

УДК 519.7

Какойченко А. І.¹, студент,
Порхун О. В.², к.ф.-м.н., асистент.

Використання RBF-мережі для короткострокового прогнозування фінансових ринків

¹ Київський національний університет імені
Тараса Шевченка, 83000, м. Київ, пр-т.
Глушкова 4д,
e-mail: a.kakoychenko@gmail.com

² Київський національний університет імені
Тараса Шевченка, 83000, м. Київ, пр-т.
Глушкова 4д,
e-mail: elena_porkhun@mail.ru

A. I. Kakoichenko¹, student,
O. V. Porkhun², Ph.D., Assistant Professor.

Short-term financial markets forecasting with RBF network

¹ Taras Shevchenko National University of Kyiv,
83000, Kyiv, Glushkova st., 4d,
e-mail: a.kakoychenko@gmail.com

² Taras Shevchenko National University of Kyiv,
83000, Kyiv, Glushkova st., 4d,
e-mail: elena_porkhun@mail.ru

У даній статті розглянуто процес побудови програмного забезпечення для короткострокового прогнозування напряму динаміки ринку за допомогою RBF-мережі. Тестування отриманої програми з використанням даних про всі заявки ф'ючерсного контракту на індекс РТС показало широкі перспективи даного підходу.

Ключові слова: фондовий ринок, прогнозування, RBF-мережі, машинне навчання.

The purpose of the paper is to demonstrate the process of developing the algorithm for short-term financial markets forecasting with RBF-networks (radial basis function networks). The paper also includes overview of the RBF-networks algorithm and deep analysis of the market microstructure. Predictive model based on a fact, that the financial markets pricing process is unambiguously described by flows of transactions. There are three most popular types of transactions: add order, remove order and modify order. Classifier analyzes the features vector, calculated with these flows and state of the order book, and gives predictions about the further market direction. Algorithm application includes learning with the historical data for the RBF-network estimation.

Verification of the software implementation of the algorithm with the data about transactions of the RTS index future contract showed bright results and wide prospects of this approach for risk calculating algorithms and high frequency trading or market-making strategies developing.

Key Words: stock market, forecasting, RBF-networks, machine learning.

Статтю представив чл.-кор. НАН України, д.ф.-м.н. Анісімов А.В.

Використання алгоритмів машинного навчання для прогнозування стану ринку є досить популярним напрямком штучного інтелекту. Як правило, такі дослідження побудовані на аналізі і прогнозуванні цінових рядів. Стан ринку у деякий момент часу у таких роботах описується цінами фінансових інструментів. Таким чином ігнорується інформація про поточний стан та динаміку книги заявок, що робить такі підходи недостатньо ефективними для короткострокового прогнозування та розробки стратегій високочастотної торгівлі.

Однак розвиток високочастотної торгівлі у середині 2000-х років та потреба у алгоритмах для автоматизованого підтримання ліквідності призвів до появи робіт, основаних на аналізі стану книги заявок та характеристиках потоків надходження та видалення нових заявок. Наприклад, у роботі [1] ставиться задача визначення оптимальної стратегії маркет-мейкінгу на американських акціях. Більш цікавою може здатися стаття [2]. У ній дослідники звертають увагу на потоки транзакцій на постановку, видалення та задоволення лімітних заявок на строковому ринку

Московської біржі. Автори припускають, що проміжки часу між транзакціями мають гамма розподіл з параметрами, залежними від часу, хоч статистична перевірка цього твердження авторами даного дослідження і показала його суперечливість [3].

Метою даного дослідження є розробка та

де $n_s(t)$ – кількість лімітних заявок на продаж у момент часу t , $n_B(t)$ – кількість лімітних заявок на купівлю у момент часу t .

Лімітна заявка на продаж

$$s_i = \langle p_i^s, a_i^s, t_i^s \rangle,$$

де p_i – ціна заявки, a_i – залишковий обсяг заявки, t_i – момент часу постановки заявки у чергу.

