доступные фундаментальные факторы и тренды рынка.

Цель технического анализа заключается в том, чтобы использовать исторические данные и паттерны для прогнозирования будущих ценовых движений и принятия решений о покупке или продаже активов. Технические трейдеры и инвесторы часто используют технический анализ для определения точек входа и выхода из рынка. Он помогает трейдерам и инвесторам принимать решения о покупке или продаже финансовых инструментов на основе сигналов, полученных из анализа ценовых графиков, индикаторов и паттернов.

1.3 Искусственная нейронная сеть

1.3.1 Формальная модель нейрона

Искусственная нейронная сеть (ИНС) – упрощенная модель биологической нейронной сети, представляющая собой совокупность искусственных нейронов, взаимодействующих между собой [18]. Искусственные нейроны, из которых состоит ИНС, имеют намного более простую структуру: у них есть несколько входов, на которых они принимают различные сигналы, преобразуют их и передают другим нейронам. Другими словами, искусственный нейрон – это функция

$$R^n \to R, \tag{1}$$

которая преобразует несколько входных параметров в один выходной [18].

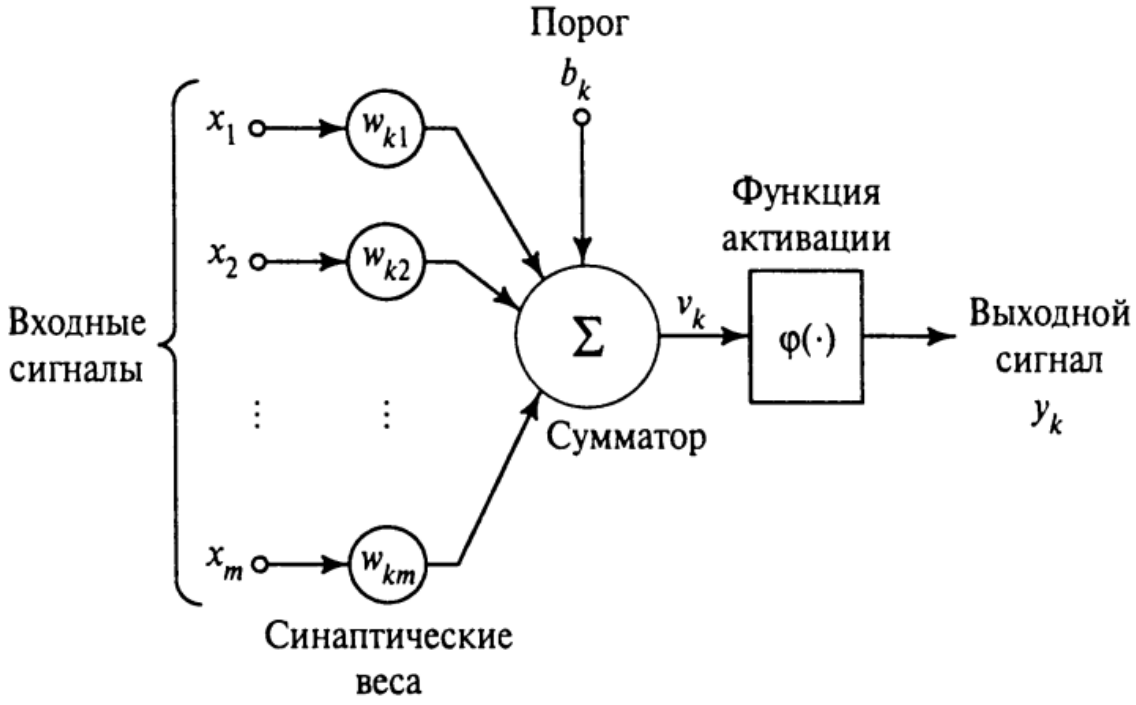

Рисунок 1 – Модель нейрона

На рисунке 1 показана модель нейрона, лежащего в основе ИНС. В этой модели можно выделить три основных элемента:

1) Набор *синапсов* или *связей*, каждый из которых характеризуется своим *весом* или *силой.* В частности, сигнал x_j на входе синапса j, связанного с нейроном k, умножается на вес w_{kj}.

2) *Сумматор* складывает входные сигналы, взвешенные относительно соответствующих синапсов нейрона. Эту операцию можно описать как *линейную комбинацию*.

3) *Функция активации* ограничивает амплитуду выходного сигнала нейрона. Эта функция также называется функцией *сжатия*. Обычно нормализованный диапазон амплитуд выхода нейрона лежит в интервале $[0, 1]$ или $[-1, 1]$.

В модель нейрона, показанную на рисунке 1, включен *пороговый элемент*, который обозначен символом b_k. Эта величина отражает увеличение или уменьшение входного сигнала, подаваемого на функцию активации.

В математическом представлении функционирование нейрона k можно описать следующей парой уравнений:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \,, \tag{2}$$

$$y_k = \varphi(u_k + b_k), \tag{3}$$

где $x_1, x_2, \dots, x_m$ – входные сигналы; $w_{k1}, w_{k2}, \dots, w_{km}$ – синаптические веса нейрона k; u_k – *линейная комбинация входных воздействий*; b_k – порог; φ – *функция активации*; y_k – выходной сигнал нейрона. Использование порога b_k обеспечивает эффект аффинного преобразования выхода линейного сумматора u_k.

Как правило, в большинстве нейронных сетей есть так называемый *входной слой*, который выполняет только одну задачу – распределение входных сигналов остальным нейронам. Нейроны этого слоя не производят никаких вычислений. В остальном нейронные сети делятся на основные категории, представленные ниже.

Однослойная нейронная сеть – сеть, в которой сигналы от входного слоя сразу подаются на выходной слой, который и преобразует сигнал и сразу же выдает ответ [18].

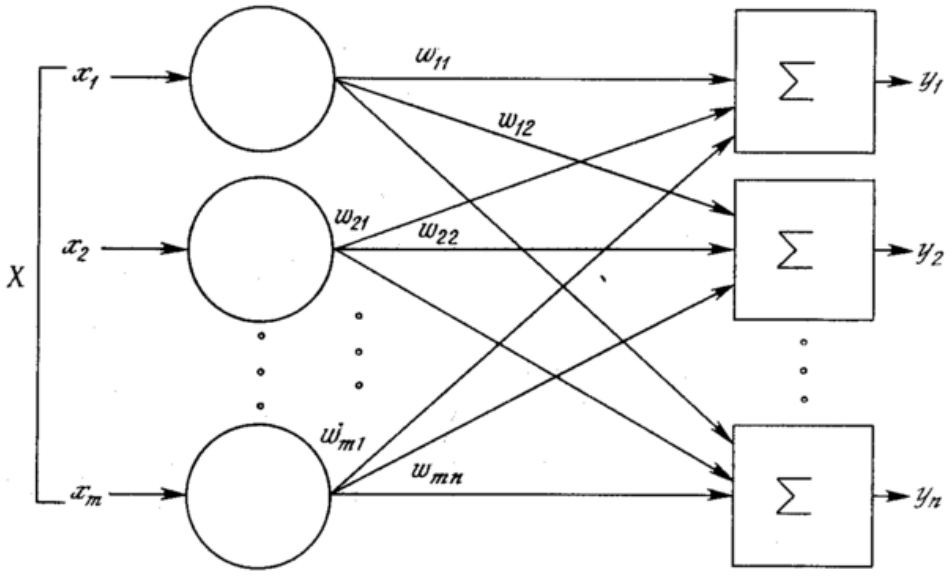

Рисунок 2 – Схема однослойной нейронной сети

Как видно из схемы однослойной нейронной сети, представленной на рисунке 2, сигналы $x_1, x_2, \dots, x_m$ поступают на входной слой, а затем сигналы распределяются на выходной слой обычных нейронов. На каждом ребре от

нейрона входного слоя к нейрону выходного слоя написано число – вес соответствующей связи.

Многослойная нейронная сеть – нейронная сеть, состоящая из входного, выходного и расположенного между ними одного или нескольких скрытых слоев нейронов [18].

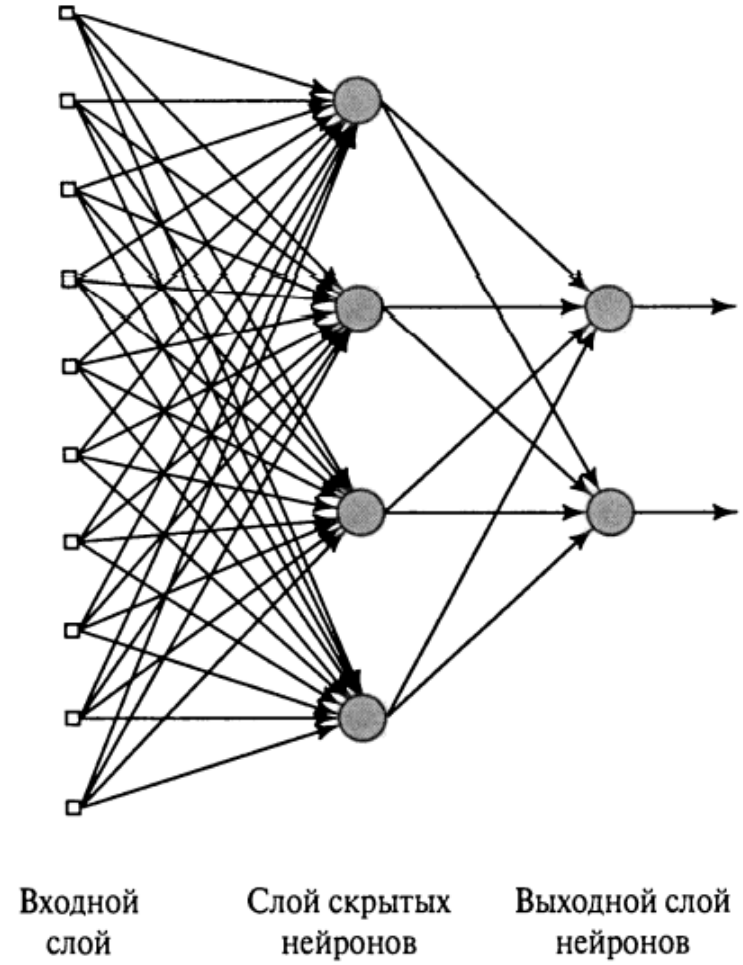

Рисунок 3 – Схема многослойной нейронной сети

Помимо входного и выходного слоев эти нейронные сети содержат промежуточные, скрытые слои. Такие сети обладают гораздо большими возможностями, чем однослойные нейронные сети, однако методы обучения нейронов скрытого слоя были разработаны относительно недавно. Работу скрытых слоев нейронов можно сравнить с работой большого завода. Продукт (выходной сигнал) на заводе собирается по стадиям на станках. После каждого станка получается какой-то промежуточный результат. Скрытые слои тоже преобразуют входные сигналы в некоторые промежуточные результаты [18].

Сети прямого распространения – искусственные нейронные сети, в которых сигнал распространяется строго от входного слоя к выходному. В обратном направлении сигнал не распространяется.

Все сети, описанные выше, являлись сетями прямого распространения, как следует из определения. Такие сети широко используются и вполне успешно решают определенный класс задач: прогнозирование, кластеризация и распознавание [18].

Сети с обратными связями – искусственные нейронные сети, в которых выход нейрона может вновь подаваться на его вход. В более общем случае это означает возможность распространения сигнала от выходов к входам.

В сетях прямого распространения выход сети определяется входным сигналом и весовыми коэффициентами при искусственных нейронах. В сетях с обратными связями выходы нейронов могут возвращаться на входы. Это означает, что выход какого-нибудь нейрона определяется не только его весами и входным сигналом, но еще и предыдущими выходами (так как они снова вернулись на входы) [18].

1.3.2 Обучение нейронной сети

Обучение нейронной сети – это поиск такого набора весовых коэффициентов, при котором входной сигнал после прохода по сети преобразуется в нужный нам выходной. Это определение соответствует и биологическим нейросетям. Наш мозг состоит из огромного количества связанных друг с другом нейросетей, каждая из которых в отдельности состоит из нейронов одного типа (с одинаковой функцией активации). Наш мозг обучается благодаря изменению синапсов – элементов, которые усиливают или ослабляют входной сигнал [18].

Если обучать сеть, используя только один входной сигнал, то сеть просто запомнит правильный ответ, а как только мы подадим немного измененный сигнал, вместо правильного ответа получим бессмыслицу. Мы ждем от сети способности обобщать какие – то признаки и решать задачу на различных

входных данных. Именно с этой целью и создаются обучающие выборки. *Обучающая выборка* – конечный набор входных сигналов, по которым происходит обучение сети.

После обучения сети, то есть, когда сеть выдает корректные результаты для всех входных сигналов из обучающей выборки, ее можно использовать на практике. Однако, прежде чем сразу использовать нейронную сеть, обычно производят оценку качества ее работы на так называемой тестовой выборке. *Контрольная выборка* – конечный набор входных сигналов, по которым происходит оценка качества работы сети.

Обучение нейронной сети можно разделить на два подхода: обучение *с учителем* и обучение *без учителя*. В первом случае веса меняются так, чтобы ответы сети минимально отличались от уже готовых правильных ответов, а во втором случае сеть самостоятельно классифицирует входные сигналы.

1.3.3 Задача обучения с учителем

При обучении с учителем нейронная сеть обучается на размеченном наборе данных и предсказывает ответы, которые используются для оценки точности алгоритма на обучающих данных.

Одним из типов задач обучения с учителем, является задача классификации, которая имеет следующий вид.

Задано множество *объектов* X, множество *допустимых ответов* Y, и существует *целевая функция*

$$y^*: X \to Y, \tag{4}$$

значения которой

$$y_i = y^*(x_i) \tag{5}$$

известны только на конечном множестве объектов $\{x_1, \dots, x_\ell\} \subset \mathrm{X}$. Пары «объект-ответ» (x_i, y_i) называются *прецендентами*. Совокупность пар

$$X^\ell = (x_i, y_i)|_{i=1}^{\ell} \tag{6}$$

называется *обучающей выборкой*.

Задача *обучения по прецендентам* заключается в том, чтобы по выборке X^{ℓ} *восстановить зависимость* y^*, то есть построить *решающую функцию*

$$\alpha: X \to Y, \tag{7}$$

которая приближала бы целевую функцию

$$y^*(x), \tag{8}$$

причем не только на объектах обучающей выборки, но и на всем множестве X. Решающую функцию α называют *алгоритмом*.

