



**межвузовский сборник
научных трудов**

**МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**



2021

Межвузовский сборник научных трудов

**МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

РГРТУ, 2021

Редакционная коллегия

д-р техн. наук, проф., засл. работник высшей школы РФ А.Н. Пылькин (отв. редактор, Рязанский государственный радиотехнический университет); д-р техн. наук, проф., зав. каф. Г.В. Овечкин (Рязанский государственный радиотехнический университет); д-р техн. наук, проф. Е.Е. Ковшов (Московский государственный технологический университет «Станкин»); д-р техн. наук, проф. К.А. Майков (МГТУ им. Н.Э. Баумана); д-р техн. наук, проф. В.В. Белов (Рязанский государственный радиотехнический университет); д-р техн. наук, проф. В.В. Золотарев (Институт космических исследований РАН, г. Москва); д-р техн. наук, проф. И.Ю. Каширин (Рязанский государственный радиотехнический университет); д-р техн. наук, проф. В.Н. Малыш («Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» г. Липецк); И.А. Благодарова (Рязанский государственный радиотехнический университет).

Рецензент: к.т.н., доцент, зав. каф. «Информатики, информационных технологий и защиты информации» Липецкого государственного педагогического института Скуднев Д.М.

Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. А.Н. Пылькина – Рязань: Издательство ИП Коняхин А.В. (Book Jet), июнь 2021. – 84 с.

ISBN 978-5-907400-56-6

Представлены статьи, посвященные различным аспектам разработки программных средств ЭВМ, вычислительных систем и сетей, вопросам математического моделирования, методам обработки информации.

Предназначен для научно-педагогических работников вузов, инженеров, аспирантов и студентов старших курсов.

Авторская позиция и стилистические особенности публикаций полностью сохранены.

ISBN 978-5-907400-56-6

ВВЕДЕНИЕ

Межвузовский сборник научных трудов содержит результаты научных исследований и разработок по следующим направлениям:

- программное обеспечение вычислительных систем, новые информационные технологии;
- прикладная математика, теория информации, искусственный интеллект;
- ЭВМ в системах обработки, управления и обучения;
- автоматизированное проектирование аппаратных средств вычислительных систем;
- информационные технологии в экономических и социальных системах.

Сборник сформирован на основе статей, в которых рассматриваются различные аспекты разработки программных средств ЭВМ, вычислительных систем и сетей, включая вопросы автоматизации проектирования, теории обработки информации и математического моделирования, и предназначен для студентов, аспирантов и преподавателей технических вузов и научных работников.

Материалы для сборника предоставлены сотрудниками и студентами:

- Рязанского государственного радиотехнического университета им. В.Ф. Уткина г. Рязань;

УДК 004.432

Пылькин А.Н., Соколова Ю.С.
СОВРЕМЕННЫЙ PASCALABC.NET,
КАК УДАЧНАЯ ОТЕЧЕСТВЕННАЯ РАЗРАБОТКА
ДЛЯ ИЗУЧЕНИЯ БАЗОВЫХ ПРИНЦИПОВ ПРОГРАММИРОВАНИЯ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

Рассматривается ряд вопросов, связанных с использованием PascalABC.NET при обучении программированию, а также языковые средства современной версии языка Pascal, позволяющие писать короткий, понятный и эффективный код.

В последние годы интерес к IT-индустрии продолжает расти, а перед новичками и преподавателями возникает актуальный вопрос, какой язык программирования стоит изучить для успешного старта.

Выбор усложняется тем, что рейтинги популярности языков программирования ежегодно меняются [1], а самих языков очень много, и каждый из них имеет свои особенности, сферы применения и решает свои задачи. Выбор конкретного языка в профессиональной сфере зависит от цели деятельности. Например, в качестве серверных языков популярны PHP, Java, Ruby. Из клиентских языков наиболее перспективен JavaScript. Для разработки сложных высоконагруженных проектов и игровых приложений требуется C++. Лидерами в разработке приложений для мобильных устройств считаются Java, Objective-C, Swift. Также лидирующее положение занимают C# и Python, которые используются для разработки программ в сфере бизнеса, веб-сервисов, мобильных приложений, игр. Любой опытный программист знает несколько языков, но, как правило, программный код на работе он пишет на одном.

Долгое время базовым для обучения программированию в старших классах и на первых курсах вузов считался разработанный Н. Виртом язык Turbo Pascal, которого впоследствии заменил PascalABC.NET, поскольку он готовил хороший фундамент для изучения языков, используемых в промышленной разработке. На Pascal и C++ начинало программировать большинство разработчиков.

Однако в последние годы фаворитом для изучения основ программирования является Python, который славится своей простотой и лаконичностью. Немногословный и понятный синтаксис, похожий на псевдокод и не усложненный громоздкими конструкциями, а также сильная динамическая типизация способствуют быстрому и безболезненному обучению новичков основам программирования. Кроме того, этот язык постоянно развивается, а сообщество вокруг Python – одно из самых сильных в мире IT.

Вместе с Python развиваются и совершенствуются другие языки программирования, включая PascalABC.NET [2, 3].

PascalABC.NET был создан около 15 лет назад в Южном федеральном университете [3]. Разработчики преследовали цель создания хорошего современного языка программирования для обучения школьников с мощной, хорошо русифицированной средой программирования, поддерживающей технологию IntelliSense, содержащей средства автоформатирования, встроенный отладчик и встроенный дизайнер форм, включающего в себя лучшее, что предлагают другие современные языки, такие как C#, Kotlin, Python, Haskell и другие. В качестве платформы реализации была выбрана платформа Microsoft.NET.

Синтаксически PascalABC.NET близок к языку Delphi, однако включает множество конструкций языка C# и содержит все современные языковые средства: классы, перегрузку операций, интерфейсы, обработку исключений, обобщенные классы и подпрограммы, сборку мусора, лямбда-выражения, средства параллельного программирования [3, 4]. Кроме того, он содержит собственные расширения языка Pascal, нацеленные на написание компактных, интуитивно понятных программ с использованием современных языковых возможностей.

Многое язык PascalABC.NET взял от языка Python. Это прежде всего легковесные языковые и библиотечные средства, обеспечивающие компактность и интуитивную понятность программ. Кроме того, часто код решения задачи, написанный на Python, фактически идентичен коду на PascalABC.NET.

Рассмотрим решение задачи (№ 14 из ЕГЭ по информатике) поиска количества повторений цифры 0 в девятеричной записи числа $N = 729^{77} + 13^{16} - 18$. В обоих случаях будет получено число 214.

<i>Код на PascalABC.NET</i>	<i>Код на Python</i>
<pre>## var N:=729bi**77+13bi**16-18; var k := 0; while N > 0 do begin if N mod 9 = 0 then k += 1; N := N div 9; end; print(k)</pre>	<pre>N = 729**77+13**16-18 k = 0 while N > 0: if N % 9 == 0: k += 1 N = N // 9 print(k)</pre>

В рассмотренном примере для хранения значения N используется неограниченный целый тип, обозначаемый в PascalABC.NET идентификатором `BigInteger`.