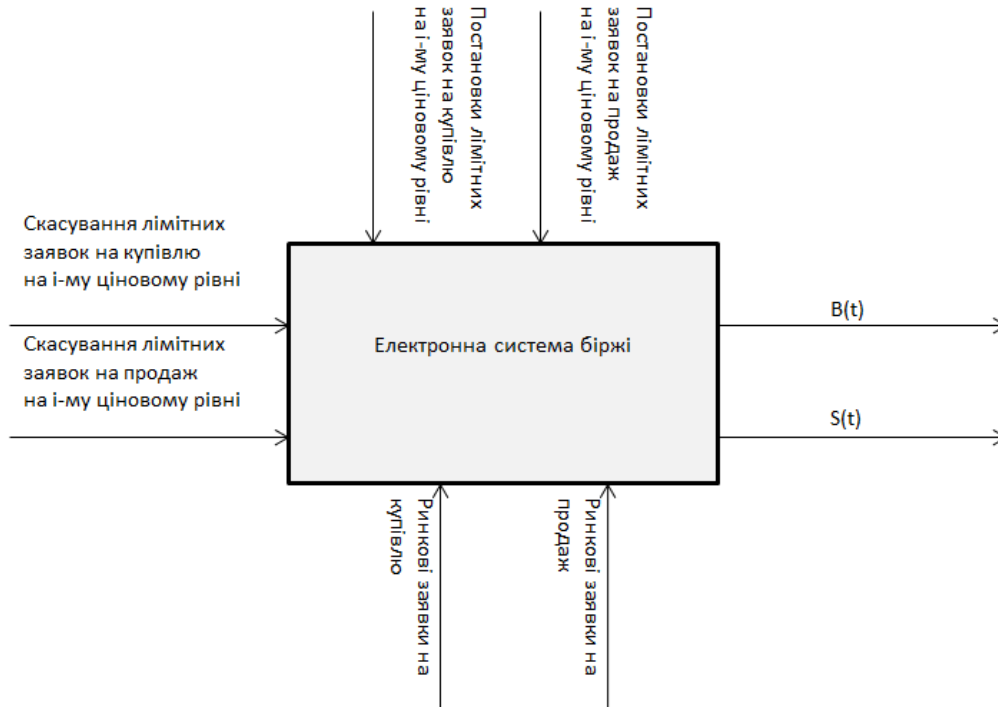


Рис. 1 Схема функціонування електронної системи біржі

подальша перевірка ефективності програмного забезпечення для короткострокового прогнозування динаміки фінансових ринків за допомогою алгоритмів машинного навчання.

Модель ринку

На електронних ринках з неперервним двостороннім аукціоном біржова ціна фінансового інструменту є результуючою характеристикою торгівельної системи, поточний стан якої можна розглядати, як, так звану, книгу заявок. Книга заявок є чергою з пріоритетом. У даному випадку пріоритетом є ціна (у черзі заявок на купівлю більша ціна має більший пріоритет, а у черзі заявок на продаж менша ціна має більший пріоритет). Тобто першою у книзі заявок є заявка, яка першою увійшла до книги серед заявок, що мають найбільший пріоритет(1). Будемо у кожний момент часу $t \in T$, де T – множина моментів часу у торговій сесії, розглядати множину заявок на купівлю $B(t)$ та на продаж $S(t)$.

$$S(t) = \{s_1, s_2, \dots, s_{n_s(t)}\},$$

$$B(t) = \{b_1, b_2, \dots, b_{n_B(t)}\},$$

Лімітна заявка на купівлю

$$b_i = \langle p_i^b, a_i^b, t_i^b \rangle,$$

де p_i – ціна заявки, a_i – залишковий обсяг заявки, t_i – момент часу постановки заявки у чергу.

Тобто, згідно з (1) для $S(t)$ маємо: $i < j \leftrightarrow (p_i^s < p_j^s) \vee ((p_i^s = p_j^s) \wedge (t_i^s < t_j^s))$, для $B(t)$ маємо: $i < j \leftrightarrow (p_i^b > p_j^b) \vee ((p_i^b = p_j^b) \wedge (t_i^b < t_j^b))$.