Признак f объекта x – это результат измерения некоторой характеристики объекта. Формально признаком называется отображение

$$f: X \to D_f, \tag{9}$$

где D_f – множество допустимых значений признака. В частности, любой алгоритм

$$\alpha: X \to Y \tag{10}$$

также можно рассматривать как признак. Пусть имеется набор признаков $f_1, \dots, f_n$. Вектор $(f_1(x), \dots, f_n(x))$ называют *признаковым описанием* объекта $x \in X$. В дальнейшем мы не будем различать объекты из X и их признаковые описания, полагая

$$X = D_{f_1} \times \dots \times D_{f_n}. \tag{11}$$

Если $Y = \{0, \dots, M\}$, то это задача *классификации* на M непересекающихся классов. В этом случае всё множество объектов X разбивается на несколько классов. Алгоритм $\alpha(x)$ должен давать ответ на вопрос «какому классу принадлежит объект x?» [18].

1.3.4 Задача обучения без учителя

При обучении без учителя модель использует неразмеченные данные, из которых алгоритм самостоятельно пытается извлечь признаки и зависимости.

Задача классификации объектов на основе их сходства друг с другом, когда принадлежность обучающих объектов каким-либо классам не задаётся, называется задачей *кластеризации*.

Задача кластеризации (или обучения без учителя) заключается в следующем. Имеется обучающая выборка

$$X^{\ell} = \{x_1, \dots, x_{\ell}\} \subset \mathrm{X} \tag{12}$$

и функция расстояния между объектами

$$\rho(x, x'). \tag{13}$$

Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике ρ, а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X^{\ell}$ приписывается метка кластера y_i.

Алгоритм кластеризации – это функция

$$\alpha: X \to Y, \tag{14}$$

которая любому объекту $x \in X$ ставит в соответствие метку кластера $y \in Y$. Множество меток Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации.

1.3.5 Метод обратного распространения ошибки

Обучение методом обратного распространения ошибки предполагает два прохода по всем слоям сети: прямого и обратного. В этом режиме алгоритм обрабатывает примеры из обучающей выборки следующим образом [18].

1) *Инициализация*. Предполагая отсутствие априорной информации, генерируются синаптические веса и пороговые значения с помощью датчика равномерно распределенных чисел со средним значением 0.

2) *Предъявление примеров обучения*. В сеть подаются образы из обучающей выборки. Для каждого образа последовательно выполняются прямой и обратный проходы, описанные далее.

3) *Прямой проход*. Пусть пример обучения представлен парой $\big(\boldsymbol{x}(n), \boldsymbol{d}(n)\big)$, $\boldsymbol{x}(n)$ – входной вектор, предъявляемый входному слою сенсорных узлов; $\boldsymbol{d}(n)$ – желаемый отклик, предоставляемый выходному слою нейронов для формирования сигнала ошибки. Вычисляются индуцированные локальные

поля и функциональные сигналы сети, проходя по ней послойно в прямом направлении. Результат работы нейрона j слоя l вычисляется по формуле:

$$v_j^{(l)}(n) = \sum_{i=0}^{m_0} \omega_{ji}^{(l)}(n) y_i^{(l-1)}(n), \tag{15}$$

где $y_i^{(l-1)}(n)$ – выходной (функциональный) сигнал нейрона i, расположенного в предыдущем слое $l-1$, на итерации n; $\omega_{ji}^{(l)}(n)$ – синаптический вес связи нейрона j слоя l с нейроном i слоя $l-1$. Для $i = 0$:

$$y_0^{(l-1)}(n) = +1, \tag{16}$$

$$\omega_{j0}^{(l)}(n) = b_j^l(n) \tag{17}$$

где (17) – порог, применяемый к нейрону j слоя l. Если используются сигмоидальная функция, то выходной сигнал нейрона j слоя l выражается следующим образом:

$$y_j^{(l)}(n) = \varphi_j\left(v_j(n)\right). \tag{18}$$

Если нейрон j находится в первом скрытом слое (т. е. $l = 1$), то

$$y_j^{(0)}(n) = o_j(n). \tag{19}$$

Вычисляется сигнал ошибки:

$$e_j(n) = d_j(n) - o_j(n), \tag{20}$$

где $d_j(n)$ – это j-й элемент вектора желаемого отклика $\mathbf{d}(n)$.

4) *Обратный проход*. Вычисляются локальные градиенты узлов сети по следующей формуле:

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n)\varphi_j'\left(v_j^{(L)}(n)\right), \text{для нейрона } j \text{ выходного слоя } L, \\ \varphi_j'\left(v_j^{(l)}(n)\right)\sum_k \delta_k^{(l+1)}(n)\delta_{kj}^{(l+1)}(n) \text{ для нейрона } j \text{ слоя } l \end{cases} \tag{21}$$

где штрих в функции $\varphi_j'(\cdot)$ обозначает дифференцирование по аргументу. Изменение синаптических весов слоя l сети выполняется в соответствии с обобщенным дельта – правилом

$$\omega_{ji}^{(l)}(n+1) = \omega_{ji}^{(l)}(n) + \alpha\left[\omega_{ji}^{(l)}(n-1)\right] + \eta\partial_j^{(l)}(n) y_j^{(l-1)}(n), \tag{22}$$

где η – параметр скорости обучения; α – постоянная момента.

5) *Итерации*. Последовательно выполняются прямой и обратный проходы (согласно пунктам 3, 4), предъявляя сети все примеры обучения из обучающей выборки, пока не будет достигнут критерий останова.

1.3.6 Подбор структуры сети

Выбор структуры сети, то есть числа слоёв, числа нейронов и числа связей для каждого нейрона, является, пожалуй, наиболее сложной проблемой. Существуют различные стратегии поиска оптимальной структуры сети: постепенное наращивание, построение заведомо слишком сложной сети с последующим упрощением, поочерёдное наращивание и упрощение.

Проблема выбора структуры тесно связана с проблемами *недообучения* и переобучения. Слишком простые сети не способны адекватно моделировать целевые зависимости в реальных задачах. Слишком сложные сети имеют избыточное число свободных параметров, которые в процессе обучения настраиваются не только на восстановление целевой зависимости, но и на воспроизведение шума.

Если в конкретной задаче гипотеза о линейной разделимости классов выглядит правдоподобно, то можно ограничиться однослойным персептроном. Двухслойные сети позволяют представлять извилистые нелинейные границы, и в большинстве случаев этого хватает. Трёхслойными сетями имеет смысл пользоваться для представления сложных многосвязных областей. Чем больше слоёв, тем более богатый класс функций реализует сеть, но тем хуже сходятся градиентные методы, и тем труднее её обучить.

Выбор числа нейронов в скрытом слое производят различными способами, но ни один из них не является лучшим.

1) *Визуальный способ*. Если граница классов слишком сглажена, значит, сеть переупрощена, и необходимо увеличивать число нейронов в скрытом слое. Если граница классов испытывает слишком резкие колебания, на тестовых данных наблюдаются большие выбросы, веса сети принимают большие

по модулю значения, то сеть переусложнена, и скрытый слой следует сократить. Недостаток этого способа в том, что он подходит только для задач с низкой размерностью пространства (небольшим числом признаков).

2) *Оптимизация количества нейронов* по внешнему критерию, например, по критерию средней ошибки на независимой контрольной выборке. Зависимость внешних критериев от параметра сложности обычно имеет характерный оптимум. Недостаток этого способа в том, что приходится много раз заново строить сеть при различных значениях числа нейронов.

3) *Динамическое добавление нейронов*. Сначала сеть обучается при заведомо недостаточной мощности скрытого слоя. Обучение происходит до тех пор, пока ошибка не перестанет убывать. Тогда добавляется один или несколько новых нейронов. Веса новых связей инициализируются небольшими случайными числами, либо добавленные нейроны обучаются по – отдельности как однослойные персептроны. Во втором случае можно рекомендовать обучать новый персептрон на случайной подвыборке, возможно, добавив в неё те объекты, на которых текущая сеть допустила наибольшие ошибки. Веса старых связей не меняются. Затем проводится настройка сети методом обратного распространения.

1.4 Торговые алгоритмы

Алгоритмы торговли на бирже – это компьютерные программы, которые автоматизируют процесс принятия решений о покупке или продаже финансовых инструментов на основе заранее заданных правил и параметров. Эти алгоритмы позволяют трейдерам выполнять торговые операции с высокой скоростью и точностью, что может быть особенно полезно на быстром и динамичном рынке.

Существует множество различных типов алгоритмов торговли, каждый из которых применяется в разных ситуациях и строится на разных стратегиях. Некоторые распространенные алгоритмы включают:

1) *Алгоритмы рыночного ордера*. Эти алгоритмы направлены на мгновенное исполнение ордера по текущей рыночной цене. Они обеспечивают

высокую вероятность исполнения, но не учитывают ценовые изменения, происходящие во время исполнения ордера.

2) *Алгоритмы лимитного ордера*. Эти алгоритмы направлены на исполнение ордера по определенной цене или лучшей. Они позволяют трейдерам контролировать цену исполнения, но могут столкнуться с ограничениями в случае недостатка ликвидности на рынке.

3) *Алгоритмы стоп-лосс и тейк-профит*. Эти алгоритмы устанавливают автоматические уровни стоп-лосса (предельные уровни убытков) и тейк-профита (уровни зафиксированной прибыли) для управления рисками и защиты прибылей.

4) *Алгоритмы арбитража*. Эти алгоритмы ищут возможности для получения прибыли путем эксплуатации ценовых различий на разных рынках или биржах. Они могут быть использованы для быстрого выявления и выполнения выгодных сделок.

5) *Алгоритмы анализа и прогнозирования*: Эти алгоритмы используют математические модели и статистические методы для анализа рыночных данных и прогнозирования будущих ценовых движений.

Это лишь некоторые примеры алгоритмов торговли на бирже. Каждый алгоритм может быть настроен и адаптирован в соответствии с индивидуальными требованиями трейдера или фондовой компании. Важно отметить, что использование алгоритмов торговли требует тщательного тестирования, мониторинга и постоянного обновления, чтобы учесть изменения на рынке и предотвратить нежелательные последствия, связанные с автоматическим исполнением сделок.

Далее в данной работе будут подробнее рассмотрены алгоритмы стоп-лосс и тейк-профит на основе подсистемы формирования сигналов, описанной в работе [1].

1.5 Торговые роботы

Торговые роботы, также известные как эксперты или советники (Expert Advisors), являются программными приложениями, которые автоматически выполняют торговые операции на финансовых рынках без необходимости вмешательства человека. Они используют заранее заданные правила и стратегии для принятия решений о покупке, продаже или закрытии позиций.

Преимущества использования торговых роботов перед живым мониторингом включают:

1) *Автоматизация*. Торговые роботы позволяют автоматизировать процесс торговли, что освобождает трейдера от необходимости постоянного мониторинга рынка. Роботы могут работать 24 часа в сутки, 7 дней в неделю, без усталости и эмоционального влияния.

2) *Скорость и точность*. Торговые роботы могут реагировать на рыночные события и изменения цен мгновенно и с высокой точностью. Они способны выполнить торговые операции с высокой скоростью, что особенно важно в быстром и динамичном рынке.

3) *Систематичность*. Торговые роботы работают на основе заранее заданных правил и стратегий, что обеспечивает систематичность в принятии решений о торговле. Это исключает эмоциональные факторы и субъективность, которые могут повлиять на принятие решений человеком.

4) *Возможность тестирования*. Торговые роботы позволяют проводить тестирование и оптимизацию стратегий на исторических данных. Это позволяет трейдерам оценить эффективность своей стратегии и внести необходимые корректировки перед применением робота в реальной торговле.

Однако важно понимать, что использование торговых роботов имеет и свои ограничения и риски. Роботы могут быть настроены неправильно или быть подвержены техническим сбоям, что может привести к нежелательным результатам. Кроме того, рынки подвержены нестабильности и внеплановым событиям, которые могут вызвать изменения, которые не были заложены в алгоритм робота. Поэтому важно тщательно разрабатывать и тестировать

торговые роботы перед их применением в реальной торговле. Трейдеры должны учитывать факторы риска, устанавливать правильные параметры и надлежащим образом мониторить и обновлять роботов.

Также стоит отметить, что успешность торговых роботов зависит от выбора и разработки эффективной торговой стратегии. Стратегия должна быть основана на анализе рынка и иметь преимущества перед другими участниками рынка. Роботы не могут гарантировать прибыль и могут быть подвержены риску потери капитала, особенно в нестабильных рыночных условиях.

1.6 Обзор существующих решений

При использовании торговых алгоритмов, для автоматического трейдинга, трейдеру необходимо решить следующие проблемы:

1) поиск и обработка данных;
2) подбор и оптимизация алгоритма для торговли на целевой валютной паре;
3) проверка качества работы алгоритма;

Разрабатываемая программная система должна обладать следующим функционалом:

1) автоматическая загрузка данных;
2) генерация обучающих выборок;
3) отображение торговых сигналов;
4) ведение автоматической торговли;
5) обучение моделей для формирования торговых сигналов;
6) тестирование моделей на исторических данных;
7) сбор статистики на основе проведенного тестирования;

В рамках дипломной работы были рассмотрены следующие существующие программные решения.

1.6.1 AmiBroker

AmiBroker – платформа для технического анализа фондового рынка. Функционал программы позволяет писать и тестировать алгоритмические торговые системы. Доступна только десктопная версия AmiBroker. Ее можно использовать на ОС Window и Linux [21].

Функционал терминала AmiBroker сосредоточен на техническом анализе и написании алгоритмических стратегий. Он позволяет получать котировки практически со всех мировых бирж. Бесплатно приложение предоставляет только дневные биржевые данные. Котировки в реальном времени предоставляются только через платные сервисы [21].

Чтобы скачать AmiBroker, необходимо приобрести лицензионную версию терминала.