Ещё один пример, демонстрирующий схожесть синтаксических конструкций этих языков программирования, также взят из сборников для подготовки к ЕГЭ по информатике (задача № 12): на вход программы для исполнителя «Редактор» поступила строка из 52 подряд идущих комбинаций «AB» (все буквы заглавные, латинские), требуется определить, какая строка получится после выполнения следующего алгоритма (в результате должна получиться строка 'BA'):

```

ПОКА Нашлось(АА) ИЛИ Нашлось(ВВ) ИЛИ Нашлось(АВ)
  Заменить(АА, В)
  Заменить(ВВ, А)
  Заменить(АВ, ВА)
КОНЕЦ ПОКА

```

Код на PascalABC.NET

```

##
var s := 'AB' * 52;
while ('AA' in s) or ('BB' in s) or ('AB' in s) do
  begin
    s := s.replace('AA', 'B', 1);
    s := s.replace('BB', 'A', 1);
    s := s.replace('AB', 'BA', 1);
  end;
print(s)

```

Код на Python

```

s = 'AB' * 52
while ('AA' in s) or ('BB' in s) or ('AB' in s):
    s = s.replace('AA', 'B', 1)
    s = s.replace('BB', 'A', 1)
    s = s.replace('AB', 'BA', 1)
print(s)

```

PascalABC.NET, аналогично Python, позволяет использовать в программах идентификаторы, содержащие кириллические символы [3]. Так в предыдущем примере, реализовав подпрограммы Нашлось() и Заменить(), можно было оставить представленный исполнителю «Редактор» алгоритм фактически без изменений:

```

##
var s: string;
var Нашлось: string → boolean := v → s.Contains(v);
var Заменить: (string, string) → () := (v, w) →

```

```
begin s := s.Replace(v, w, 1) end;  
s:='AB'*52;  
while Нашлось('AA') or Нашлось('BB') or  
    Нашлось('AB') do  
begin  
    Заменить('AA', 'B');  
    Заменить('BB', 'A');  
    Заменить('AB', 'BA');  
end;  
s.Print
```

Как можно заметить из примеров, в коде на PascalABC.NET нет привычных операторов `begin ... end`, разработчики заменили их на символы `##`. Кроме того, использование сочетания символов `###` включает в PascalABC.NET режим спортивного программирования, в котором операторы записываются в краткой форме: вместо `Print` можно писать `Pr`, вместо `Where` – `Wh`, вместо `Count` – `Cnt` и т.п.

Также из последнего листинга видно, что PascalABC.NET работает с кортежами, позволяет писать программы в функциональном стиле с использованием лямбда-выражений, которое легко распознается в тексте программы по характерному знаку операции \rightarrow (произносится «переходит в ... »).

Использование кортежа в качестве выходных параметров у функции позволяет передавать в основную программу сразу несколько значений, например, с помощью функции можно сразу определить сумму и произведение цифр в десятичном числе (пример не является учебным, он демонстрирует работу с кортежами):

```
##  
function SumProduct(n:integer): (integer, integer);  
begin  
    var (s, p) := (0, 1);  
    while n > 0 do  
begin  
    var d := n mod 10;  
    (s, p, n) := (s + d, p * d, n div 10);  
end;  
    result := (s, p)  
end;  
  
var a := ReadInteger('Введите число: ');  
var (s, p) := SumProduct(a);  
s.Println;  
p.Println
```


Модуль `School`, встроенный в современный `PascalABC.NET`, содержит реализацию алгоритмов, часто встречающихся в школьных задачах. Его использование делает текст программы яснее и короче, особенно для тех, кому не требуется углубляться в математические и технические детали этих алгоритмов. Так, с использованием модуля `School` и лямбда-выражений программа нахождения всех простых чисел из диапазона от 0 до 100 будет занимать три строки и иметь следующий синтаксис:

```
##
uses school;
  (1..100).Where(x → x.isPrime).Print
```

В `PascalABC.NET` программное решение задачи нахождения суммы нечётных положительных чисел, ввод которых заканчивается точкой, может быть записано одной строкой:

```
##
ReadSeqIntegerWhile(x → x<>0).Where(x → x.isOdd and (x > 0)).Sum.Print
```

Из приведенных примеров видно, что реализации некоторых функций имеют два формата – для вызова в виде функции и для записи в точечной нотации. Например, для вывода значения переменной `s` можно писать `Println(s)` или `s.Println`.

Но, несмотря на наличие мощной современной системы программирования на базе языка `Pascal`, в учебных заведениях на компьютерах с установленным `PascalABC.NET` продолжают писать код с синтаксисом для `Turbo Pascal` и преподавать устаревшие `Free Pascal` или `Delphi`. Многие не видят разницы между этими языками, полагая, что она минимальна или вообще отсутствует.

Следующий пример демонстрирует различие в кодах, написанных на `Turbo Pascal`, на `PascalABC.NET` и на `Python`, при решении задачи поиска максимума и его позиции в случайно сгенерированном массиве из 10 чисел, каждое из которых лежит в диапазоне от -100 до 100 (если максимумов несколько, определить номер первого).

Код на Turbo Pascal

```
var
  a: array[1..10] of integer;
  max, imax, i: integer;
begin
  randomize;
  for i:=1 to 10 do a[i]:=random(101)-random(101);
```

```
for i:=1 to 10 do write(a[i]:4);
writeln;
max := a[1];
imax := 1;
for i:=1 to 10 do
  if a[i] > max then
    begin
      max := a[i];
      imax := i;
    end;
writeln(max, ' ', imax)
end.
```

Код на PascalABC.NET

```
##
var a := ArrRandom(10, -100, 100);
a.Println;
a.Max.Print;
(a.IndexMax+1).Print;
```

Код на Python

```
from random import randint
a = [randint(-100, 100) for i in range(10)]
print(a)
print(max(a))
print(a.index(max(a)) + 1)
```

Как показывает рассмотренный пример, коды на Turbo Pascal и PascalABC.NET сильно различаются. И, несмотря на то, что в PascalABC.NET можно писать код в стиле старого Turbo Pascal, в нем заложены конструкции, присущие другим языкам программирования. Более того, он умеет понимать, форматировать и выполнять с определенными ограничениями программы, написанные на языке C# [3].

Если вспомнить про язык программирования Python [5, 6], который разрабатывался с целью повышения продуктивности самого программиста, нежели кода, который пишется, то PascalABC.NET также позволяет концентрироваться на сути алгоритма, а не на технических деталях благодаря своим встроенным библиотекам. Но, в отличие от языка Python, язык PascalABC.NET имеет статическую типизацию, что позволяет фиксировать множество ошибок на этапе компиляции и делает программу более безопасной. Кроме того, недостаток в скорости ограничивает область применения Python. Высокопроизводительное приложение на чистом Питоне написать не получится, потребуется прибегать к помощи других языков или использовать статически типизированные расширения. По-

этому, если вы планируете стать всесторонне развитым разработчиком, только знания Python недостаточно.

PascalABC.NET за пределами сферы образования практически не используется, но он имеет множество языковых возможностей, поддерживает множество стилей программирования, поскольку обобщает языковые и библиотечные возможности таких языков, как C#, Python, Kotlin, входящих в TOP-10 рейтинга востребованных языков программирования 2021 г. [1].

Не стоит гнаться за изучением современных языков программирования, поскольку, как показывает практика, на смену одним кодам приходят другие. Гораздо важнее обучать базовым принципам программирования, а этого можно добиться, используя на начальном этапе изучения технологий программирования язык PascalABC.NET. Так, этот язык взят за основу в курсе «Основы программирования» у студентов 1 курса мехмата ЮФУ по направлению «Фундаментальная информатика и информационные технологии» [3]. Объясняется это тем, что PascalABC.NET является мультипарадигменным языком, позволяющим программировать в структурном, объектно-ориентированном и функциональном стилях, а теоретическая и практическая база, полученная при его изучении, позволяет эффективно освоить современные технологии программирования и в очень короткий срок изучать впоследствии такие языки как Python, C++, C#, Java, Haskell, Ruby и др.