На динаміку книги заявок впливають такі події:

- 1) Постанова лімітної заявки. Означає намір придбати або продати певний обсяг активу за певною ціною. Заявка додається до книги заявок. Якщо після постановки заявки виникає ситуація, що ціна першої заявки у черзі заявок на купівлю більша, чи рівна ціні першої заявки у черзі заявок на продаж, то послідовно відбуваються угоди між новою заявкою та першою заявкою у черзі заявок з протилежним напрямом, доки ціна першої заявки у черзі заявок на купівлю не стане меншою

за ціну першої заявки у черзі заявок на продаж.

- 2) Ринкова заявка. Означає намір миттєво придбати або продати певний обсяг фінансового інструменту за ринковою ціною. При постановці ринкової заявки з обсягом k електронна система біржі послідовно проводить угоди між ринковою заявкою та першою заявкою у черзі з протилежним напрямом, доки не вичерпається обсяг ринкової заявки, чи усі лімітні заявки з протилежним напрямом не будуть вдоволені. Якщо обсяг усіх заявок з протилежним напрямом вичерпується раніше, ніж обсяг ринкової заявки, то клієнт отримує виконання лише на різницю між початковим та залишковим обсягом заявки, і така заявка більше у торговельному процесі участі не бере. На деяких ринках (наприклад, на усіх ринках Московської та Української біржі) ринкові заявки не підтримуються електронною системою біржі. У даній роботі будемо вважати ринковими лімітні заявки такі, що після постановки цих заявок виникає ситуація, що ціна першої заявки у черзі заявок на купівлю більша, чи рівна ціні першої заявки у черзі заявок на продаж.
- 3) Відміна лімітної заявки. Означає скасування та вилучення з книги заявок певної заявки.
- 4) Зміна параметрів лімітної заявки. Означає відміну заявки та моментальну постановку нової з новими параметрами. Ми будемо розглядати таку транзакцію, як дві послідовні транзакції зняття лімітної заявки та постановки нової.

Ці чотири потоки однозначно описують стан торговельної системи у довільний момент часу.

Кращу ціну купівлі визначатимемо, як

$$b(t) = \begin{cases} \sup\{p_i^b: < p_i^b, a_i^b, t_i^b > \in B(t)\}, \\ \text{якщо } n_B(t) > 0 \\ 0, \text{ якщо } n_B(t) = 0 \end{cases}.$$

А кращу ціну продажу визначатимемо, як

$$s(t) = \begin{cases} \inf\{p_i^s: < p_i^s, a_i^s, t_i^s > \in S(t)\}, \\ \text{якщо } n_S(t) > 0 \\ p_{max}, \text{ якщо } n_B(t) = 0 \end{cases},$$

де p_{max} - деяке заздалегідь визначене велике число, таке, що ринкова ціна не може його перевищувати. Наприклад, при програмуванні моделі мовою C# та використанні типу decimal для опису цін, можна використовувати

максимальне значення для цього типу – 79228162514264337593543950335.

Алгоритм машинного навчання

Для даної задачі було використано мережу радіальних базисних функцій (RBF-мережу). Це нейронна мережа прямого поширення сигналу, яка містить проміжний (прихований) шар радіально симетричних нейронів. Такий нейрон перетворює відстань від даного вхідного вектора до відповідного йому "центру" по деякому нелінійному закону. Радіальна функція - це функція $f(x)$, що залежить тільки від відстані між x і фіксованою точкою простору X . Для визначення наших радіальних функцій введемо метрику: Нормальний розподіл (Гауссіан) $p_j(x) = N(x; \mu_j, \Sigma_j)$ з діагональною матрицею коваріації Σ_j можна записати у вигляді

$$p_j(x) = N_j e^{-\frac{1}{2\rho_j(x, \mu_j)}},$$

де $N_j = (2\pi)^{-\frac{n}{2}} (\sigma_{j1}, \dots, \sigma_{jn})^{-1}$ - нормувальний множник,

$\rho_j(x, x')$ - зважена евклідова метрика в n -вимірному просторі X :

$$\rho(x, x') = \sum_{d=1}^n \sigma_{jd}^{-2} |\xi_d - \xi'_d|,$$

$$x = (\xi_1, \dots, \xi_n),$$

$$x' = (\xi'_1, \dots, \xi'_n).$$

Чим менше відстань $\rho_j(x, \mu_j)$, тим вище значення щільності в точці x . Тому щільність $p_j(x)$ можна розглядати як функцію близькості вектора x до фіксованого центру.