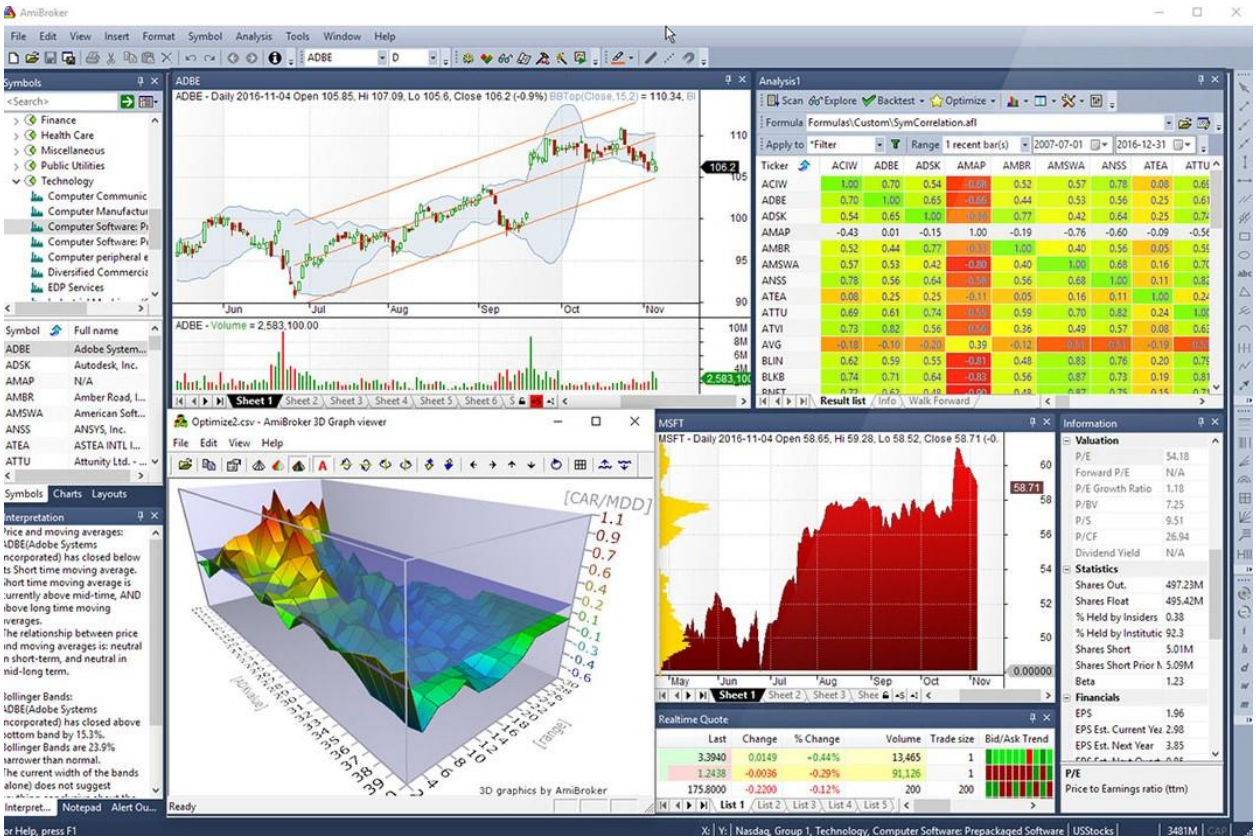

Рисунок 4 – Интерфейс AmiBroker

Таким образом, к функциональным особенностям AmiBroker можно отнести:

1) автоматическая загрузка данных;
2) отображение торговых сигналов;

3) ведение автоматической торговли;
4) обучение моделей для формирования торговых сигналов;
5) тестирование моделей на исторических данных.

1.6.2 NeuroShell Trader

NeuroShell Trader – это семейство продуктов, разработанное специально для трейдеров и призванное помочь им в принятии решений при торговле. В нем реализованы технологии искусственного интеллекта, позволяющие прогнозировать финансовые временные ряды на основе технического анализа, строить и оптимизировать торговые стратегии [22].

Будучи специализированным инструментом для трейдеров, NeuroShell Trader имеет интуитивно понятный графический интерфейс, возможности для импорта данных и большую библиотеку индикаторов. NeuroShell Trader позволяет экспортировать временные ряды в текстовые файлы или в буфер обмена для последующего использования в других программах [22].

NeuroShell Trader позволяет отображать биржевые данные на рабочих листах в виде графиков, диаграмм, японских свечей.

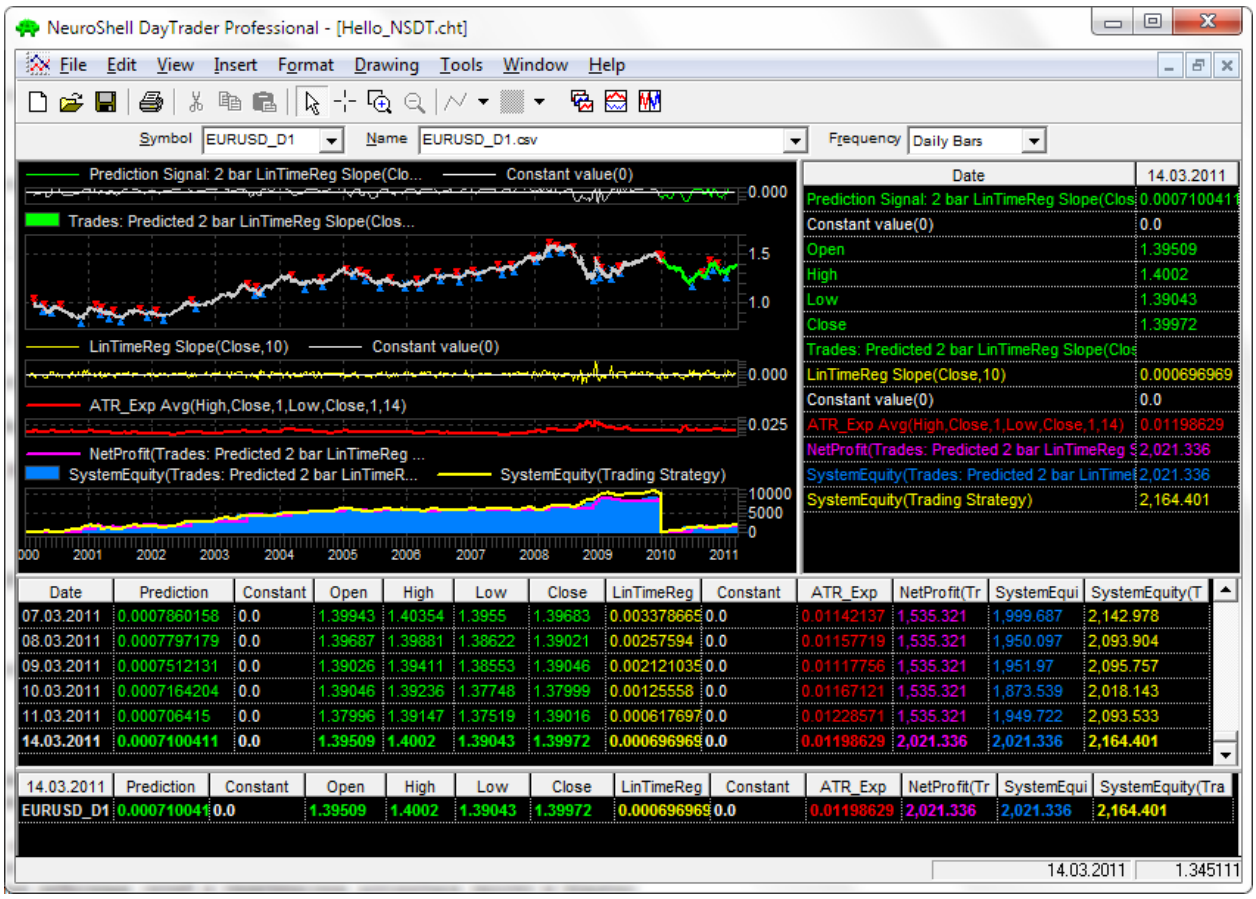

Рисунок 5 – Интерфейс NeuroShell Trader

Таким образом, к функциональным особенностям NeuroShell Trader можно отнести:

1) автоматическая загрузка данных;
2) отображение торговых сигналов;
3) обучение моделей для формирования торговых сигналов;
4) тестирование моделей на исторических данных.

1.6.3 QuikFisher

Универсальный торговый робот QuikFisher – это торговая система, в основе которой лежит технический анализ. Главными достоинствами всех торговых систем является отсутствие человеческого фактора, то есть эмоций, которые часто оказывают отрицательное влияние на работу трейдера.

Важным свойством QuikFisher является гибкая настройка – 20 переменных, с помощью которых можно настраивать работу QuikFisher с учетом

персональных рисков, торговых предпочтений и в соответствии с изменяющейся рыночной ситуацией.

QuikFisher – универсальный робот. За счет большого количества инструментов и гибкости в настройках QuikFisher является весьма продуктивным, он может эффективно и прибыльно работать, как с высоколиквидными, так и низко ликвидными инструментами, как на растущем, так и на падающем рынках.

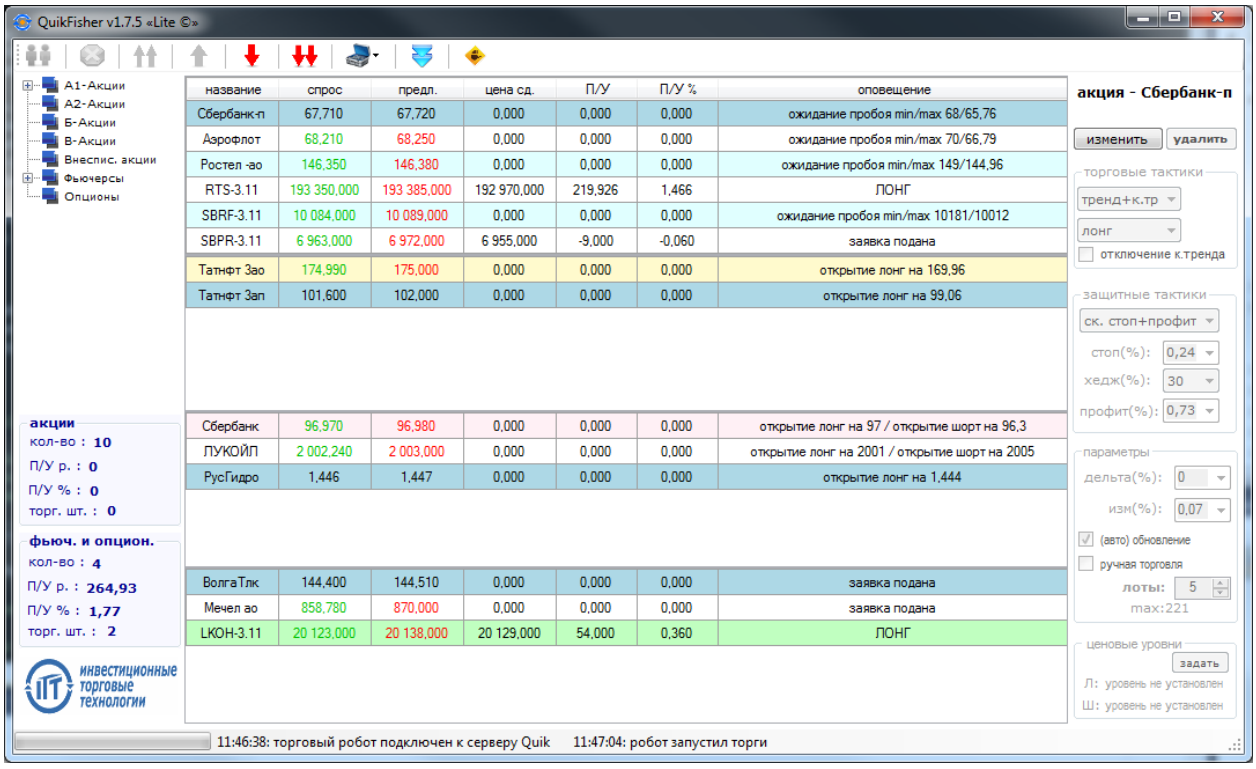

Рисунок 6 – Интерфейс QuikFisher

Таким образом, к функциональным особенностям QuikFisher можно отнести:

1) ведение автоматической торговли.

1.6.4 QuikHunter

QuikHunter является импульсным торговым роботом, созданным для технического анализа рынка. Он работает с терминалом Quik. Особенность торгового алгоритма QuikHunter в том, что он ловит импульсные сильные ценовые движения, случающиеся десятки раз в течение торговой сессии. Обычно сильные импульсные движения происходят после снижения торговой рыночной

активности, когда цена преодолела важные уровни поддержки или сопротивления, или же после выхода экономически важной статистики, либо после открытия зарубежных рынков и в других подобных случаях.

Как правило, поймать сильное импульсное движение вручную и войти в рынок выгодно является сложной задачей, с которой робот справляется великолепно. Обычно трейдер видит импульсное движение и не успевает открыть позиции вовремя, поскольку рынок от него «убегает», а QuikHunter может за доли секунды выгодно открыть позицию.

Крайне тяжело отслеживать момент, когда именно произойдет импульсное движение. Чтобы не упустить момент входа, приходится постоянно мониторить цену и графики: стоит отвлечься на пару минут, и рынок «уходит».

Суть работы торгового робота QuikHunter в постоянном отслеживании цены и ее изменений, и при появлении важного или сильного импульса робот моментально входит в позицию. Биржевой игрок может фильтровать импульсы самостоятельно, выбирая наиболее важные и сильные. Имея собственную торговую стратегию, трейдер может настроить робота таким образом, чтобы QuikHunter ловил длинные либо короткие импульсы, все зависит от потребностей конкретного момента времени.

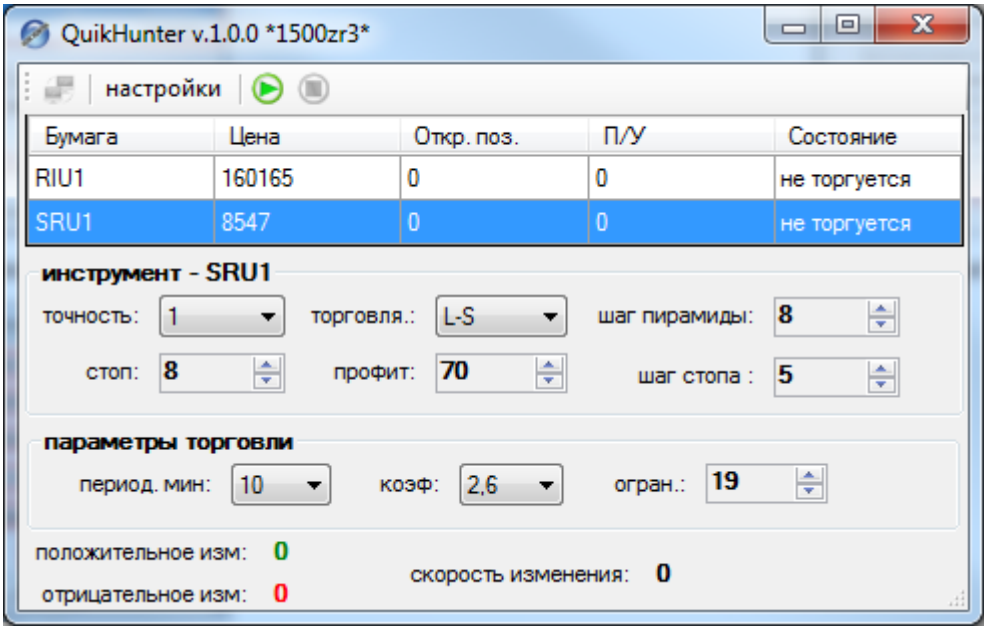

Рисунок 7 – Интерфейс QuikHunter

Таким образом, к функциональным особенностям QuikHunter можно отнести:

1) ведение автоматической торговли.