PascalABC.NET является удачным решением для начального знакомства с современными технологиями программирования еще и потому, что это программное обеспечение является свободно распространяемой отечественной разработкой с интегрированным электронным задачником Programming Taskbook [7], который предназначен для обучения программированию и содержит учебные задания, охватывающие все основные разделы базового курса программирования: от скалярных типов и управляющих операторов до сложных структур данных и рекурсивных алгоритмов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. <https://habr.com/ru/post/543346/> (дата обращения 20.05.2021).
2. https://pascalabcnet.github.io/mydoc_for_EGE.html (дата обращения 22.05.2021).
3. Осипов А.В. PascalABC.NET: Введение в современное программирование. – Ростов-на-Дону, 2019 – 572 с.
4. Осипов А.В. PascalABC.NET: выбор школьника. Часть 3. / А.В. Осипов. – Ростов-на-Дону, 2020 – 146 с.
5. Лутц, Марк. Изучаем Python, том 1, 5-е изд.: Пер. с англ. – СПб.: ООО «Диалектика», 2019. – 832 с.
6. Лутц, Марк. Изучаем Python, том 2, 5-е изд.: Пер. с англ. – СПб.: ООО «Диалектика», 2020. – 720 с.
7. <http://www.ptaskbook.com/ru/> (дата обращения 28.05.2021).

УДК 681.3.06

Маркова В.В.
РАЗБИЕНИЕ ГРАФА РАЗВИТИЯ МОДЕЛИРУЮЩЕЙ
АВТОМАТИЗИРОВАННОЙ ДЕЛОВОЙ ИГРЫ
НА ЭТАПЫ ПО СМЫСЛУ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

В статье описывается формирование графа развития автоматизированной деловой игры по смысловым понятиям.

Важным аспектом при формировании имитационной модели объекта автоматизированной деловой игры [1, 2] является ее смысловая полнота и достаточность. Для определения смысловой полноты имитационной модели объекта игры необходимо разбиение графа развития игры на этапы по смыслу.

Процесс развития игры – множество элементарных смысловых понятий, на основе которых построена игра.

Обозначим $D = \{D_1, D_2, \dots, D_m\}$ – множество производных или общих смысловых понятий.

Основываясь на объединении смысловых понятий по уровням общности [3], построим граф, у которого вершиной является наиболее общее понятие D_N . Граф строится по уровням общности сверху вниз, т.е. от наиболее общего смыслового понятия к нулевому уровню по всем уровням общности. В уровне D_i смысловое понятие является вершиной и соединяется дугами со смысловыми понятиями, являющимися элементами множества D_i . Таким образом, получим *модель знаний игрока*, представленную в форме графа развития игры. Вершины графа представляют смысловые понятия, изучаемые игроком. Дуги обозначают конкретизацию понятий, т.е. игрок, двигаясь от вершины к вершине, все более и более детально исследует изучаемое смысловое понятие. *Модели конкретного игрока* соответствует трасса графа развития игры.

Обучение осуществляется с помощью выбора наиболее общего смыслового понятия из списка наиболее общих понятий и соответственно цели обучения. Игрок движется по графу развития игры, выбирая нужное понятие из списка предлагаемых на данном уровне обучения смысловых понятий, пока не дойдет до понятия с нулевым уровнем общности. Если данное смысловое понятие будет являться целью обучения, то обучение по наиболее общему понятию закончено, и ветвь с наиболее общим понятием помечается как изученная и отсекается, предоставляя игроку возможность изучить оставшиеся наиболее общие смысловые понятия. Если достигнутое игроком смысловое понятие с нулевым уровнем общности не является целью обучения, то игроку предоставляется возможность вернуться на несколько уровней назад и повторить обучение.

Модель игрока состоит из совокупности следующих множеств:

- множества смысловых понятий, которые необходимо изучить игроку для достижения цели обучения;
- множества смысловых понятий, которые игрок изучил в процессе обучения.

На основе смысловых понятий D_{N-1} уровня общности строится граф развития игры. *Процесс формирования графа развития игры* осуществляется следующим образом.

1. Выбирается область игры, т.е. главное смысловое понятие.
2. Определяется цель игры, которую необходимо достигнуть.
3. Выделяются основные этапы процесса игры.
4. Для каждого этапа определяются смысловые понятия, близкие к данному этапу.
5. На основе выбранных понятий и их взаимосвязей для каждого этапа строится ориентированный подграф развития игры.

Таким образом, осуществляется разбиение графа развития игры на этапы по смыслу для его дальнейшего разбиения на этапы по структуре [1].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Маркова В.В. Обобщенная схема проектирования моделирующей автоматизированной деловой игры по работе в системе 1С:Предприятие. Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. А.Н. Пылькина – М.: Горячая линия-Телеком, 2015. 77 с.

2. Маркова В.В., Баринов И.В. Разработка моделирующей автоматизированной деловой игры по работе в системе 1С:Предприятие. Вопросы образования и науки в XXI веке : сборник научных трудов по материалам Международной научно-практической конференции 29 апреля 2013 г.: в 11 частях. Часть 5; М-во обр. и науки РФ. Тамбов: Изд-во ТРОО «Бизнес-Наука-Общество», 2013. 163 с.

3. Маркова В.В., Баринов В.В. Представление сценария моделирующей автоматизированной деловой игры по работе в системе 1С:Предприятие в виде полного графа игры. Программные информационные системы: Межвуз. сб. науч. тр. / Под ред. А.Н. Пылькина – Рязань: РГРТУ, 2010. 152 с.

УДК 004.85

Головкин Н.В., Цуканова Н.И.

ГЕНЕРАЦИЯ ИЗОБРАЖЕНИЙ КОШАЧЬИХ МОРДОЧЕК
С ПОМОЩЬЮ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНОЙ
НЕЙРОННОЙ СЕТИ DCGAN

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

В статье рассматривается генерация изображений кошачьих мордочек с помощью генеративно-состязательной нейронной сети.

Постановка задачи. Необходимо создать и обучить генеративно-состязательную сеть (в дальнейшем – GAN) для генерации изображений кошачьих мордочек. В качестве набора данных для обучения нейронной сети использовался набор данных с сайта Kaggle. Он состоит из RGB изображений кошачьих мордочек 64 x 64 пикселей (рис. 1). Всего в наборе приблизительно 16 тысяч изображений.

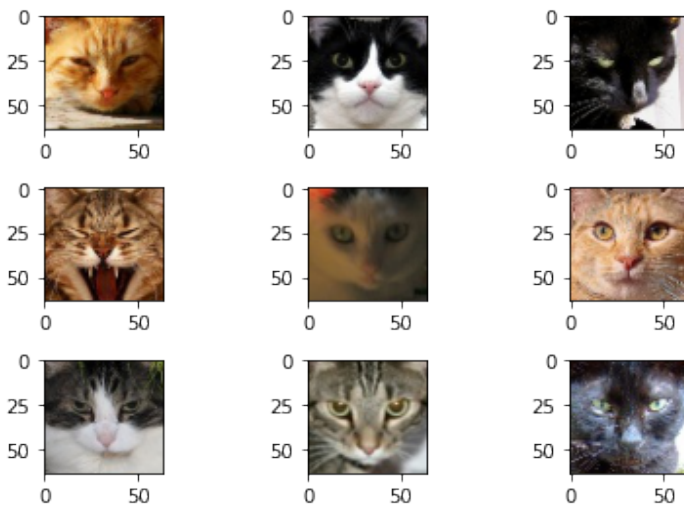


Рис. 1. Изображения кошачьих мордочек, взятые из обучающего набора данных (реальные изображения)

Генеративно-состязательная сеть (ГАН) состоит из 2-х нейронных сетей, взаимодействующих друг с другом: генератора и дискриминатора.

Сеть-генератор – это сеть, которая берет вектор случайных чисел (произвольная точка в скрытом пространстве) и декодирует его в нужное представление (в данном случае – изображение). В программе на языке Python для генератора используется **сверточная сеть**. Для всех сверточных слоев, кроме первого и последнего, используется активационная

функция Leaky ReLU. Считается, что данная функция предотвращает «смерть» градиента.