Постановка завдання полягає у тому, щоб побудувати алгоритм, який би вирішував завдання класифікації Байєсівським алгоритмом (окремий випадок EM-алгоритму) в припущенні, що щільність розподілу подана в вигляді суміші гаусівських розподілів з діагональними матрицями коваріації.

Нехай $|Y| = M$ - число класів, кожен клас $y \in Y$ має свою щільність розподілу $p_y(x)$ і представлений частиною вибірки $X_y^l = \{(x_i, y_i) \in X^l | y_i = y\}$. Тут Y - множина відповідей (класів), $y \in Y$, x_i належить множині об'єктів X .

Висувається гіпотеза, що щільності класів $p_y(x)$, $y \in Y$, можливо представити у вигляді комбінації k_y компонент. Кожна компонента має n - вимірну гаусівську щільність з параметрами:

$\mu_{yj} = (\mu_{yj1}, \dots, \mu_{yjn})$ - центр,

$\Sigma_{yj} = \text{diag}(\sigma_{yj1}, \dots, \sigma_{yjn})$ - коваріаційна матриця, $j = 1, \dots, k_y$.

$p_y(x) = \sum_{j=1}^{k_y} \omega_{yj} p_{yj}(x)$ - комбінація щільностей,
 $p_{yj}(x) = N(x; \mu_{yj}, \Sigma_{yj})$ - щільність кожної
компоненти комбінації.
 $\sum_{j=1}^{k_y} \omega_{yj} = 1, \omega_{yj} > 0$ - умови нормування та
невід'ємності ваг.

Алгоритм класифікації

Запишемо основну формулу байесівського

приймаючим остаточне рішення про віднесення
об'єкта x до одного з класів. Таким чином, при
класифікації об'єкта x оцінюється його
близькість до кожного з центрів μ_{yj} по метриці
 $\rho_{yj}(x, \mu_{yj}), j = 1, \dots, k_y$. Об'єкт відноситься до
того класу, до чиїх центрів він розташовується
ближче.

Описаний тривірневий алгоритм класифікації

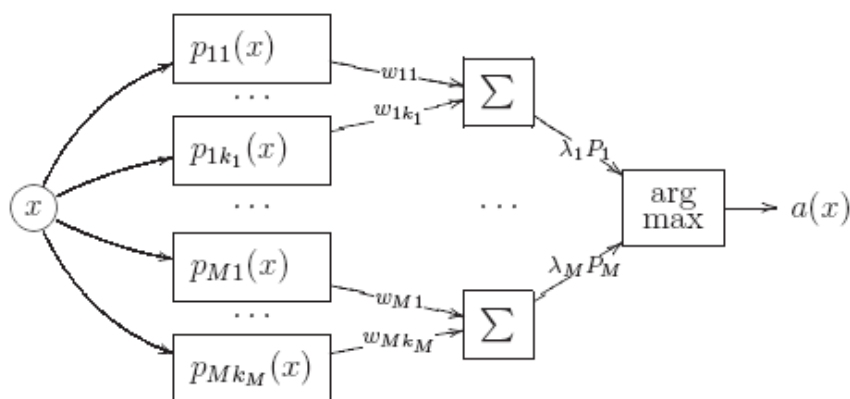


Рис. 2 Схема RBF – мережі

класифікатора $a(x) = \operatorname{argmax}_{y \in Y} \lambda_y P_y p_y(x)$. Тут
 Y - множина відповідей (класів), $x \in X$, P_y -
апріорна ймовірність класу y , $p_y(x)$ – функція
правдоподібності класу y , λ_y - ціна помилки на
об'єкті класу y . Виразимо щільність кожної
компоненти $p_{yj}(x)$ через зважену евклідову
відстань від об'єкта x до центра компоненти μ_{yj}
(іншими словами – підставимо в основну
формулу байесівського класифікатора замість
 $p_y(x)$ формули, які ми припустили в гіпотезі):

$$a(x) = \operatorname{argmax}_{y \in Y} \lambda_y P_y \sum_{j=1}^{k_y} N_{yj} e^{-\frac{1}{2\rho_{yj}(x, \mu_{yj})}},$$