1.6.5 SuperADX

Торговый робот SuperADX – это торговая система, предназначенная для автоматической торговли. В SuperADX есть возможность использовать робота как советника, в этом варианте робот будет лишь прогнозировать, но не вести торговлю. Следует отметить, что торговый робот работает только в связке с торговым терминалом Quik.

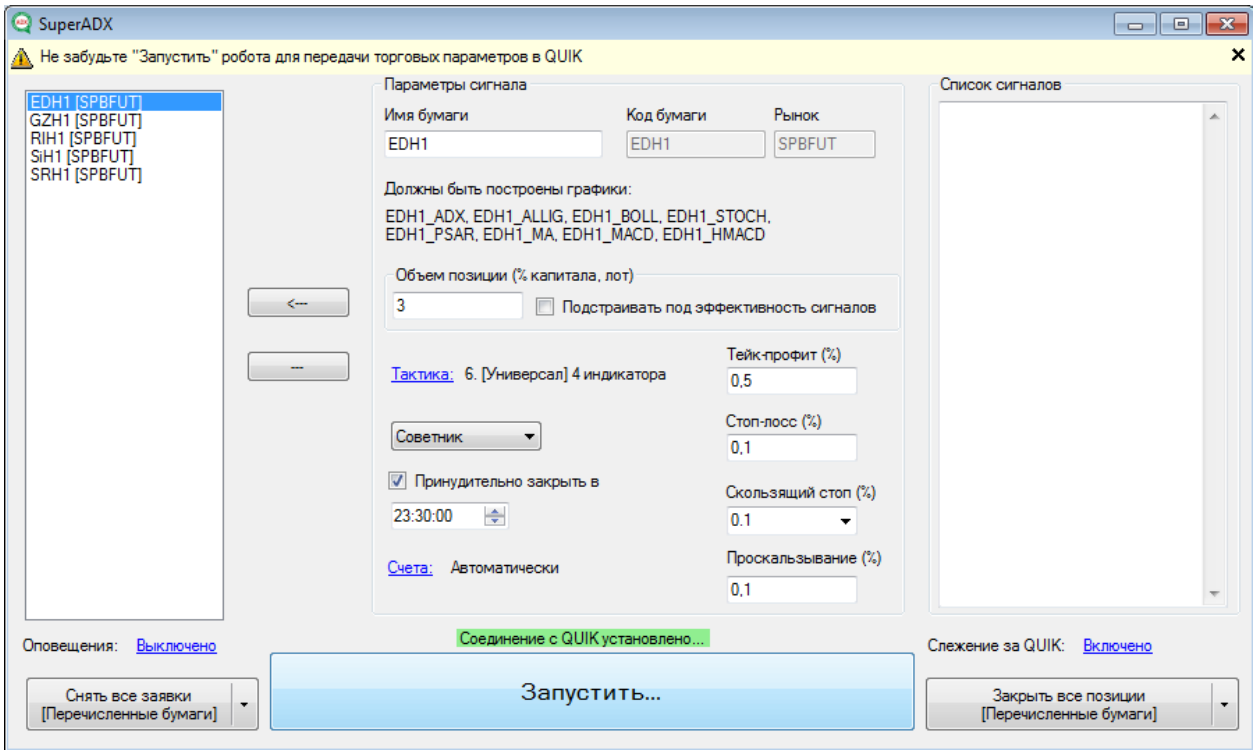

Рисунок 8 – Интерфейс SuperADX

Таким образом, к функциональным особенностям SuperADX можно отнести:

1) отображение торговых сигналов;
2) обучение моделей для формирования торговых сигналов;
3) ведение автоматической торговли.

1.6.6 Заключение

В рамках этой работы были рассмотрены пять программных систем для автоматического трейдинга. Их краткое описание выражено в таблице 1.

Т а б л и ц а 1 – Сравнение функциональных особенностей

	AmiBroker	NeuroShell	QuikFisher	QuikHunter	SuperADX
Автоматическая загрузка данных	+	+	−	−	−
Генерация обучающих выборок	−	−	−	−	−
Ведение автоматической торговли	+	−	+	+	+
Отображение торговых сигналов	+	+	−	−	+
Обучение моделей для формирования торговых сигналов	+	+	−	−	+
Тестирование моделей на исторических данных	+	+	−	−	−
Сбор статистики на основе проведенного тестирования	−	−	−	−	−

Проведенный анализ показал, что существующие программные средства реализуют только часть требований, выдвинутых выше. В частности, ни одно из рассмотренных программных решений не реализуют генерацию обучающих выборок и сбор статистики на основе проведенного тестирования. Кроме того, абсолютно все из рассмотренных программных средств не дают доступа к просмотру исходного кода, что не позволяет считать их безопасными.

Таким образом, разработка системы для трейдинга на основе нейронных сетей остаётся актуальной.

1.7 Вывод по главе

Нейронные сети обладают способностью эффективно обрабатывать большие объемы данных. Это особенно полезно на финансовых рынках, где существует огромное количество исторических данных, новостей и другой информации, которые могут влиять на цены активов. Нейронная сеть способна анализировать эти данные и находить сложные и скрытые закономерности, недоступные для человеческого анализа, которые могут быть полезны для прогнозирования рыночных трендов и принятия решений о торговле.

2 Анализ предметной области

Рассматриваемая профессиональная деятельность в данной работе – это разработка программной системы для трейдинга на основе нейронных сетей. Подробнее о разработке и обучении нейронных сетей в рамках данной предметной области можно прочитать в работе [1].

В данной предметной области можно выделить следующие задачи:

1) разработка алгоритмов для формирования обучающих выборок для нейронных сетей:
   a) выделение точек разворота тренда;
   b) формирование одномерных данных по предыстории;
   c) формирование двумерных данных по предыстории;
2) разработка и обучение нейронных сетей, решение проблемы нехватки данных при помощи трансферного обучения [1];
3) разработка модели автоматической торговой системы с использованием нейронной сети.

2.1 Анализ профессиональной деятельности

Предметная область, рассматриваемая в данной работе – международный валютный рынок. Профессионалами в этой предметной области являются трейдеры, главная цель которых – заработать на разнице валютных курсов. Трейдер в реальном времени анализирует валютную пару, представленную в виде временного ряда, и принимает решение о заключении сделки. Для этого необходимо классифицировать элемент временного ряда, соответствующий настоящему моменту времени, а именно определить изменится ли направление движения цены после данного элемента или нет. От класса элемента зависит, будет ли совершена сделка и если будет, то принимается решение о покупке или продаже.

Таким образом трейдером решаются следующие задачи:

1) классификация временного ряда валютной пары;

2) принятие решения о заключении сделки купли-продажи;

3) ведение статистики совершенных сделок.

Задачу классификации временного ряда можно решать двумя основными способами:

1) на основе фундаментального анализа;

2) на основе технического анализа.

Фундаментальный анализ – это оценка множества экономических факторов, существенно влияющих на курс валюты.

Технический анализ – это способ оценки ситуации на финансовом рынке для принятия торговых решений, основанный только на выявлении статистических закономерностей движения цен. В техническом анализе трейдеры пытаются определить, когда купить или продать биржевой актив, наблюдая только за движением графика цен.

В данной работе будут рассмотрены методы генерации данных, основанные на техническом анализе.

2.2 Алгоритмы формирования обучающих выборок

2.2.1 Выделение точек разворота тренда

Имеется временной ряд изменений курса валюты. Временным рядом называется последовательность

$$P = \{p_t\} \tag{23}$$

значений курса обмена одной валюты к другой в моменты времени $t \in T$ соответственно, упорядоченная в хронологическом порядке, то есть в порядке возрастания временного параметра.

Введём понятие тренда. В математике трендом принято называть основную тенденцию изменения временного ряда. На бирже, трендом называется направленное движение ценового графика в одну из сторон (вверх или вниз). То есть, тренд в трейдинге – это ситуация, когда цена определенное время возрастает или снижается.

Необходимо сформировать подпоследовательность

$$P_k = \{p_{t_k}\} \tag{24}$$

последовательности P, состоящую только из элементов, являющихся точками разворота тренда, причем

$$p_i, p_{i+1} \in P_k \Longrightarrow |p_i - p_{i+1}| \geq delta, i \in T, \tag{25}$$

где

$$delta = \frac{\max(\{p_t\}) - \min(\{p_t\})}{100} h, h \in \mathbb{N}. \tag{26}$$

Под точкой разворота тренда понимают элемент временного ряда, в котором происходит изменение общего направления движения цены с восходящего на нисходящее или наоборот.

Дополним подпоследовательность $\{p_{t_k}\}$ элементами, случайно взятыми между точками разворота тренда, назовём их нейтральными.

Для визуализации работы алгоритма был построен график на рисунке 9. Здесь красным и зеленым цветом обозначены точки разворота тренда с нисходящего на восходящий и с восходящего на нисходящий соответственно. Так же оранжевым цветом обозначены нейтральные точки. На графике представлена часть временного ряда, примерно соответствующая пяти дням с промежутками 15 минут.

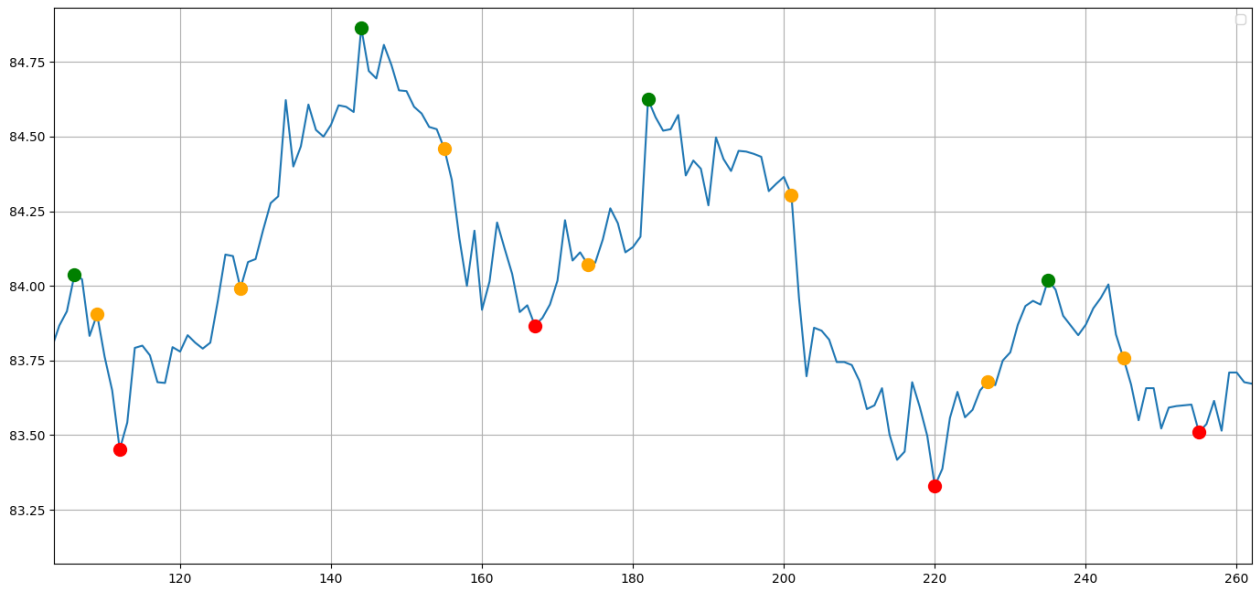

Рисунок 9 – Точки разворота тренда для валютной пары USD/RUB

Полученные классы точек можно трактовать как: «BUY», «SELL» и «HOLD». Их будем использовать в качестве меток классов для формирования

обучающей выборки. Далее рассмотрим алгоритмы формирования признакового описания для каждого класса.

2.2.2 Формирование одномерных данных по предыстории

В случае технического анализа каждому элементу p_i получившейся подпоследовательности P_k поставим в соответствие вектор из n предыдущих значений временного ряда P:

$$x_i = (p_{i-n+1} \dots p_i). \quad (27)$$

Каждый элемент подпоследовательности P_k кодируется следующим образом:

1) если p_i является точкой разворота тренда, в которой происходит изменение направления с восходящего на нисходящее, то p_i представим в виде $(x_i, (0,0,1))$;

2) если p_i является точкой разворота тренда, в которой происходит изменение направления с нисходящего на восходящее, то p_i представим в виде $(x_i, (0,1,0))$;

3) если p_i не является точкой разворота тренда, то p_i представим в виде $(x_i, (1,0,0))$.

В случае фундаментального анализа, помимо последовательности P значений курса, имеется набор из m временных рядов фундаментальных факторов:

$$\{F_j = f_t^j\}, j = 1 \dots m. \quad (28)$$

Каждому элементу p_i подпоследовательности P_k поставим в соответствие матрицу из m строк, где каждая строка – это вектор из n предыдущих значений временного ряда F_j:

$$x_i^j = \left(f_{i-n+1}^j \dots f_i^j\right). \quad (29)$$

Таким образом, для технического анализа имеется множество пар

$$DF = \{(x_i, y_i)\}, \quad (30)$$

где $x_i \in \mathbb{R}^n$ – вектор признаков элемента, а

$$y_i \in (1,0,0), (0,1,0), (0,0,1) \quad (31)$$

– класс элемента соответственно. Для фундаментального анализа:

$$DF = \{(X_i, y_i)\}, \quad (32)$$

где $X_i \in \mathbb{R}^{m \times n}$ – матрица признаков элемента.