Рис. 2. Изображения кошачьих мордочек, полученные генератором (фейковые изображения)

Сеть-дискриминатор – это сеть, на вход которой поступают изображения с выхода генератора, в результате она должна определять, являются ли они реальными или сгенерированными. Для всех сверточных слоев, кроме последнего, используется активационная функция Leaky ReLU. Последний слой обладает активационной функцией sigmoid (на выходе число в интервале $[0; 1]$), которое может быть проинтерпретировано как вероятность того, что данные реальные, а не сгенерированные. Чтобы получить всю GAN создается новая модель и сначала к ней добавляется генератор, а затем дискриминатор. Дискриминатор и генератор обучаются отдельно друг от друга.

Для GAN и дискриминатора при обучении используется функция потерь – бинарная кроссэнтропия, а в качестве оптимизатора используется Adam. Скорость обучения дискриминатора задана меньше, чем у генератора. Это сделано, чтобы компенсировать тот факт, что дискриминатор слишком быстро обучается отличать реальные изображения от сгенерированных. Если генератор или дискриминатор будут превосходить друг друга, то сеть не сойдется. Значение потерь одной части GAN будет значительно больше, чем у другой. При обучении стоит задача нахождения точки равновесия Нэша. По сути, обучение GAN – это минимаксная игра, где выигрыш одного игрока равен проигрышу другого. Обучение нейронных сетей это циклический процесс, на каждом шаге которого выполня-

ются следующие действия. 1) Обучается дискриминатор на реальных данных. 2) Обучается дискриминатор на сгенерированных данных. 3) Рассчитывается значение потерь для дискриминатора. 4) Обучается генератор в составе GAN при замороженном дискриминаторе.

Для контроля через заданное число шагов выводится значение потерь генератора, дискриминатора и те изображения, которые на данный момент может создавать генератор. Каждые 1000 шагов происходит сохранение весов модели.

На рис. 2 показаны кошачьи мордочки, полностью созданные уже обученным генератором на основе вектора шума. Результат замечательный: генератор научился создавать изображения очень похожие на реальных кошек.

УДК 004.42

Ромашкова В.В.

ЭТАПЫ РЕАЛИЗАЦИИ АЛГОРИТМА ПРИНЯТИЯ РЕШЕНИЯ НА ОСНОВЕ ФУНДАМЕНТАЛЬНОГО АНАЛИЗА ПРИ ИНВЕСТИРОВАНИИ НА ФОНДОВОМ РЫНКЕ

Рязанский государственный радиотехнический университет им. В.Ф.Уткина, Рязань

В статье обоснована актуальность инвестирования на фондовом рынке, описаны ступени фундаментального анализа, приводятся этапы реализации метода принятия решения.

Фондовый рынок – это отличный способ выйти на пенсию раньше, чем наступит пенсионный возраст по законодательству. Мало кто откажется иметь пассивный доход и жить на дивиденды. Однако для этого необходимо накопить капитал и вложить его в прибыльный бизнес. Следует тщательно выбирать отрасли и компании, чтобы грамотно вложить и не потерять накопления. Для принятия решения по долгосрочным вложениям часто применяют фундаментальный анализ – прогнозирование стоимости компании и цены ее акций на основе анализа финансовых и производственных показателей ее деятельности.

На сегодняшний день в открытом доступе нет программного обеспечения, которое смогло бы прогнозировать цену акции и сравнивать компании между собой на основе фундаментального анализа.

Из существующих аналогов в открытом доступе можно найти сайты, которые только предоставляют несколько рассчитанных фундаментальных показателей компании. Сервисы отличаются между собой популярностью, ценой подписки и данными, которые они предоставляют. Например, сервис investors.com имеет цену 20 \$ в месяц. Недостаток в том, что процесс анализа, сравнения и выводов делают люди. Подобные сервисы являются помощниками для сбора информации.

Таким образом, автоматизации принятия решения на основе фундаментального анализа очень актуальна.

Для автоматизации процесса сравнения компаний и присуждения ей рейтинга для принятия решения, необходимо лучше изучить понятие фундаментального анализа.

Фундаментальный анализ делится он на три ступени.

Первая ступень – макроэкономические показатели, которые включают в себя динамику валового внутреннего продукта страны, где находится компания, базовые ставки центральных банков, уровень безработицы, новости.

Вторая ступень – отраслевой анализ, который включает в себя анализ состояния отрасли. Например, рост объемов производства, рост цен на факторы производства, скорость оборота капитала.

Третья ступень – показатели компании, а именно соотношение цена/прибыль (мультипликатор P/E) и рентабельность (мультипликатор ROE).

Прежде чем проводить сравнение компаний на основе фундаментального анализа, необходимо получить данные для сравнения. Наиболее распространённый и эффективный способ получения данных из различных открытых веб-сайтов для обработки – это парсинг.

Для сравнения компаний по показателям будут использованы следующие критерии:

- критерий азартного игрока,
- критерий Вальда,
- критерий Гурвица,
- критерий Севиджа,
- критерий Баеса,
- критерий Лаплас,
- критерий Ходжеса-Лемана.

Часть критериев нацелена на максимальный выигрыш, такие критерии будут давать максимальную оценку компании с максимально возможным доходом для инвестора. Другая часть критериев наоборот нацелена на минимальный проигрыш и для таких критериев чем меньше проигрыш, тем большую оценку получит компания, а инвестор потерпит минимальные убытки. Остальная часть критериев рассчитана на средний выигрыш и средний проигрыш.

Так как все критерии различаются между собой, нам необходимо выбрать каким критериям доверять. Для максимальной выгоды инвестора необходимо решить каким критериям доверять больше, а каким меньше и в какой момент времени.

Например, будем сравнивать акции компаний из одной отрасли. Если отрасль и ее показатели растут, то логично предположить, что акции всех компании в данной отрасли так или иначе будут расти. Исходя из этого,

минимизировать проигрыш было бы не выгодно. На мой взгляд, в случае с растущей отраслью лучше прислушаться к критериям, которые направлены на максимальный выигрыш.

И наоборот, если отрасль и ее показатели падают, то и котировки акций компаний будут падать. В таком случае лучше прислушаться к критериям, которые направлены на минимизацию проигрыша.

Если отрасль и ее показатели стабильны на протяжении времени и не имеют восходящего или нисходящего тренда, то следует обращать внимание на критерии, которые усредняют выигрыш и проигрыш. При таком подходе возможно достичь максимально выгодного вложения собственных средств.

Исходя из вышесказанного, для оценки доверия к критериям будем использовать показатели второй степени фундаментального анализа, т.е. показатели отрасли. Зная оценку доверия возможно использовать все критерии, склоняясь к одному или нескольким.

Далее необходимо сравнить компании между собой. Для это используем показатели первой и третьей ступеней фундаментального анализа. Например, компании схожи по показателям третьей ступени, тогда требуется сравнить их по показателям первой ступени. После всех сравнений необходимо составить рейтинг компании. В основу рейтинга возьмем ранговый метод, который говорит, что чем лучше тот или иной показатель в сравнение с другими компаниями, тем выше ранг. Например, среди всех компаний по критерию № 1 лидирует компания № 5, тогда ранг у данной компании у конкретного показателя будет равен 1. Необходимо присвоить ранги всем показателям всех сравниваемых компаний после чего сложить ранги. Предпочтительнее будет считаться компания с наименьшей суммой, то есть наименьшим рангом.

В дальнейшем будут описаны технологии, которые планируется применить для реализации описанного алгоритма.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Боровкова В.А., Боровкова В.А. Рынок ценных бумаг. 2-е изд., обновленное и дополненное. – СПб.: Питер, 2008. – 400 с.: ил. – (Серия «Учебное пособие»), ISBN 978-5-91180-465-7.