де $N_{yj} = (2\pi)^{-\frac{n}{2}} (\sigma_{yj1}, \dots, \sigma_{yjn})^{-1}$ - нормувальні
множники. Алгоритм має вигляд нейронної
мережі, що складається з трьох рівнів або шарів.

Перший шар утворений $k_1 + \dots + k_M$
гауссіанами $p_{yj}(x), y \in Y, j = 1, \dots, k_y$. На вході
вони приймають опис об'єкта x , на виході
видають оцінки близькості об'єкта x до центрів
 μ_{yj} , рівні значенням щільностей компонент в
точці x . Другий шар складається з M суматорів,
що обчислюють зважені середні цих оцінок з
вагами ω_{yj} . На виході другого шару з'являються
оцінки близькості об'єкта x кожному з класів,
рівні значенням щільностей класів $p_y(x)$. Третій
шар утворюється єдиним блоком argmax ,

називається мережею з радіальними базисними
функціями або RBF-мережею (radial basis function
network). Це одна з різновидів нейронних мереж.

Програмне забезпечення

В межах цієї роботи ми будемо
використовувати декілька модулів програмного
комплексу Watcher та програмне забезпечення
для проведення аналізу даних Weka. Програмний
комплекс Watcher призначений для накопичення
та систематизації біржових історичних даних,
аналізу, розробки, тестування, оптимізації та
застосування торгових стратегій для строкового
ринку. Розроблений співавтором даної роботи А.
Какойченком протягом 2011-2013 років для
особистого користування. Використані мови
програмування C# та C++. Підтримує біржові
протоколи прямого доступу до серверів
Української та Московської бірж, та пройшов
сертифікацію на обох біржах.

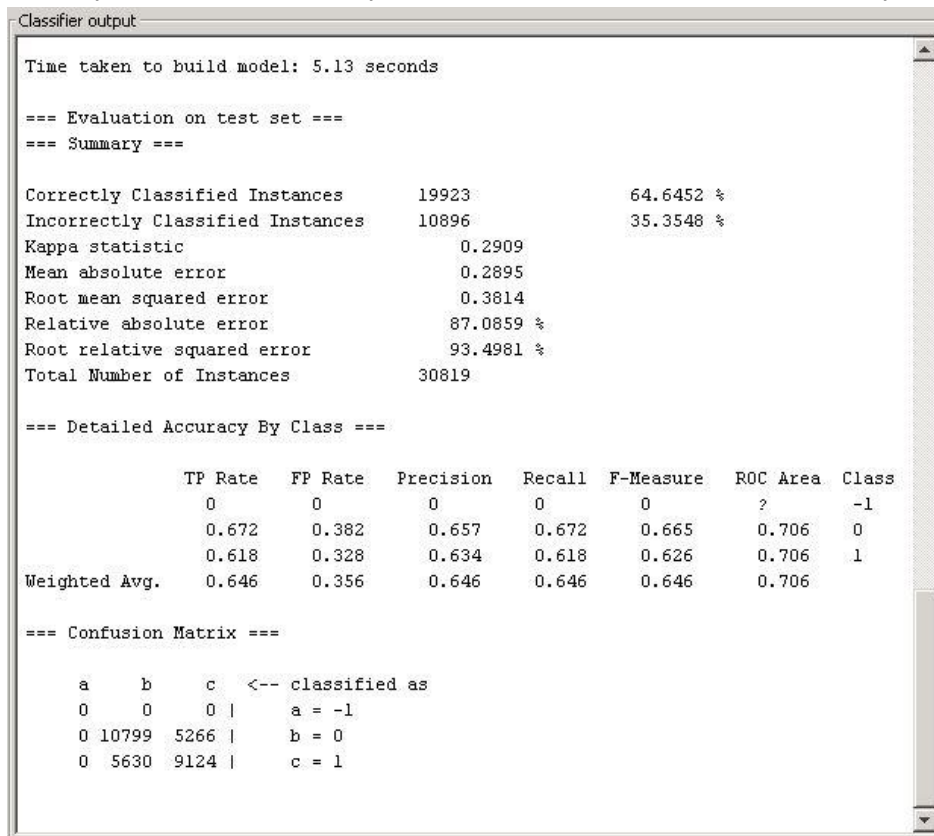
Програмний комплекс складається з чотирьох
основних модулів:

- 1) модуль для взаємодії з електронною
системою бірж (застосування стратегій
та накопичення даних);
- 2) сховище історичних даних;
- 3) тестовий полігон для аналізу
історичних даних, розробки,
тестування, та оптимізації стратегій;
- 4) модуль для взаємодії з тестовим
полігоном.

Weka — це набір засобів візуалізації та алгоритмів для аналізу даних і вирішення задач прогнозування, разом з графічною оболонкою для доступу до них. Weka дозволяє виконувати такі завдання аналізу даних, як підготовку даних

навчальну вибірку взято дані за 20 січня, а за тестову за 22.

На основі ринкових даних формується послідовність векторів ознак. Кожному вектору співставляється момент часу з точністю до



```
Classifier output
Time taken to build model: 5.13 seconds

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances      19923           64.6452 %
Incorrectly Classified Instances    10896           35.3548 %
Kappa statistic                    0.2909
Mean absolute error                 0.2895
Root mean squared error            0.3814
Relative absolute error             87.0859 %
Root relative squared error        93.4981 %
Total Number of Instances         30819

=== Detailed Accuracy By Class ===

              TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
              0       0       0          0       0          ?       -1
              0.672  0.382  0.657     0.672  0.665     0.706     0
              0.618  0.328  0.634     0.618  0.626     0.706     1
Weighted Avg.  0.646  0.356  0.646     0.646  0.646     0.706

=== Confusion Matrix ===

  a    b    c  <-- classified as
  0    0    0 |  a = -1
  0 10799 5266 |  b = 0
  0  5630 9124 |  c = 1
```

Рис.3 Результати тестування класифікатора у програмі Weka

(preprocessing), відбір ознак (feature selection), кластеризацію, класифікацію, регресійний аналіз та візуалізацію результатів.

Так, як Weka створено за допомогою мови Java, то це створює деякі перепони для інтеграції цих програмних засобів, так, як C# не підтримує прямої взаємодії з Java-бібліотеками. Через це було створено допоміжну програму мовою Java, яка взаємодіє з Weka API. Ця програма являє собою багатопотоковий сервер класифікації з доступом через TCP протокол. При отриманні відповідного вектору входів та номеру класифікатора вона повертає індекс вихідного класу. Відповідно, було створено модуль взаємодії з цим сервером для програмного комплексу Watcher.

Кроки дослідження

Для проведення дослідження використано дані з повного журналу заявок Московської біржі за 20 та 22 січня 2014 року по ф'ючерсному контракту на індекс РТС. Дані розділяються на дві частини: навчальну вибірку та тестову. За

секунди та знак (“+” чи “-“) сукупного обсягу угод проведених у наступні 5 секунд, що йдуть за моментом часу, якому відповідає даний вектор ознак. Причому, обсяги купівель сумуються зі знаком “+”, а обсяги продажів зі знаком “-“. Для визначеності вважаємо, що якщо сукупний обсяг рівний нулеві, то його знак вважаємо за “+”.

На основі навчальної та тестової вибірок біржових даних формуються послідовності таких векторів і записуються у текстові файли разом із відповідним знаком (+1 чи -1). У пакеті Weka на базі навчальної послідовності створюється класифікатор. Цей класифікатор тестується за допомогою тестової послідовності. Також, експортуючи файл класифікатора, можна його використовувати через Weka API для використання результатів класифікації у торгових стратегіях та для візуалізації.