2.2.3 Алгоритм MLP

Алгоритм был разработан в рамках подготовки данной работы. Здесь используется модель нейросетевого классификатора «многослойный персептрон» (Multi-Layer Perceptron Classifier). Модель нейронной сети обучается на одномерной выборке данных, полученной первым способом. Данный алгоритм состоит из четырёх основных этапов:

1) из исходного временного ряда выделяются точки разворота тренда, а также нейтральные точки;

2) каждой точке ставится в соответствие некоторое количество предыдущих значений ряда;

3) на полученных размеченных данных обучается многослойный персептрон, выходы которого представлены в виде вектора вероятностей принадлежности к классу;

4) для каждой точки разворота формируется по три матрицы: каждый элемент матрицы представляет собой ответ обученной нейронной сети на отрезках предыдущих значений ряда.

На рисунке 10 вы можете видеть пример построения матриц с использованием алгоритма MLP.

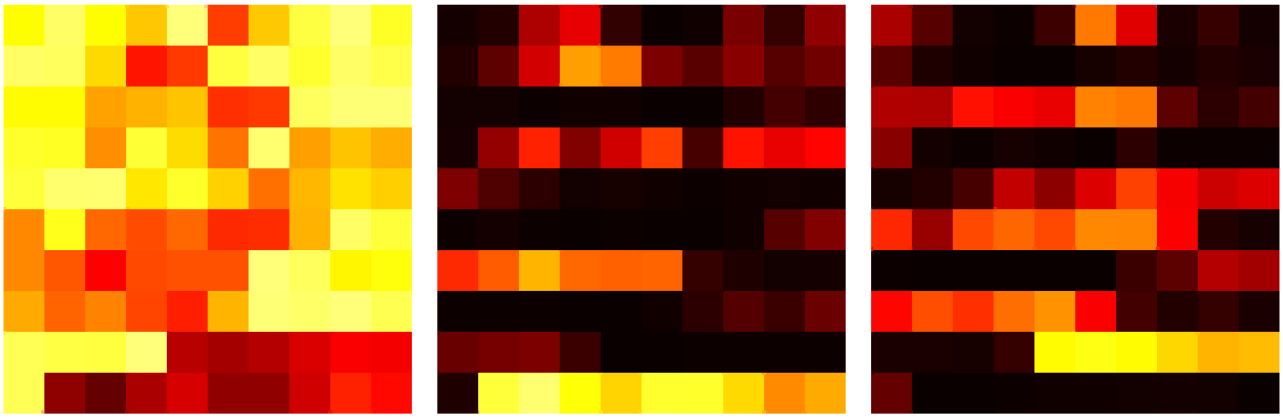

Рисунок 10 – Тепловая диаграмма MLP матриц

Далее полученные матрицы используются в качестве признакового описания разворотной точки в сверочной нейронной сети (CNN). Таким образом для создания двумерного представления признакового описания размерностью $n \times n$ потребуется предыстория n^2 значений временного ряда.

2.2.4 Алгоритм GAF

Gramian Angular Fields (Грамиановское угловое поле) [26] – это метод преобразования временных рядов в двумерные изображения, который использует Грамиановскую матрицу углов (Gramian Angular Matrix). Этот метод позволяет изучать структуры и зависимости во временных рядах с помощью анализа их угловых особенностей.

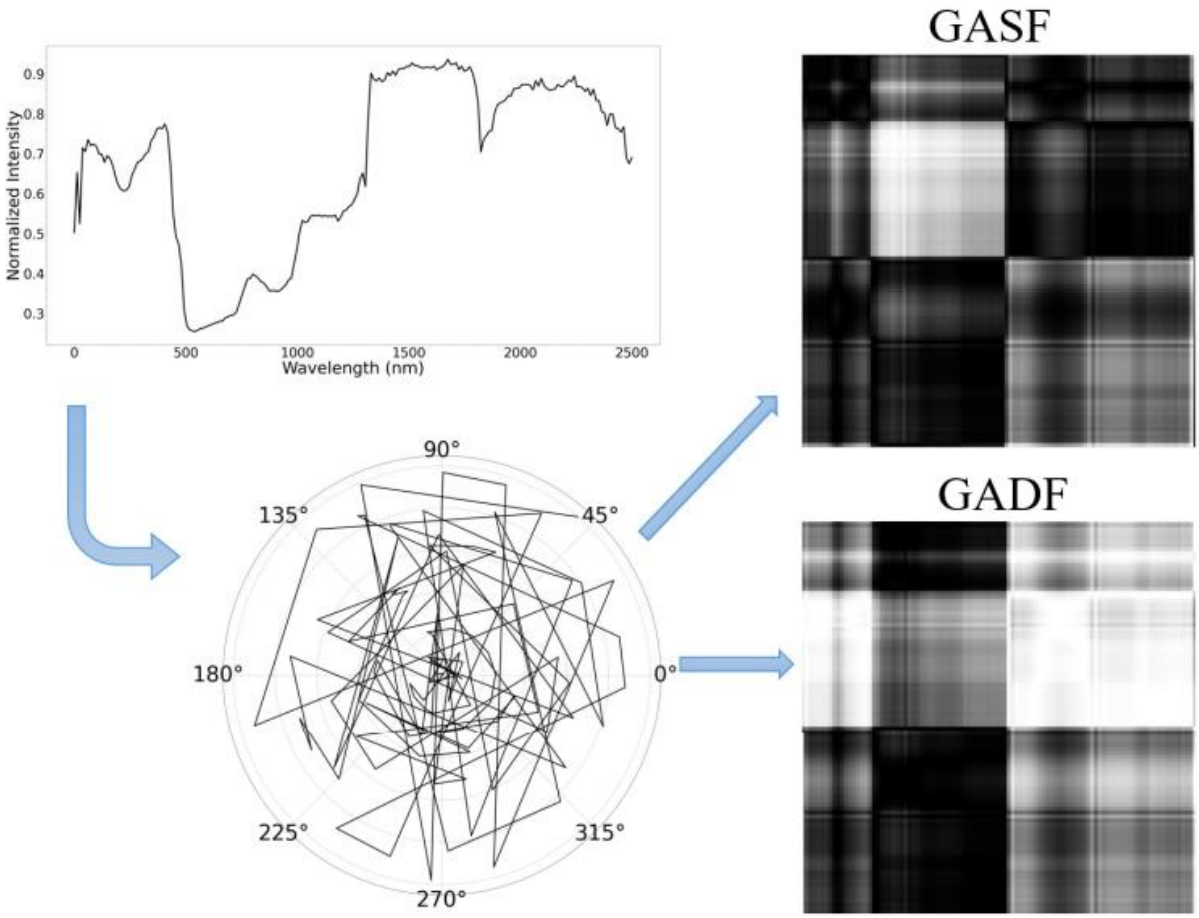

Рисунок 11 – Схема работы алгоритма GAF

С помощью данного метода временные ряды представляются в полярной системе координат, после чего создаётся матрица Грамиана, каждый элемент которой является косинусом суммы углов векторов. Полученная матрица в результате преобразуется в двумерное изображение.

Процесс создания Gramian Angular Fields включает следующие шаги:

1) Нормализация временного ряда. Первоначально значения временного ряда $X = \{x_1, x_2, \ldots, x_n\}$ нормализуются в интервале $[-1, 1]$, используя следующую формулу:

$$\widehat{x_i} = \frac{\left(x_i - max(X)\right) + \left(x_i - min(X)\right)}{max(X) - min(X)}. \quad (33)$$

2) Затем нормализованные значения ряда преобразуются в полярную систему координат по следующим формулам:

$$\begin{cases} \varphi_i = arccos(\widehat{x_i}) \\ r_i = \dfrac{t_i}{N} \end{cases}, \quad (34)$$

где t_i – это индекс текущего элемента ряда, а N – коэффициент для регуляризации диапазона полярной системы координат.

По нормализованным значениям строим GAF матрицы, тригонометрическую сумму (разницу) между каждой точкой для определения временной корреляции в различных временных интервалах:

$$GASF = \begin{bmatrix} cos(\varphi_1 + \varphi_1) & \cdots & cos(\varphi_1 + \varphi_n) \\ \vdots & \ddots & \vdots \\ cos(\varphi_n + \varphi_1) & \cdots & cos(\varphi_n + \varphi_n) \end{bmatrix} \quad (35)$$

$$GADF = \begin{bmatrix} sin(\varphi_1 - \varphi_1) & \cdots & sin(\varphi_1 - \varphi_n) \\ \vdots & \ddots & \vdots \\ sin(\varphi_n - \varphi_1) & \cdots & sin(\varphi_n - \varphi_n) \end{bmatrix} \quad (36)$$

Наглядный процесс создания GAF матриц изображён на рисунке 11. Представление временных рядов через GAF матрицы имеет несколько преимуществ. Во-первых, они позволяют сохранить временную зависимость. По главной диагонали мы можем восстановить временной ряд на основе высокоуровневых характеристик, изученных глубокой нейронной сетью. Во-вторых, можно сократить размер GAF матрицы. Так как размер матрицы Грамиана составляет $n \times n$, при длине временного ряда n, то, чтобы уменьшить размер GAF матрицы обычно применяется метод уменьшения размерности ряда Piecewise Aggregation Approximation (PAA).

Piecewise Aggregation Approximation аппроксимирует временной ряд X длины n в вектор $\bar{X}$ произвольной длины $M \leq n$, каждый элемент которого вычисляется по следующей формуле:

$$\bar{x}_i = \frac{M}{n} \sum_{j=n/M(i-1)+1}^{(n/M)i} x_j. \quad (37)$$

Это означает, что для уменьшения размерности с n до M мы сначала делим исходный временной ряд на M одинаковых по размеру частей, а затем вычисляем средние значения для каждой части. Последовательность, собранная из средних значений, является PAA-аппроксимацией (т. е. преобразованием) исходного временного ряда.

На рисунке 12 вы можете видеть пример построения GAF матриц с использованием реальных данных временного ряда валютной пары USD/RUB с предысторией 25 значений (получаются картинки 25×25 пикселей).

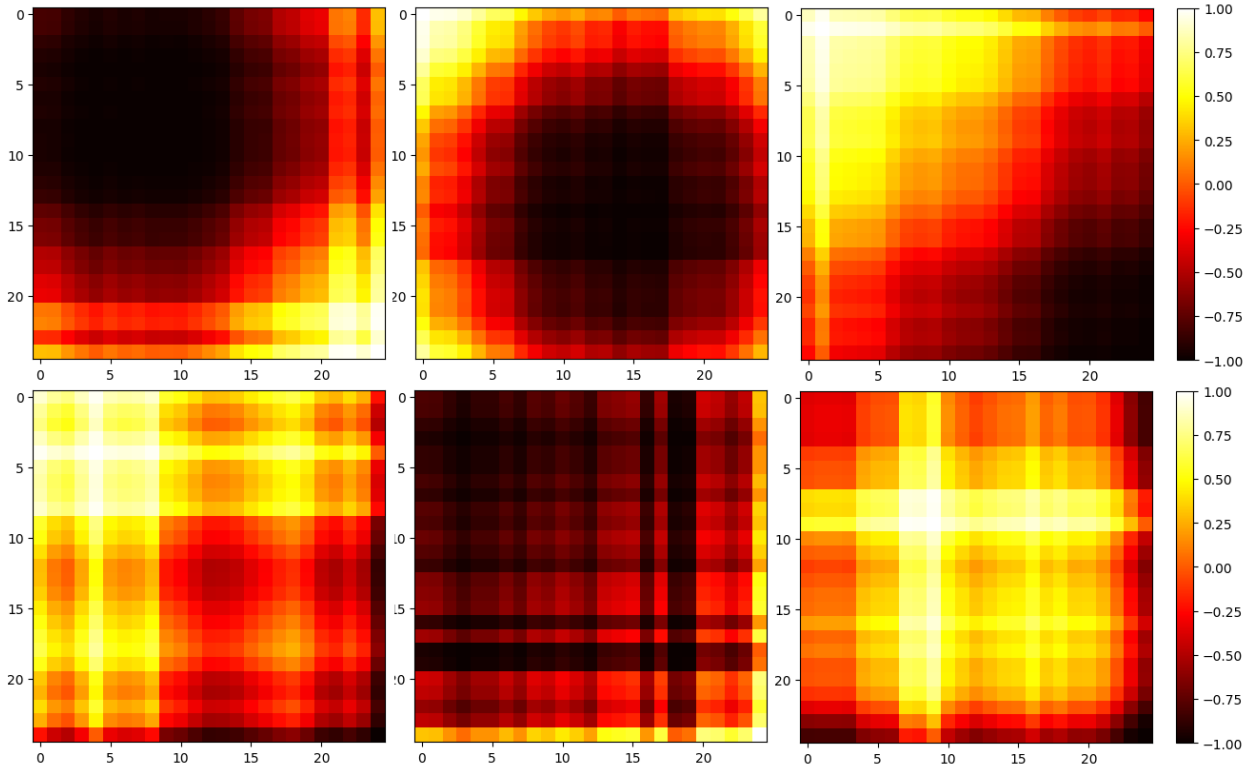

Рисунок 12 – Тепловая диаграмма GAF матриц

2.3 Модель торговой системы

Торговая система – модуль ответственный за принятие решения о заключении сделки купли-продажи. Пусть в момент времени t имеется класс y_t элемента временного ряда p_t, где y_t имеет вид (y_{t1}, y_{t2}, y_{t3}) – сигналы выходных нейронов сети, y_t интерпретируется следующим образом:

1) если $y_{t1} > y_{t2}$ и $y_{t1} > y_{t3}$, то p_t относится к нейтральному классу, в этом случае положим $y_t = 0$;

2) если $y_{t2} > y_{t1}$ и $y_{t2} > y_{t3}$, то p_t является точкой разворота тренда, в которой происходит изменение направления с нисходящего на восходящее, в этом случае положим $y_t = 1$;

3) если $y_{t3} > y_{t1}$ и $y_{t3} > y_{t2}$, то p_t является точкой разворота тренда, в которой происходит изменение направления с восходящего на нисходящее, в этом случае положим $y_t = 2$.

Введем следующие переменные: buy_size – объем сделки, $base\_currency$ – начальный капитал, выраженный в базовой валюте, $quote\_currency$ – начальный капитал, выраженный в целевой валюте.