УДК 004.85

Шурыгина О.В., Цуканова Н.И.

ПРЕОБРАЗОВАНИЕ ИЗОБРАЖЕНИЯ В ИЗОБРАЖЕНИЕ
С ПОМОЩЬЮ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫХ НЕЙРОННЫХ СЕТЕЙ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

В статье рассматривается задача применения генеративно-состязательной нейронной сети к преобразованию изображения объекта из одного вида в изображение другого вида, например снимок – в карту, вечернее изображение в дневное и т.п.

Постановка задачи. Необходимо создать и обучить генеративно-состязательную сеть для преобразования изображения в его схему и наоборот. Набор данных для обучения взят с сайта Kaggle.

Разработанная модель была обучена для 4 случаев: 1) Преобразование спутниковой карты в схему; 2) Преобразование схемы в спутниковую карту; 3) Преобразование фото дороги в схему; 4) Преобразование схемы в фото дороги.

Программа на языке Python реализована в виде 5-ти ноутбуков. Четыре содержат в себе модель генеративно-состязательной сети и процесс ее обучения на своем наборе данных. Каждая предназначена для решения одной из вышеперечисленных задач и содержит описание архитектуры генеративно-состязательной нейронной сети, процесса ее обучения и средств сохранения промежуточных результатов, таких как модели нейронных сетей, а также изображений, получаемых генератором в текущий момент времени.

Пятый ноутбук посвящен эксплуатации разработанных моделей. На вход каждой из них подаются новые данные (новое изображение), обученный генератор преобразует его в изображение соответствующего типа, которое потом можно просмотреть. Далее на рисунках приведены результаты работы GAN для каждого из 4-х случаев.



Рис. 1. Преобразование спутниковой карты в схему местности

Для решения каждой из 4-х задач загружаем одну из сохраненных моделей и соответствующий набор данных. Генерируем изображение и выводим результат. Он содержит 3 снимка: левый снимок (Source) – исходное изображение, снимок в центре (Generated) – схема объекта на изображении, которую создал генератор, снимок справа (Expected) – схема, созданная человеком, по которой генератор и дискриминатор учились решать свои задачи.



Рис. 2. Преобразование схемы в спутниковую карту
(Обратное преобразование)

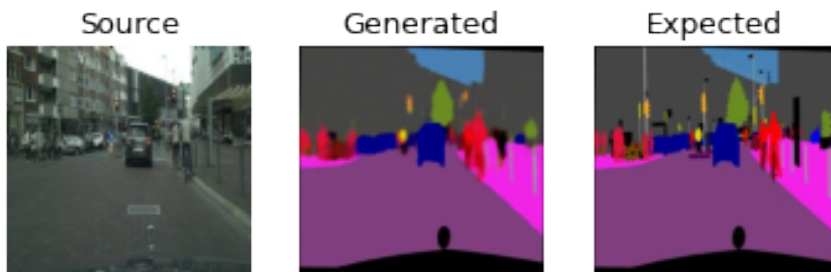


Рис. 3. Преобразование фото дороги в схему

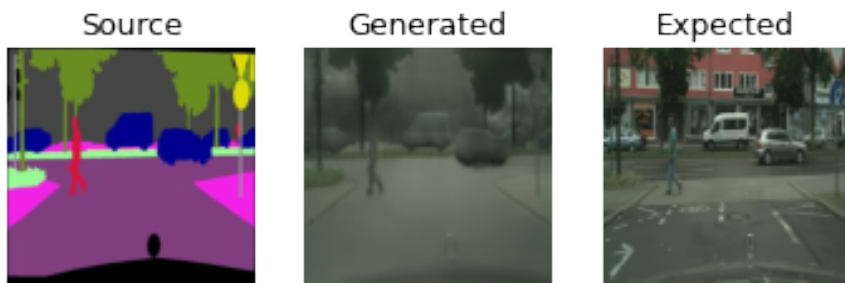


Рис. 4. Преобразование схемы в фото дороги

Снимки показывают, что генератор дает правдоподобные результаты, выделяет объекты, в некоторых случаях сгенерированный снимок почти подобен реальному. Хочется обратить внимание читателя на то, что в этом примере генератор создает новое изображение не из шума, а из другого изображения.

УДК 004.42

Перевезенцев Е.Е.

МЕТОД ПРОГНОЗИРОВАНИЯ ЦЕН НА ФОНДОВОМ РЫНКЕ

Рязанский государственный радиотехнический университет имени В.Ф.Уткина, Рязань

В статье обоснована необходимость автоматизации метода технического анализа для прогнозирования цен на фондовом рынке и описаны задачи, которые потребуются решить.

На сегодняшний день многие страны живут при рыночной экономике. Рынок – это система экономических отношений между продавцом и покупателем, которая позволяет совершать обмен необходимыми ресурсами. Рынок ценных бумаг или фондовый рынок – это совокупность экономических отношений между его участниками по поводу выпуска и обращения ценных бумаг.

Цена – это основная характеристика товара с точки зрения рыночной экономики. Цену на рынке определяет спрос. Спрос определяют люди, следовательно, можем говорить о том, что цена – это ожидания людей. Для анализа и предсказания цены используют различные методы. Один из современных и универсальных методов – это технический анализ – совокупность инструментов прогнозирования вероятного изменения цены на основе закономерностей изменений цены в прошлом в аналогичных обстоятельствах. Данный метод подразумевает три правила:

- все факторы, влияющие на цену, уже заложены в график;
- цена всегда движется в трендах;
- история повторяется.

На основе этих правил люди предсказывают направление и динамику цен на фондовом рынке. Технический анализ можно использовать как математическую модель, временной ряд или статистический метод. Почти все вычисления уже лежат на плечах вычислительных машин. В открытом доступе существуют программные обеспечения, которые предоставляют инструменты для вычисления на основе статистических методов и временных рядов, также есть возможность отображения графиков цен. Результат анализа данных, а именно прогноз цены, по-прежнему делает человек. Исключением являются нейронных сетей вида GRU, которые предсказывают цены на основе истории. Данный вид нейронных сетей имеет множество ограничений. На данный момент обучить ее под все рынки практически невозможно. Фондовый рынок – это очень перспективное направление, так как большая часть ежедневных траты населения обращает

ется в бизнесе, крупнейшие из которых представлены на биржах. На основе этого, создание и разработка метода, который сможет предсказывать динамику цены и тренды очень актуальны. Данный метод имеет несколько преимуществ.

Во-первых, если убрать человека из процесса предсказания цен на основе технического анализа, то можно избежать ошибок, связанных с верой во что-либо. Например, инвестор, используя метод технического анализа, спрогнозировал рост цены. Однако, он мог не учесть часть факторов, придать графику свой смысл и убедить себя в желаемом, что повышает вероятность ошибки.

Во-вторых, вычислительные машины могут «запомнить» больше, чем человек, это необходимо для анализа на основе прошлого.

Для решения проблемы прогнозирования цен на основе технического анализа можно применить математическую модель, усовершенствовав ее с помощью паттернов и функций зависимости между паттернами. В техническом анализе паттерны – это устойчивые повторяющиеся сочетания данных цены, объёма или индикаторов. Например, при условии снижения цены и приближения ее к своему минимуму за период требуется спрогнозировать момент роста цены. Исходя из паттернов этот момент наступит тогда, когда количество заявок на покупку превысит количество заявок на продажу.

Чтобы использовать паттерны, их необходимо добавить в программу. Для создания паттерна необходимо при заданном интервале вычислить разницу между двумя соседними точками на графике, преобразовать и сопоставить с паттернами, которые будут добавлены в программу заранее вручную.