Отже, на вхід подається вектор параметрів, що характеризують стан ринку та його динаміку на даний момент часу. Так, як у даній моделі поточний стан ринку характеризується книгою

заявок, то на її основі можна розрахувати ряд характеристик. Будемо розглядати середні ціни, за якими можна купити та продати n одиниць фінансового інструменту у час t . Позначимо їх $mb(n, t)$ та $ms(n, t)$ відповідно. Якщо сукупний обсяг лімітних заявок на продаж менший за n , то покладемо значення $mb(n, t)$ рівним p_{max} . Якщо сукупний обсяг лімітних заявок на купівлю менший за n , то покладемо значення $ms(n, t)$ рівне 0.

Неважко побачити, що отримані нами характеристики нестационарні, що унеможливорює їх пряме подання на вхід класифікатора. Проте, можна увести функцію $P(t)$, яка визначається, як середнє арифметичне від ціни першої заявки у черзі заявок на купівлю та першої заявки у черзі заявок на продаж у момент часу t . Тоді, віднявши від $P(t)$ $mb(n, t)$ та від $ms(n, t)$ $P(t)$, можна ввести такі характеристики, як відносні середні ціни, за якими можна купити та продати n одиниць фінансового інструменту у час t . Ці характеристики стаціонарні і можуть подаватися на вхід класифікатора.

Також розглянемо послідовності подій надходжень нових транзакцій. Нас цікавить інтенсивність потоків транзакцій, через це будемо підраховувати кількості та сукупні обсяги транзакцій, що відбуваються за останні 5 та 25 секунд окремо для транзакцій на постановку та

зняття лімітних та надходження ринкових заявок. Причому, окремо для транзакцій, що стосуються заявок на купівлю та продаж.

Також, додамо до вектора ознак напрям останньої угоди та відхилення ринкової ціни від середньої ціни за 5 та 25 секунд.

Отриманий вектор подається на вхід класифікатора. З результатами класифікації, отриманими на тестовій вибірці можна ознайомитися на рис. 3. Видно, що вдалося досягти результату у 65% правильно класифікованих векторів ознак.

Висновки

В процесі роботи над дослідженням було проведено глибокий аналіз мікроструктури ринку і розроблено алгоритм прогнозування напрямку динаміки ринку у короткотерміновій перспективі на основі застосування RBF-мережі до вектору ознак, що описують стан ринку та його динаміку, використовуючи книгу заявок та потоки транзакцій. Перевірка програмної реалізації алгоритму на історичних даних показала результат у 65% правильно класифікованих векторів ознак на тестовій вибірці, що є досить хорошим результатом. Перспективою подальших досліджень є застосування даного підходу для побудови високочастотних стратегій та створення алгоритмів для розрахунку короткотермінових ризиків.

Список використаних джерел

1. Avellaneda M. High-frequency trading in a limit order book / Avellaneda M., Stoikov. S. //Quantitative Finance. – 2008. – Т. 8. – №. 3. – С. 217-224.
2. Korolev V. Y. et al. Probability and statistical modeling of information flows in complex financial systems based on high-frequency data //Информатика и ее применения. – 2013. – Т. 7. – №. 1. – С. 12-21.
3. Какойченко А.І. On Analysis of futures markets by Monte Carlo method//XXIV International Conference Problems of decision making under uncertainties (PDMU-2014): Матеріали наук. праць міжнародної конференції. – Cesky Rudolec, 2014. – С. 50.

References

1. AVELLANEDA, M. & STOIKOV, S. (2008) High-frequency trading in a limit order book. *Quantitative Finance*. 8(3). p. 17-224.
2. KOROLEV, V. Y. et al. (2013) Probability and statistical modeling of information flows in complex financial systems based on high-frequency data. *Informatics and Applications*.7(1). p. 12-21.
3. KAKOICHENKO, A. (2014) On Analysis of futures markets by Monte Carlo method. In *XXIV International Conference Problems of decision making under uncertainties*. Cesky Rudolec, Monday 1-th to Friday 5-th September 2014. p. 50.

Надійшла до редколегії 22.10.14