На основании класса торговая система принимает решение по открытию позиции в размере buy_size базовой валюты. Таким образом если:

1) класс элемента y_t = 0, то позиция не открывается;

2) класс элемента y_t = 1, то торговая система совершает покупку, т. е. открывает короткую позицию;

3) класс элемента y_t = 2, то торговая система совершает продажу, т. е. открывает длинную позицию.

После совершения сделок имеется последовательность открытых позиций

$$\{(\mathrm{c}_t, t)\} \tag{38}$$

где c_t – цена валюты при открытии соответствующей позиции, t – момент времени открытия позиции.

Для описания принятия решения о закрытии позиции введём следующие константы: $take\_profit$, $stop\_loss$.

Каждый элемент c_t сравнивается с элементом p_i в настоящий момент времени. В зависимости от характера позиции принимаются следующие решения:

1) если позиция длинная и $p_i - c_i > take\_profit$, то позиция закрывается, $base\_currency \leftarrow base\_currency - buy\_size$, $quote\_currency \leftarrow quote\_currency + buy\_size \cdot p_i$;

2) если позиция короткая $\mathrm{c}_i - p_i > take\_profit$, то позиция закрывается, $base\_currency \leftarrow base\_currency + buy\_size$, $quote\_currency \leftarrow quote\_currency - buy\_size \cdot p_i$;

3) если позиция длинная и $c_i - p_i > stop\_loss$, то позиция закрывается, $base\_currency \leftarrow base\_currency - buy\_size$, $quote\_currency \leftarrow quote\_currency + buy\_size \cdot p_i$;

4) если позиция короткая и $p_i - c_i > stop\_loss$, то позиция закрывается, $base\_currency \leftarrow base\_currency + buy\_size$, $quote\_currency \leftarrow quote\_currency - buy\_size \cdot p_i$.

3 Разработка проекта системы

3.1 Контекстная диаграмма

Контекстная диаграмма графически иллюстрирует границу и связи разрабатываемой системы со всем остальным миром. Она определяет оконечные элементы, расположенные вне системы, которые определенным образом взаимодействуют с ней, а также данные, элементы управления и материальные потоки, протекающие между оконечными элементами и системой. На рисунке 13 представлена контекстная диаграмма разрабатываемой программной системы для трейдинга на основе нейронных сетей.

Рисунок 13 – Контекстная диаграмма программной системы для трейдинга на основе нейронных сетей

3.2 Архитектурно-контекстная диаграмма

Архитектурно-контекстная диаграмма – это диаграмма, которая определяет основные подсистемы, обеспечивающие все функции разрабатываемой программной системы, а также данные и интерфейсы управления между ними на высоком уровне абстракции. На рисунке 14 представлена архитектурно – контекстная диаграмма разрабатываемой программной системы для трейдинга на основе нейронных сетей.

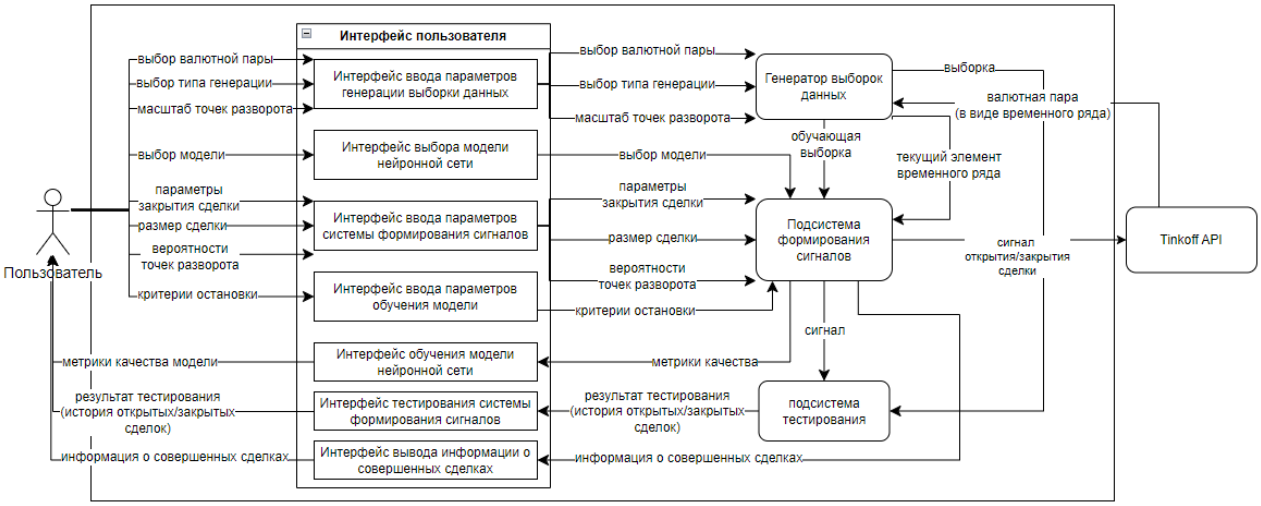

Рисунок 14 – Архитектурно-контекстная диаграмма программной системы для трейдинга на основе нейронных сетей

Программная система для трейдинга на основе нейронных сетей включает следующие компоненты:

1) генератор выборок данных;
2) система формирования сигналов;
3) система тестирования;
4) интерфейс пользователя.

Данная выпускная квалификационная работа заключается в разработке подсистем генерации выборок данных и пользовательского интерфейса.

3.3 Диаграмма потоков данных

Диаграмма потоков данных – основной инструмент структурного анализа. Она позволяет определять процессы преобразования системы, совокупность

(хранилище) данных или материалов, которыми система управляет, и потоки данных или материалов между процессами, хранилищами и внешним миром. Диаграммы потоков данных могут представлять системы на самых разных уровнях абстракции: диаграммы высокого уровня предоставляют целостный, панорамный вид компонентов данных и обработки в многоэтапном процессе, дополняющем точное, детальное представление, приведенное в функциональных требованиях.

На рисунке 15 представлен уровень 0 диаграммы потоков данных программной системы для трейдинга на основе нейронных сетей.

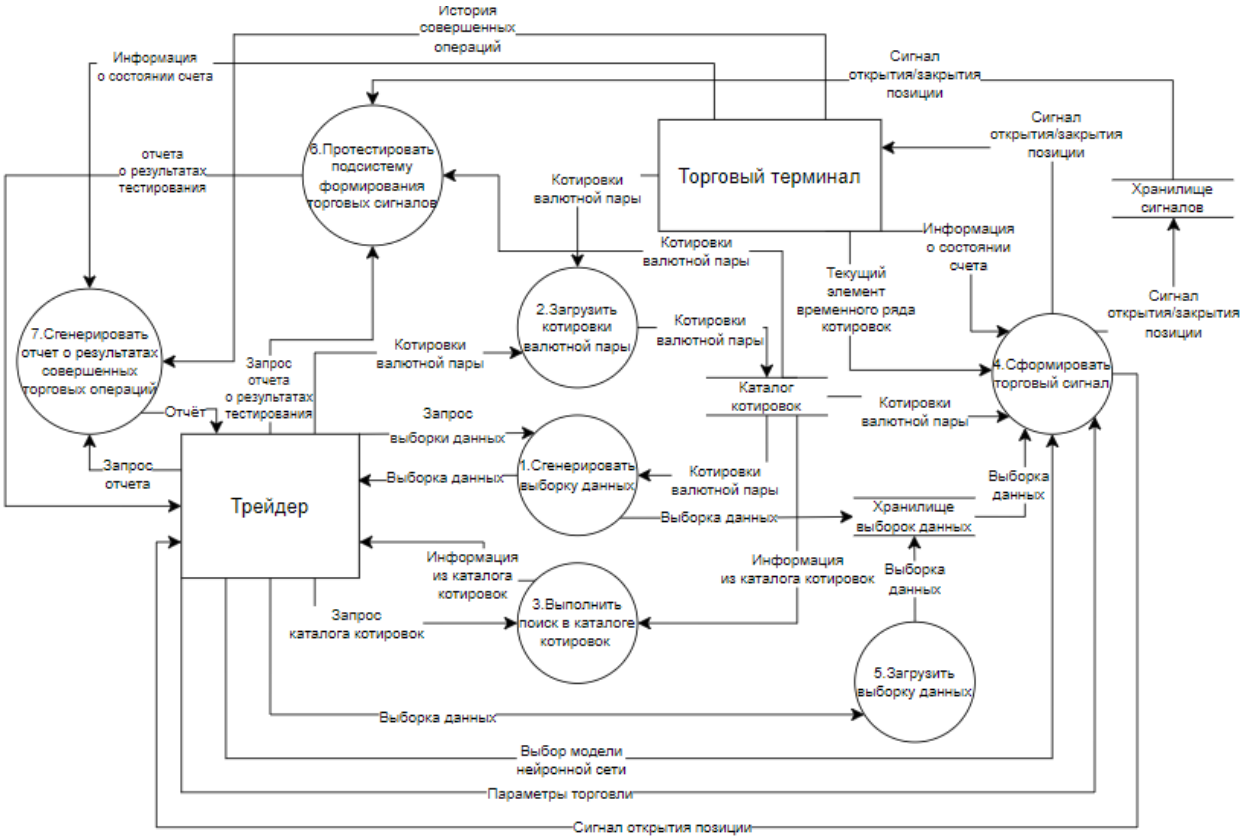

Рисунок 15 – Уровень 0 диаграммы потоков данных программной системы для трейдинга на основе нейронных сетей

3.4 Диаграмма деятельности системы

Диаграмма деятельности системы – это графическое представление последовательности действий и активностей, которые выполняются в системе или процессе. Она позволяет описать логику и поведение системы, показывая, как различные действия взаимодействуют между собой и какие условия и

переходы могут возникать. На рисунке 16 представлена диаграмма деятельности торговой системы и пользовательского интерфейса.

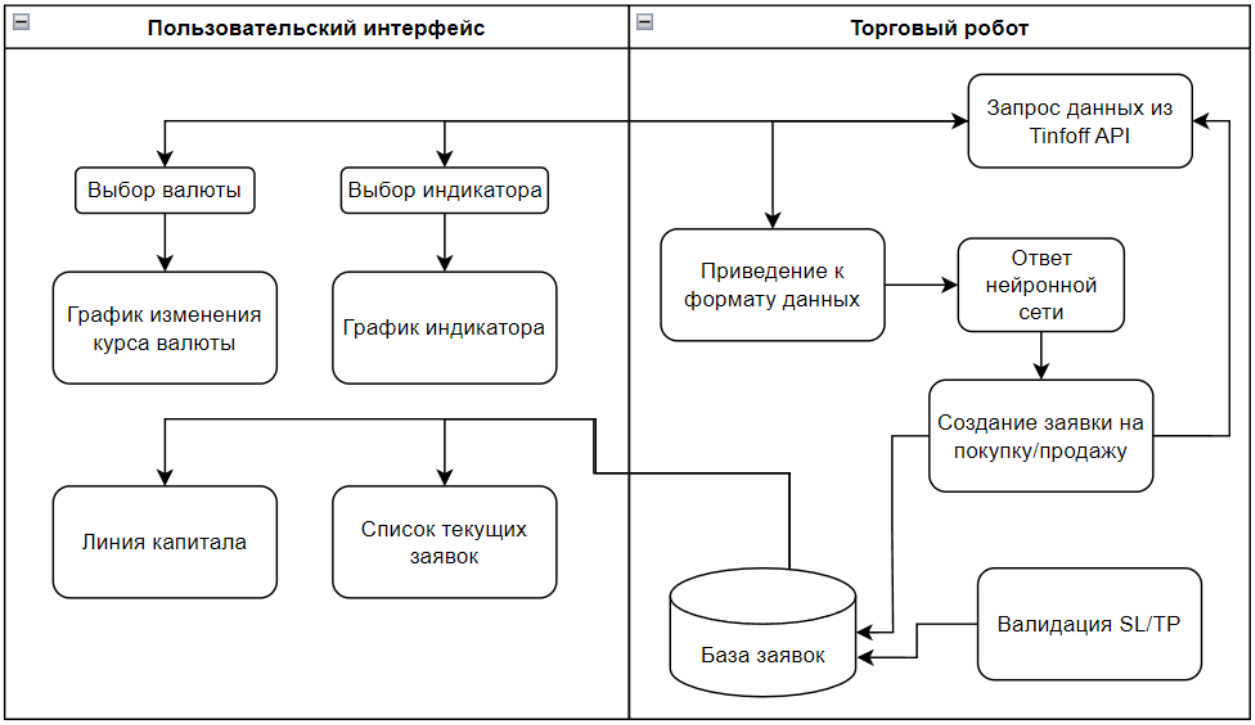

Рисунок 16 – Диаграмма деятельности системы

3.5 Use-case диаграмма

Use-case описывает последовательность взаимодействия системы и внешнего действующего лица, в результате которого действующее лицо получает полезный результат.

На рисунке 17 представлена use-case диаграмма системы для трейдинга на основе нейронных сетей.