Для реализации описанного выше метода необходимо получать график цены в реальном времени. Далее данные графика следует преобразовывать в числовой вид по временным интервалам. Важной задачей является масштабирование из коротких временных интервалов, таких как 5-10 минут, в более длинные, такие как 1 год, для возможности отображения графиков разного масштаба.

Также важными задачами являются реализация строки поиска ценной бумаги, вывода графика цены акции компании.

Подробный процесс реализации описанного выше программного обеспечения будет представлен в дальнейших научных работах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Вайсман Р. Механические торговые системы: Психология трейдинга и технический анализ / Ричард Вайсман; Пер. с англ. – М.: Альпина Паблицерз, 2011. – 229 с. ISBN 978-5-9614-1597-1.

УДК 004

Проказникова Е.Н., Калугин Д.И.

ПРИМЕНЕНИЕ РАЗЛИЧНЫХ ПРИКЛАДНЫХ МЕТОДОВ
ДЛЯ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ РЕЗУЛЬТАТОВ
БЕЙСБОЛЬНЫХ МАТЧЕЙ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

В настоящее время все большее количество людей интересуется спортивными играми и является фанатами той или иной спортивной игры, в которой соревнуется пара команд. Одним из наиболее острых вопросов в этом случае является вопрос о выигрыше или проигрыше любимой команды. Поэтому различные системы прогнозирования результатов командных игр становятся все более популярными. Это определяет актуальность разработки программного обеспечения, позволяющего с той или иной вероятностью определять исход игры.

Для разработки подобных систем используются системы искусственного интеллекта (например, методы нейросетевой кластеризации), машины нечеткого логического вывода, а также применим метод Монте-Карло.

В случае использования систем искусственного интеллекта задачу кластеризации рассматривают следующим образом.

Кластеризация – это задача распределения большого массива имеющихся данных по группам (кластерам) на основе общих или, в случае их количественного измерения, близких по величине признаков. Таким образом, отдельно взятые вектора данных (примеры), сформировавшие в результате работы искусственной нейронной сети кластер, во-первых, должны быть действительно похожи между собой, а во-вторых, сами и определяют свойства образованного ими кластера.

Кластерный анализ – процедура описания качественных и количественных свойств кластеров. Полученное описание ценно само по себе, но также может быть использовано для последующей работы с уже обученной нейронной сетью, когда выполняется отнесение нового примера к одному из уже существующих кластеров. Если процесс настройки сети и процедура кластерного анализа были выполнены качественно, на достаточно большом объеме корректных обучающих данных, можно с очень высокой долей уверенности утверждать, что описание соответствующего кластера будет соответствовать и этому новому примеру [1].

Решая задачу прогнозирования результата матча, нужно знать, что ее ответом никогда не должно быть предсказание единственного исхода. Вопрос лишь в частоте реализации того или иного события. Таким образом, в результате решения задачи должны быть получены вероятности

каждого из возможных исходов игры, и их сумма должна быть строго равна единице [1].

Нейронные сети самообучаются на специально составленных для этого выборках данных. Предпочтителен вариант самоорганизации нейросетей, когда в процессе обучения не только настраиваются весовые коэффициенты, но формируется сама структуры сети: образуются новые кластеры, если количественные и качественные характеристики примера не подходят достаточно близко ни к одному из имеющихся кластеров, а также возможно удаление кластера (нейрона), если в течение текущей эпохи обучения он не был задействован.

Решение же конкретной задачи прогнозирования исхода подачи можно представить в виде последовательности следующих этапов:

- сбор исходных данных об имевших место спортивных событиях того же вида спорта и ранга;
- выбор архитектуры нейронной сети для решения задачи;
- выбор или разработка программного продукта для решения задачи;
- формирование обучающей выборки и ее структуризация под требования используемого программного обеспечения;
- настройка алгоритма обучения;
- обучение и кластерный анализ;
- практическое использование обученной нейронной сети для прогнозирования вероятностей возможных исходов.

В случае использования машин нечеткого логического вывода задача прогнозирования результатов бейсбольной подачи сводится к поиску функционального отображения вида:

$$X = \{x_1, x_2, \dots, x_n\} \rightarrow D \in \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, d_{11}, d_{12}, d_{13}\},$$

где X – вектор влияющих факторов, которыми могут быть уровень мастерства команды в атаке, уровень мастерства команды в защите, стадион, на котором проходит матч и т. д.; D – результат подачи, который оценивается терминами: d_1 – хит-бай-питч, d_2 – страйк, d_3 – болл, d_4 – фол, d_5 – сингл, d_6 – дабл, d_7 – трипл, d_8 – хоум-ран, d_9 – флай-аут, d_{10} – граундаут, d_{11} – попаут, d_{12} – дабл-плей, d_{13} – сэк-флай.

Определим факторы, которые влияют на результат подачи:

- x_1 – мастерство команды в атаке;
- x_2 – мастерство команды в защите;
- x_3 – стадион, на котором проходит матч;
- x_4 – количество занятых баз в текущий момент;
- x_5 – количество подач, сделанных питчером в данный момент;
- x_6 – мастерство текущего питчера;
- x_7 – количество ударов (хитов), сделанных в этом иннинге;

Под методом Монте-Карло понимается численный метод решения математических задач при помощи моделирования случайных величин [3].

Применительно к системе прогнозирования результатов бейсбольных матчей, необходимо моделировать исходы каждой подачи.

Чтобы определить результат каждой подачи, необходимо получить шесть случайных чисел. Каждое из чисел определяет конкретное событие из группы:

x_1 – попадание мяча в страйк-зону (мяч попал в зону, мяч пролетел мимо зоны, мяч попал в игрока с битой);

x_2 – реакция игрока на летящий мяч – игрок взмахнул битой, игрок не взмахнул битой;

x_3 – попадание битой по мячу – игрок ударил битой по мячу, игрок промахнулся;

x_4 – сила удара по мячу – игрок выбил фолл-бол, сингл, дабл, трипл или хоум-ран;

x_5 – выведение игрока в аут – игрок не выведен в аут, игрок получил флай-аут, попаут, граундаут;

x_6 – дополнительные условия – в аут выведены сразу два игрока нападения, игрок на второй или третьей базе перебежал на следующую базу (сэк-флай).

Для моделирования исходов бейсбольных подач необходимо 14 основных коэффициентов, определяющих вероятность каждого из событий и 6 дополнительных коэффициентов, корректирующих вероятности появления основных событий. Дополнительными коэффициентами являются:

k_1 – стадион, на котором проводится матч;

k_2 – номер бэттера в линейке отбивающих;

k_3 – количество ударов (хитов), сделанных в этом иннинге;

k_4 – количество подач, сделанных питчером в данный момент;

k_5 – количество занятых баз в текущий момент;

k_6 – мастерство текущего питчера.

Рассмотрим основные преимущества и недостатки каждого из методов:

Основным преимуществом методов нейросетевой кластеризации является возможность самоорганизации нейросети, когда в процессе обучения формируется структура сети.

К недостаткам данной группы методов можно отнести большие временные затраты. Также результаты, полученные в процессе работы нейросети, труднее объяснить.

Преимуществом использования машины нечеткого логического вывода является резкое уменьшение числа производимых операций, что в свою очередь приводит к увеличению быстродействия. Недостатком является отсутствие стандартной методики конструирования таких систем. Также переход от вероятностного подхода к нечеткому не увеличивает точность прогнозов.

Метод Монте-Карло является наиболее простым для освоения, его проще всего модернизировать и отлаживать. Но точность данного метода является небольшой.