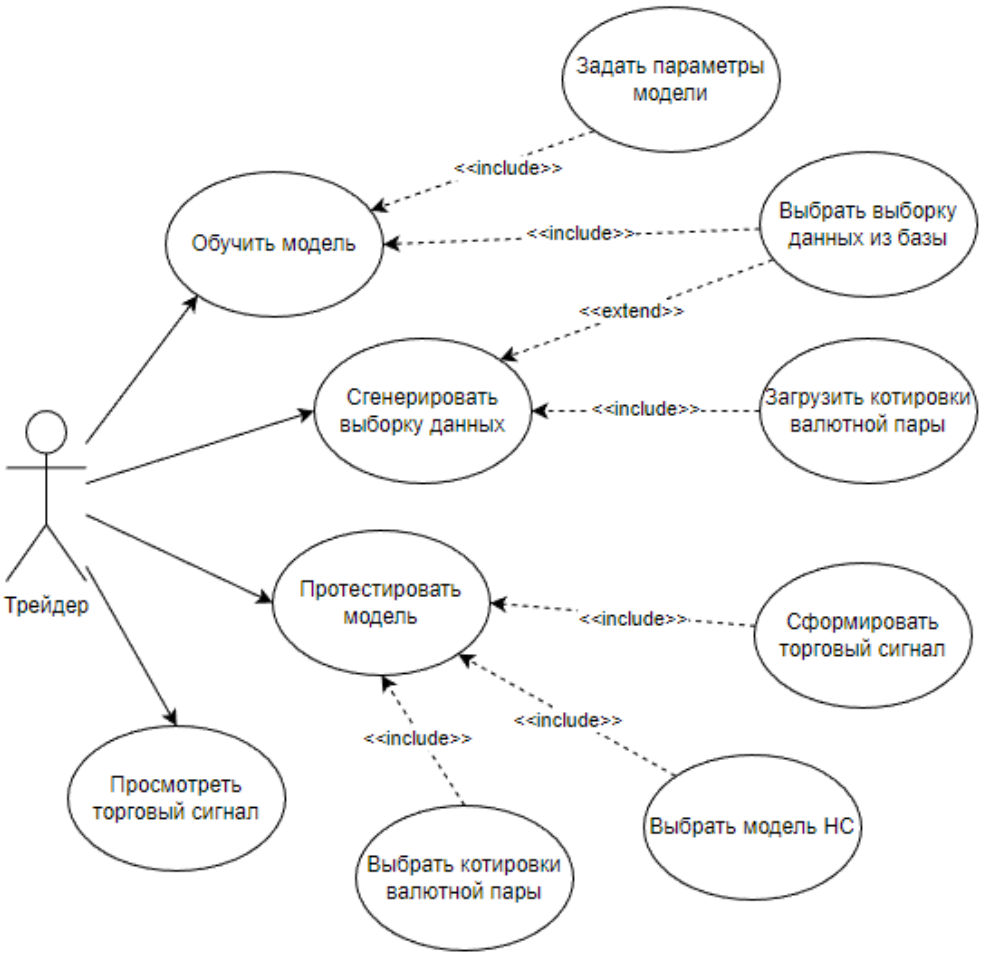

Рисунок 17 – Use-case диаграмма системы для трейдинга на основе нейронных сетей

3.6 Функциональные требования

Ниже представлена таблица, которая содержит функциональные требования к подсистемам пользовательского интерфейса и генерации выборок данных.

Т а б л и ц а 2 – Функциональные требования

Функция	Описание функциональных требований
Получение данных из терминала	Необходимо получать актуальные данные биржи в режиме реального времени. Под данными подразумевается актуальные цены закрытия на выбранную валюту
Создание заявок на покупку/продажу актива	Необходимо создавать заявки на покупку или продажу используя терминал биржи используя личный аккаунт

Окончание таблицы 2

Функция	Описание функциональных требований
Ведение статистики по совершенным сделкам	Необходимо хранить информацию о совершенных сделках
Ведение автоматической торговли	Необходимо реализовать алгоритм автоматической торговли на бирже. Алгоритм должен открывать и закрывать сделки согласно заранее заданным параметрам.
Создание двумерных выборок данных	Генератор выборок должен создавать двумерные матрицы данных из заданного временного ряда
Получение сигналов нейронной сети	Торговый робот должен использовать сигналы нейронной сети для открытия сделок. Для этого необходимо приводить данные в виде временного ряда к требуемому нейронной сетью формату

4 Реализация и тестирование системы

В данной главе будет описаны технологии и детали реализации программного средства для трейдинга на основе нейронных сетей. Так же будет проведено и описано тестирование. Программная система представляет собой композицию из трёх отдельных подсистем: торговый робот, программный интерфейс, пользовательский интерфейс. Файловая структура каждого из проектов представлена на рисунках 18–20.

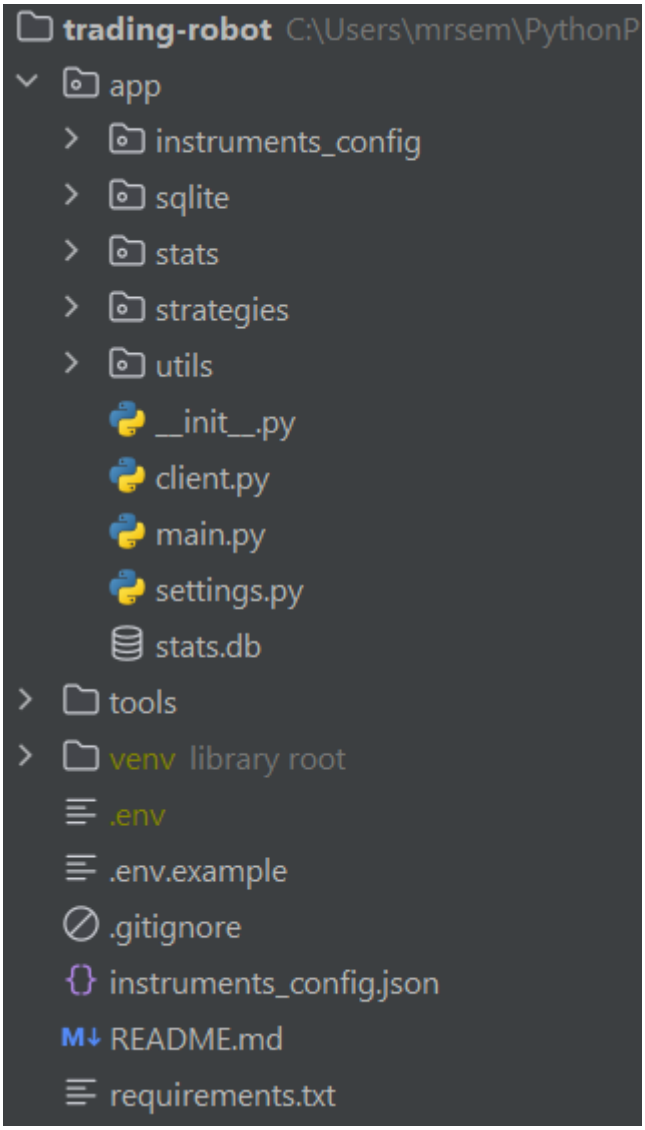

Рисунок 18 – Файловая структура подсистемы торговый робот

Подсистема торговый робот реализована на языке программирования Python с использованием стандартных библиотек. Задача подсистемы осуществлять автоматическую торговлю валютой через Tinkoff API и

фиксировать результаты сделок в базу данных, в данном случае используется SQLite.

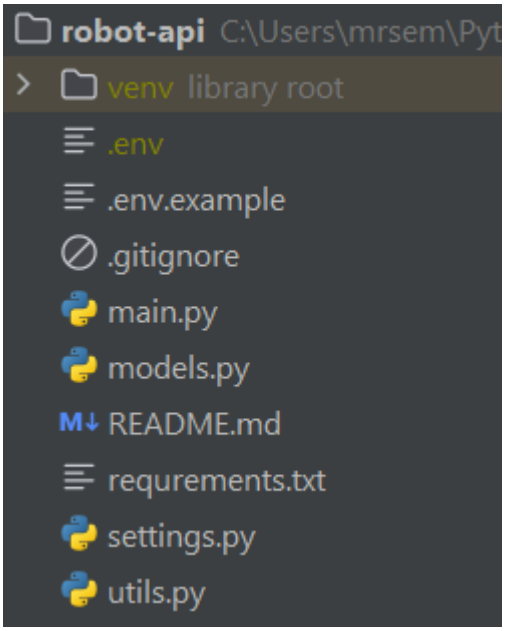

Рисунок 19 – Файловая структура подсистемы программный интерфейс

Задача подсистемы программный интерфейс – быть так называемой «прослойкой» между торговым роботом и пользовательским интерфейсом. Подсистема является «бэкендом» на языке программирования Python с использованием фреймворка FastAPI. Задача подсистемы принимать запросы от пользовательского интерфейса и отдавать ему данные из базы SQLite, которая заполняется на основе сделок торгового робота.

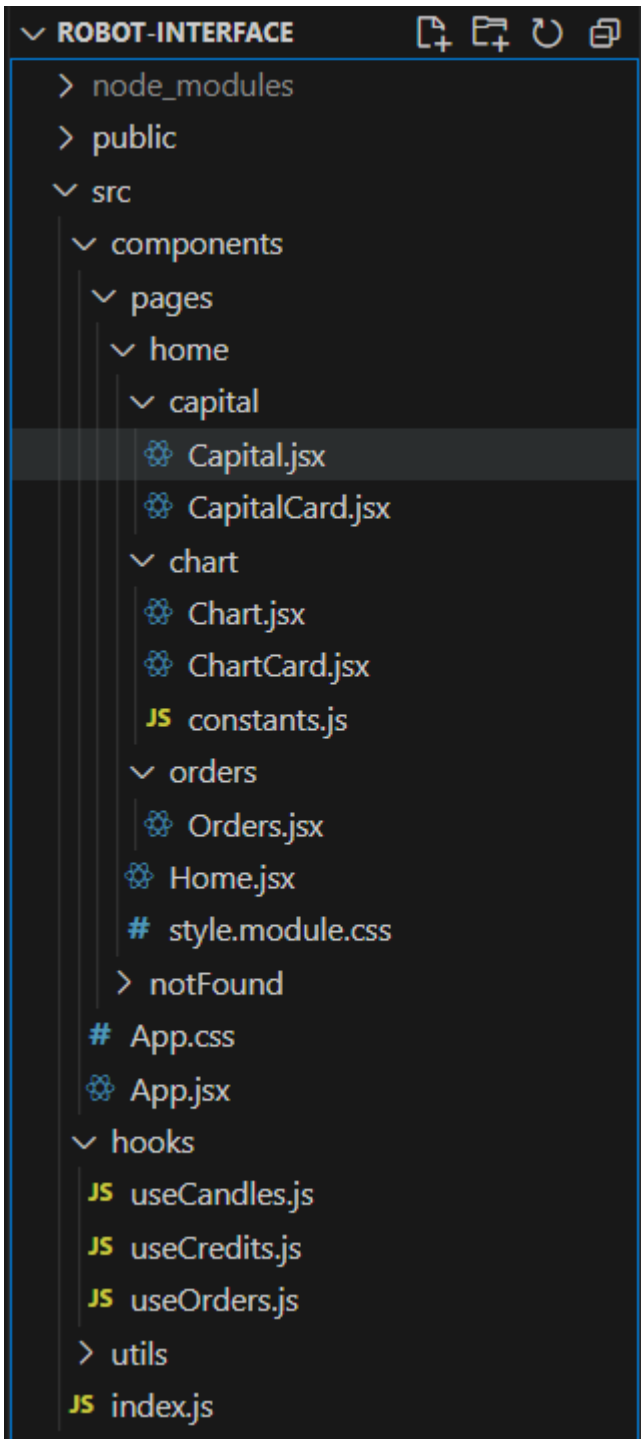

Рисунок 20 – Файловая структура подсистемы пользовательский интерфейс

Пользовательский интерфейс реализован с помощью библиотеки React для языка программирования JavaScript. Задача подсистемы отображать действия торгового робота.

4.1 Подсистема торговый робот

Для запуска торгового робота пользователю необходимо создать файл .env в корне проекта и задать токен для Tinkoff API, благодаря которому приложение получит доступ к аккаунту пользователя и получит разрешение на обработку запросов. Затем нужно настроить конфигурационный файл в формате .json, в нём указываются параметры алгоритма автоматической торговли. На рисунке 21 показан пример такого файла. Здесь figi – уникальный идентификатор валютной пары для Tinkoff API.

Рисунок 21 – Структура файла с настройками торговой системы

Для запуска торговли нужно запустить файл main.py. Робот, после того как удостоверится о том, что биржа открыта, начинает торговлю. Каждую минуту робот посылает запрос на Tinkoff биржу о получении последних цен закрытия заданной валютной пары. Далее на основе сигнала нейронной сети робот принимает решение о формировании запроса на открытие позиции. Если позиция открывается робот создает запись в базе данных. В конце итерации

робот проходит по всем открытым сделкам и принимает решение о закрытии на основе стоп-лосс тейк-профит стратегии. Так же, для удобства пользователя, робот логирует в консоль все свои действия.

4.2 Подсистема программный интерфейс

На рисунке 22 изображены «эндпоинты» программного интерфейса.

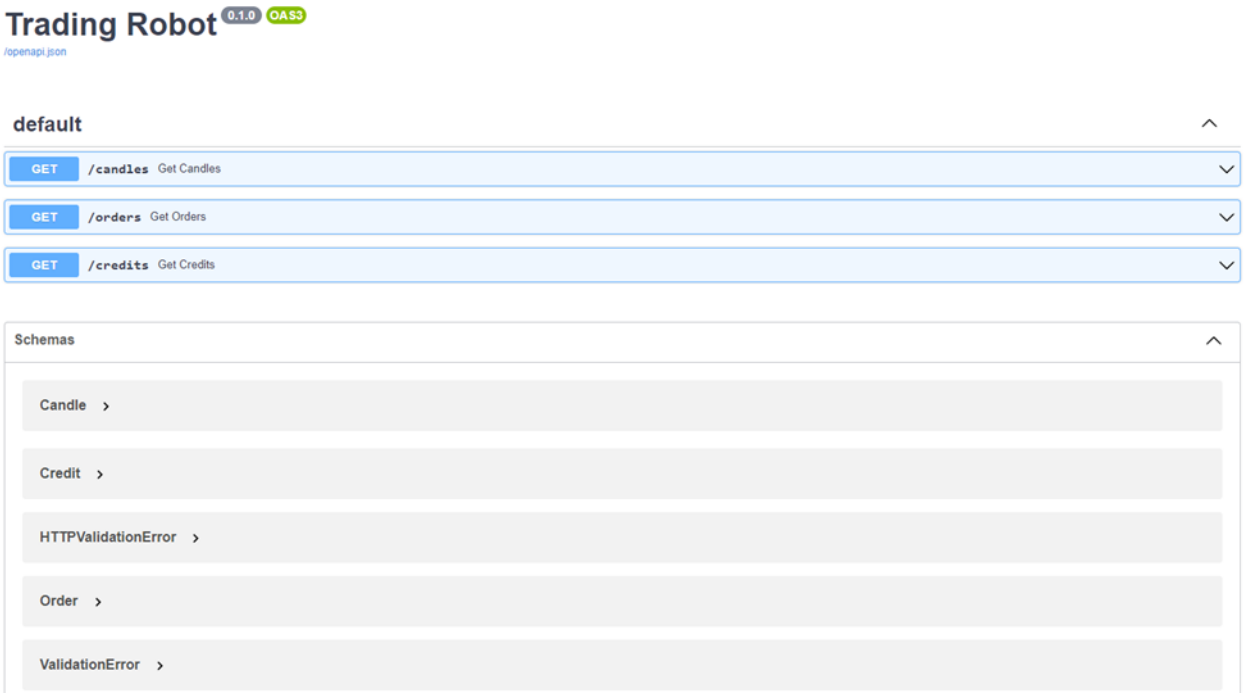

Рисунок 22 – Методы получения данных подсистемы программного интерфейса

4.3 Подсистема пользовательский интерфейс

При запуске подсистемы и переходе на сайт по адресу http://localhost:3000 вы увидите несколько окон, приведённых на рисунках 23–26.

Рисунок 23 – Окно отображения текущего курса валюты

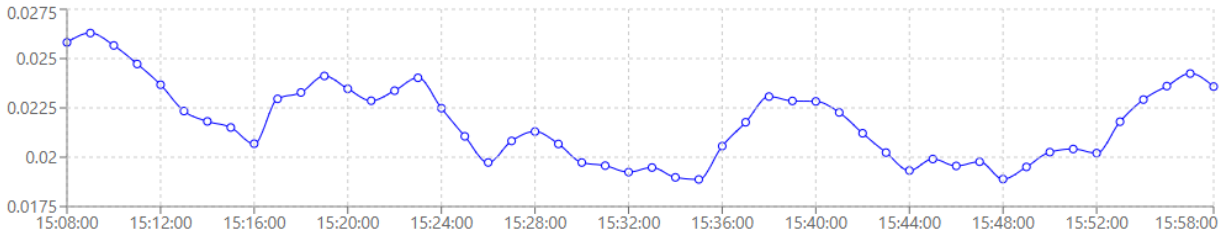

Рисунок 24 – Окно отображения выбранного технического индикатора

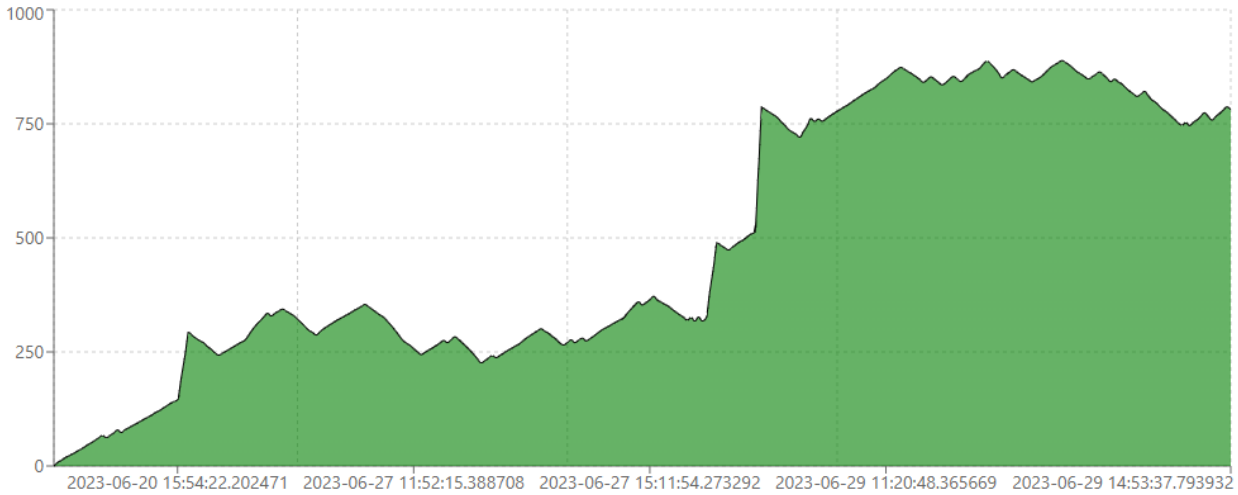

Рисунок 25 – Окно отображения линии капитала

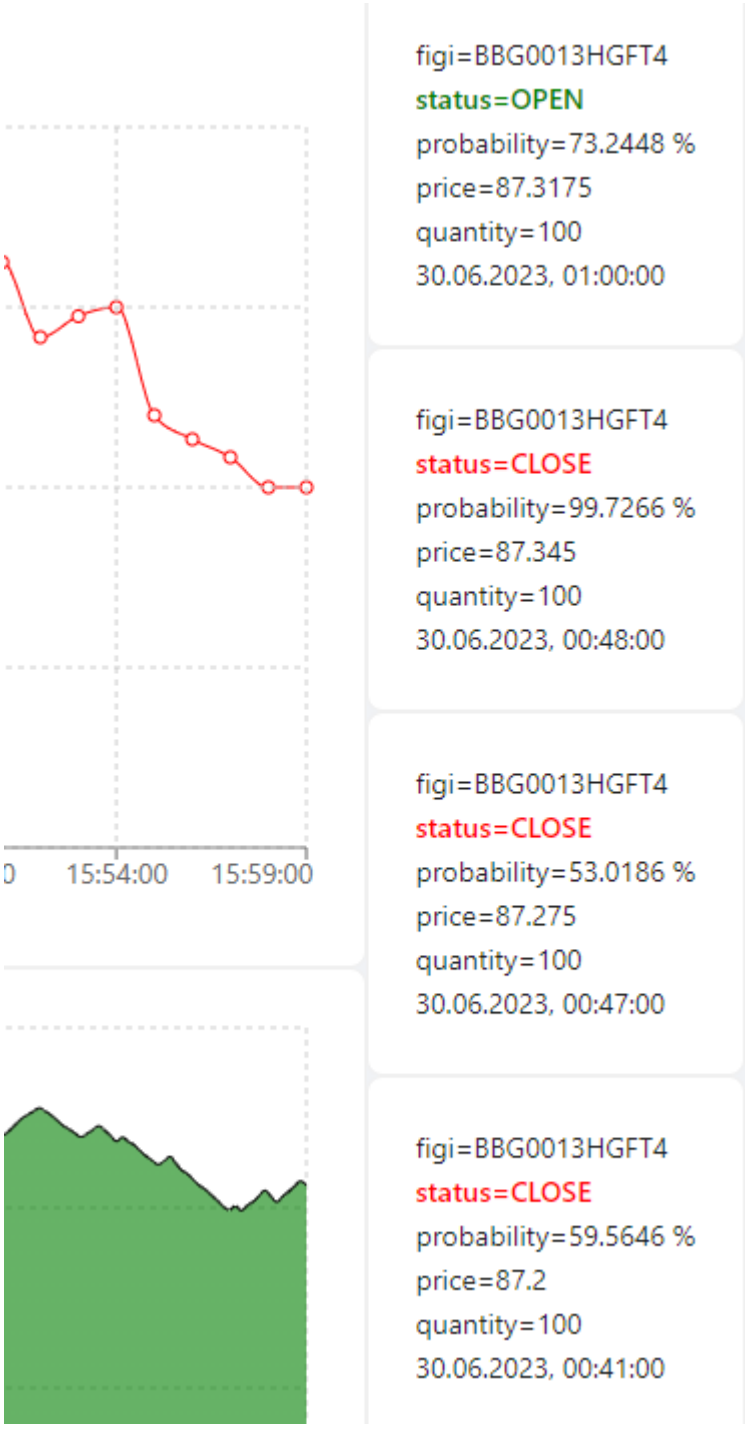

Рисунок 26 – Окно отображения текущих позиций

4.4 Тестирование программной системы

Для тестирования реализованной программной системы были реализованы тестовые ситуации, которые могут возникнуть у пользователя в процессе работы с системой. Подробнее о тестовых ситуациях смотрите в таблице 3.

Т а б л и ц а 3 – Тестовые ситуации

№	Тестовая ситуация	Начальное состояние системы	Действие	Ожидаемый результат
1	Запуск программы	Загрузка основных окон программы	Отправка запроса на получения данных от программного интерфейса	Отрисовка графиков согласно полученной информации
2	Выбор торгового инструмента	Окно выбора торгового инструмента	Пользователь выбирает торговый инструмент из выпадающего списка	Отрисовка графиков согласно полученной информации
3	Выбор технического индикатора	Окно выбора технического индикатора	Пользователь выбирает технический индикатор из выпадающего списка	Отрисовка графиков согласно полученной информации
4	Ошибка сервера при выборе торгового инструмента	Окно отображения графика котировок	Отправка запроса на сервер для получения данных	Окно вывода текста ошибки
5	Ошибка сервера при запросе истории изменения капитала	Окно отображения линии капитала	Отправка запроса на сервер для получения данных	Окно вывода текста ошибки
6	Ошибка сервера при запросе данных по текущим заявкам	Окно отображения текущих заявок	Отправка запроса на сервер для получения данных	Окно вывода текста ошибки

Заключение

Целью выпускной квалификационной работы являлась разработка программной системы для трейдинга на основе нейронных сетей, подсистемы генерации обучающих выборок и пользовательский интерфейс.

Для достижения поставленной цели были решены следующие задачи:

1) проведен обзор рассматриваемой предметной области;
2) проведен анализ предметной области и построена ее модель;
3) сформирован технический проект программной системы;
4) реализована и протестирована программная система.

Таким образом, поставленная цель выпускной квалификационной работы на тему «Разработка программной системы для трейдинга на основе нейронных сетей» была успешно достигнута. Перспективами развития данного программного средства являются разработка новых торговых стратегий, алгоритмов генерации обучающих выборок и моделей нейронных сетей, а также внедрение систем оповещения пользователя о ситуации на бирже.

Список источников

1. Лемеш В.Е Разработка программной системы для трейдинга на основе нейронных сетей. Подсистема формирования торговых сигналов // ВКР по направлению подготовки программная инженерия ДВФУ. 2023.
2. Ананченко И. В. Классификация и общая оценка торговых роботов для валютного рынка Forex // European research. 2020. – С. 24–26.
3. Ананченко И. В., Чагина П. А. Перспектива машинного обучения нейросетей для разработки и оптимизации торговых роботов // Евразийское Научное Объединение. 2021. № 1–2. – С. 78–82.
4. Ананченко И. В., Чагина П. А. Практическое применение нейронных сетей в трейдинге: проблемы и решения // European research. 2020. – С. 27–30.
5. Букунов С. В., Климин П. Ю. Автоматизированная торговая система для работы на финансовых рынках // Инженерный вестник. 2019. № 4. – С. 32.
6. Варламов М. С., Тесля Н. Б. Информационные технологии в финансовой сфере с использованием нейронных сетей // Экономика XXI века. 2020. – С. 455–458.
7. Васяева Т. А., Мартыненко Т. В., Суббота Н. С. Прогнозирование финансовых временных рядов с помощью нейронных сетей с использованием библиотеки Keras в Python // Информатика и кибернетика. 2019. № 2. – С. 41–50.
8. Галиуллина А. Ш. Нейронные сети и технический анализ в трейдинге // Научно-практические исследования. 2018. № 2. – С. 21–24.
9. Гасанов И. И., Ерешко А. Ф. Программный комплекс для высокочастотной биржевой торговли // Управление развитием крупномасштабных систем. 2021. – С. 405–413.
10. Гончаров А. Б., Исаев А. Л. О методах технического анализа, применяемых в торговых роботах частных экспертных систем поддержки принятия решений // Тенденции развития науки и образования. 2020. № 62–7. – С. 53–55.

11. Гюлумян А. Ю., Нишатов Н. П. Искусственный интеллект: эволюция, виды нейронных сетей, использование на финансовом рынке // Финансовые рынки и банки. 2018. № 1. – С. 74–78.

12. Железко Б. А., Стадник А. О., Синявская О. А. Использование технического анализа и индикаторов в алгоритмическом трейдинге // Экономическая наука сегодня. 2022. № 15. – С. 119–130.

13. Иванов М. А. Использование торговых роботов на рынке Форекс: преимущества и недостатки // European science. 2018. № 6. – С. 23–27.

14. Иэн Гудфеллоу. Глубокое обучение: учеб. пособие / Иэн Гудфеллоу, Йошуа Бенджио. – Изд. 2-е: MIT Press. 2016. 654 с.

15. Коноплева Ю. А., Пакова О. Н., Гаврилов К. К. Валютные рынки и валютные операции в условиях цифровизации // Вестник Северо-Кавказского федерального университета. 2020. № 6. – С. 104–111.

16. Документация к Tinkoff Invest API. 2020 – Режим доступа: https://tinkoff.github.io/investAPI/glossary/

17. Плеханов Л., Плеханов С. Нейронные сети как инструмент распознавания фигур. // Рынок ценных бумаг. №3–2005. – С. 68–72.

18. Саймон Хайкин. Нейронные сети: полный курс – 2006 г. – С. 219–241.

19. Чагина П. А. Анализ возможности применения машинного обучения и нейронных сетей̆ для проектирования и оптимизации торговых роботов // Альманах научных работ молодых ученых Университета ИТМО. 2022. – С. 388–390.

20. Чагина П. А. Использование возможностей нейронных сетей для оптимизации алгоритмов торговых роботов // Молодежный исследовательский потенциал. 2021. – С. 15–28.

21. Шангаряев Т. М., Кусков В. М. Торговые роботы как инструмент совершения торговых операций на финансовых рынках // Colloquium-journal. 2019. № 13–11. – С. 191–193.

22. Программа «Amibroker» – Режим доступа: https://amibroker.com/

23. Программа «Neuroshell Trader» – Режим доступа: https://try.neuroshell.com/features/

24. Weber E.A. Transfer Learning with Time Series Data: A Systematic Mapping Study. // IEEE Access, Dec. 2021. 25 с.

25. Aslan O., Yilmaz A.A. A New Malware Classification Framework Based on Deep Learning Algorithms. // IEEE Access, Jun. 2021. 17 с.

26. Sidike Paheding, Abel A. Reyes, Anush Kasaragod, Thomas Oommen «Gramian Angular Field encoded Neighborhood Attention U-Net for Pixel-Wise Hyperspectral Image Classification» – Michigan Technological University Houghton, MI, USA 2019. 409-416 с.

27. Jin Xiao, Yi Hu. A hybrid transfer learning model for crude oil price forecasting. // Statistics and its Interface, Jan. 2017. 13 с.

28. Ruijun Liu, Yuqian Shi. A Survey of Sentiment Analysis Based on Transfer Learning. // IEEE Access, Jun. 2019. 12 с.

29. Pan S.J., Yang Q. A survey on transfer learning. // IEEE Access, Oct. 2009. 14 с.

30. Jordan J. B. Cross-Domain MLP and CNN Transfer Learning for Biological Signal Processing: EEG and EMG. // IEEE Access, Jan. 2020. 13 с.