Вывод. В данной статье рассмотрены три метода прогнозирования: группа методов нейросетевой кластеризации, машина нечеткого логического вывода, метод Монте-Карло. В разрабатываемой системе будет использован метод Монте-Карло. Этот метод выбран из-за своей доступности для освоения и настройки.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Прогнозирование исходов спортивных игр методами нейросетевой кластеризации: [Электронный ресурс] // Искусственный интеллект, искусственный разум, системы искусственного интеллекта. URL: <https://neuro-nus.com/stat/205-prognosis-sport-neural-cluster.html> (Дата обращения: 02.04.2021).

2. Штовба С.Д., Вивдюк В.В. Прогнозирование результатов футбольных матчей на основе нечетких правил // Вестник молодых ученых. Серия: Экономические науки. – 2002. – №6. – С.54-60.

3. Метод Монте-Карло и его точность [Электронный ресурс] // Хабр. URL: <https://habr.com/ru/post/274975/> (Дата обращения: 06.05.2021).

УДК 004.9

Дмитриева Т.А., Чевагин В.В., Благодаров Е.А.

ОБЗОР АНАЛОГОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОРГАНИЗАТОРА ПРЕПОДАВАТЕЛЯ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

Рассматриваются аналоги программного обеспечения организатора преподавателя, приводятся их достоинства и недостатки.

Несомненно, актуальной является задача автоматизации ведения преподавателем различных записей, связанных с учебным процессом, таких как посещаемость студентов, защита лабораторных работ, информация о сдаче студентами зачетов, экзаменов и прочее. Поэтому разработка программного обеспечения, предоставляющего преподавателя возможности организатора, практически значима. Удобнее данное программное обеспечение реализовать в виде мобильного приложения, так как мобильное устройство часто находится в зоне доступа преподавателя, значит информация в организатор будет заноситься в текущем моменте времени, поэтому она будет всегда актуальной.

Рассмотрим существующие аналоги данного программного обеспечения.

Leader Task. Один из наиболее распространенных многофункциональных организаторов, позиционируется как удобный инструмент для управления делами (рис. 1).

LeaderTask [1] позволяет формировать, систематизировать и сортировать списки задач, хранить и структурировать информацию и осуществлять календарное планирование.

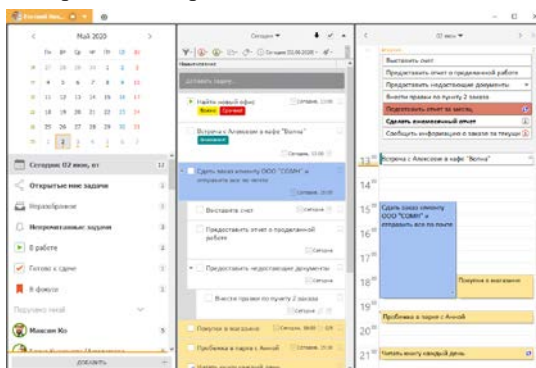


Рис. 1. Сервис для управления делами и поручениями LeaderTask

2. School Calendar. Универсальный календарь для преподавателей и учеников средних школ и колледжей [1]. School Calendar позволяет составлять расписание, добавлять задания и напоминания, вести дневник (рис. 2).



Рис. 2. Программа School Calendar

Возможно использование календаря не только для одного ученика, но и для целого класса или для преподавательского состава. Позволяет собирать и систематизировать подробную информацию об учениках и

преподавателей. Есть несколько вариантов представления календаря на экране, можно менять его цвет.

3. Прозрачный органайзер. Органайзер с необычным дизайном (прозрачным окном), находящийся прямо на рабочем столе Windows и не затрудняющий доступ к ярлыкам и файлам, находящимся на рабочем столе [1].



Рис. 3. Прозрачный органайзер

У программы имеются широкие возможности по цветовым настройкам интерфейса, позволяющие подобрать его к любой картинке рабочего стола.

4. Органайзер учителя 2 Pro. «Органайзер учителя» (рис. 4) – это инструмент, позволяющий облегчить труд преподавателя путем сведения большинства необходимых записей в одном приложении [2].

19.10.2015	Домашняя	У доски	Сумма	За занятия	
Козлов Михаил	10	8	18	5	
Смирнов Александр	10	7	17		
Соколов Артем			0	н	

Рис. 4. Органайзер учителя 2 Pro

5. Органайзер учителя 0.12.12 beta. Содержит журнал оценок (рис. 5) и посещаемости, заметки, расписание преподавателя ВУЗа, является унифицированным информационным ресурсом преподавателя [3].

Расписание							
	20 МАР ПН	21 МАР ВТ	22 МАР СР	23 МАР ЧТ	24 МАР ПТ	25 МАР СБ	26 МАР ВС
Менюшка недели							
№	Время	Тип	Название события	Группы	Место		
1	00:26 -01:27	лекция 10	Дисциплина 10	Группа 10	A10		
2	05:47 -06:48	лекция 2	Дисциплина 2	Группа 2	A2		
3	07:41 -08:42	лекция 8	Дисциплина 8	Группа 8	A8		
4	08:11 -09:12	лекция 1	Дисциплина 1	Группа 1	A1		
5	09:40 -10:41	лекция 3	Дисциплина 3	Группа 3	A3		
6	09:41 -10:42	лекция 9	Дисциплина 9	Группа 9	A9		
7	12:05 -13:06	лекция 11	Дисциплина 11	Группа 11	A11		
8	12:13 -13:14	лекция 6	Дисциплина 6	Группа 6	A6		
9	16:16 -17:17	лекция 12	Дисциплина 12	Группа 12	A12		
10	16:26 -17:27	лекция 5	Дисциплина 5	Группа 5	A5		
11	17:34 -18:35	лекция 4	Дисциплина 4	Группа 4	A4		
12	23:24 -00:25	лекция 7	Дисциплина 7	Группа 7	A7		

Рис. 5. Органайзер учителя 0.12.12 beta

6. TeacherKit. TeacherKit [4] не ограничивает свою аудиторию пользователей только учителями, возможности гибкой настройки позволяют использовать его также преподавателями колледжей и ВУЗов, аспирантами, занимающимися преподавательской практикой, специалистами компаний и т.д. (рис. 6).

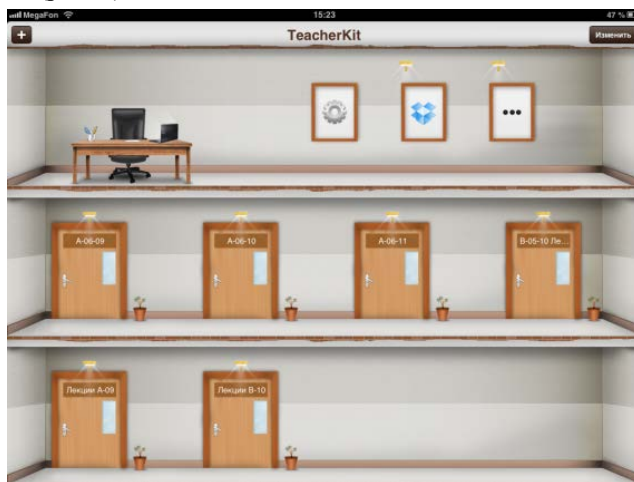


Рис. 6. Программа TeacherKit

На основе информации о данных программных продуктах можно составить таблицу сравнения (табл. 1).

Таблица. Сравнение организеров преподавателя

	LeaderTask	School Calendar	Прозрачный организер	Организер учителя 2 Pro	Организер учителя 0.12.12 beta	TeacherKit
Авторизация	+	+	-	-	-	+
Мобильная версия	+	+	-	+	+	+
Есть версия для IOS	+	+	-	-	-	+
Есть версия для Android	+	+	-	+	+	+
Есть версия для PC	+	-	-	-	+	-
Добавление обычных заметок	+	+	+	+	+	+
Добавление календарных заметок	-	+	+	-	+	+
Добавление напоминаний	-	+	+	+	-	+
Расписание поддерживает двухнедельный цикл (числитель\знаменатель)	+	+	-	+	-	+
Защита паролем	-	+	-	-	-	-
Есть многопользовательский режим	+	-	-	-	-	+
Экспорт информации	-	+	-	-	-	+
Универсальность (организер может применяться в работе любым преподавателем ВУЗа или учителем школы)	+	-	-	-	-	+
Независимость (нет необходимости в подключении к серверу, все данные хранятся у пользователя)	-	-	+	-	-	+

Обновляемость (разработчик выпускает обновления хотя бы раз в месяц)	+	+	–	–	–	+
Возможность отмечать отсутствующих студентов/учеников	+	+	–	+	–	+
Редактирование расписания	+	+	–	+	+	+
Идентификация студента/ученика с помощью фото	–	–	–	+	–	+

Можно заметить, что ни одно программное обеспечение не удовлетворяет необходимым требованиям, некоторые из рассмотренных программ имеют избыточный функционал, многие имеют специфику работы, подходящую для учителя, но не для преподавателя. Достоинства представленных органайзеров следует учесть при разработке собственного мобильного приложения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Какой органайзер нужен учителю? – URL: <http://didaktor.ru/kakoj-organajzer-nuzhen-uchitelju> (дата обращения: 15.05.2021).
2. Органайзер учителя 2 Pro. – URL: <https://www.apkandroid.ru/teachers-organizer-2-pro/com.sorokinkirill.teacherorganizer.redefining> (дата обращения: 28.05.2021).
3. Органайзер учителя 0.12.12 beta. – URL: <https://apk-dl.com/Органайзер-учителя-beta/com.sorokinkirill.teacherorganizer> (дата обращения: 01.06.2021).
4. Обзор TeacherKit. Удобный органайзер для преподавателей. – URL: <https://www.ferra.ru/review/apps/obzor-teacherkit-udobnyu-organayzer-dlya-prepodovateley.htm> (дата обращения: 06.06.2021).

УДК 004

Журавлев П.В.

УЧЁНЫЙ, ИМЯ КОТОРОГО НОСИТ НАШ УНИВЕРСИТЕТ!

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань



Уткин Владимир Федорович (1923-2000), конструктор ракетной и военно-космической техники, дважды Герой Социалистического Труда, уроженец местечка Пустобор Касимовского уезда (ныне Касимовский район).

В.Ф. Уткин родился 17 октября 1923 г. в местечке Пустобор Касимовского уезда (ныне на территории Касимовского района). Отец работал плановиком-экономистом на чугунолитейном заводе, мать занималась домашним хозяйством и воспитанием четырех сыновей.

В 1932 г. пошел в начальную школу в поселке Лашма, куда к этому времени переехала семья, в 1941 г. Уткин с отличием окончил среднюю школу № 2 г. Касимова и стал курсантом Ульяновского училища связи. Во время Великой Отечественной войны Владимир Федорович служил старшим механиком военно-телеграфной станции в 2787-й истребительной авиационной дивизии, дошел до Берлина.

После демобилизации Уткин работал в ремесленном училище № 5 при Лашманском чугунолитейном заводе. С 1946 г. – студент факультета реактивного вооружения Ленинградского военно-механического института (ныне Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова – БГТУ «ВОЕНМЕХ»). Весной 1952 г. он защитил диплом и получил распределение в ОКБ-586 г. Днепропетровска, где проработал 38 лет.

Как уже отмечалось выше, большую часть жизни Владимир Федорович Уткин посвятил работе в конструкторском бюро «Южное» на машиностроительном заводе в Днепропетровске, куда попал по распределению сразу после окончания вуза. Этот завод строили всей страной, а бюро считалось одним из ведущих в Союзе, поэтому такое распределение можно было считать большой удачей. Изначально планировалось, что предприятие будет выпускать автомобили, но в тот момент в Советском Союзе набирала обороты космическая кампания, и заводские мощности руководство решило использовать для производства ракет.

Профессиональные достижения

Уткин Владимир Федорович начинал свой трудовой путь в бюро инженером-конструктором, а продолжал – старшим инженером, начальни-

ком группы, замначальника отдела, заместителем главного конструктора и, наконец, гендиректором КБ «Южное». С 1954 по 1960 г. – начальник группы, сектора, отдела, с 1960 г. – заместитель начальника и главного конструктора КБ «Южное». За время своей работы в конструкторском бюро Уткин проявил себя талантливым ученым и мудрым руководителем, умевшим находить альтернативные научно-технические решения при минимальных ресурсно-временных затратах. Эта стратегия была основной в деятельности Владимира Федоровича. В период с 1971 по 1990 г. – главный, затем – Генеральный конструктор КБ «Южное». Под его руководством создано 4 поколения стратегических ракетных комплексов, более 80 типов спутников военного и научного назначения, выведено на орбиты более 300 космических аппаратов, создано несколько типов ракетопосылателей. При непосредственном участии Владимира Федоровича разработаны и сданы на вооружение большинство типов ракет «СС74», «СС79», «СС718», «СС724».

С ноября 1990 г. Уткин – директор Центрального научно-исследовательского института машиностроения (ЦНИИмаш) – головного отраслевого института ракетно-космической техники. В декабре 1994 г. была сформирована независимая российско-американская комиссия по проблемам обеспечения совместных пилотируемых полетов, сопредседателями которой были назначены Уткин и астронавт США Т. Стаффорд. Под руководством Владимира Федоровича разработана Российская программа ракетно-космической техники.

КБ «Южное» отличилось созданием космических летательных аппаратов и ракет-носителей современного типа. Под чутким руководством Уткина разработаны, выпущены и сданы в эксплуатацию четыре ракетных комплекса, сравнявших космические достижения Советского Союза с аналогичными американскими.

Настоящей гордостью бюро стала ракета «Зенит», являющаяся высокоэффективной, экологически чистой и способной выводить на орбиту до двенадцати тонн полезного груза; а также твердотопливный агрегат РТ-23 и супермощная ракета Р-36М, которая не имеет аналогов в Соединенных Штатах Америки и в кругу международных военных специалистов известна как «Сатана».

Международное сотрудничество

Очень много времени и сил Владимир Федорович посвятил различным международным проектам. Одной из самых продуктивных стала масштабная программа «Интеркосмос», в рамках которой ученые из разных стран совместными усилиями осваивали околоземное пространство. Еще можно вспомнить проект «Аркад», реализованный вместе с французами.

Уткин – автор ряда крупных изобретений. Список его основных научных трудов содержит около 200 наименований. В 1964 г. ему присуждена ученая степень доктора технических наук, в этом же году он стал

лауреатом Ленинской, а в 1981 г. – Государственной премии СССР. С 1972 г. – член-корреспондент, в 1976 г. – академик АН УССР, в 1984 г. избран академиком АН СССР. Владимир Федорович был президентом Российской академии космонавтики им. К.Э. Циолковского, действительным членом Международной академии астронавтики, почетным членом Российской академии ракетных и артиллерийских наук.

Награжден шестью орденами Ленина, двумя орденами Красной Звезды, орденами Отечественной войны 2-й степ., Трудового Красного Знамени, «За заслуги перед Отечеством» 2-й и 3-й степ., 16 медалями, почетным знаком «Отличный связист», лауреат золотой медали им. С.П. Королева. Дважды Герой Социалистического Труда. Решением исполкома Рязанского городского Совета народных депутатов № 236 от 27 мая 1987 г. «за выдающийся вклад в науку и технику, активную общественную деятельность» В.Ф. Уткину присвоено звание «Почетный гражданин города Рязани».

Умер 15 февраля 2000 г. в Москве. Похоронен на Троекуровском кладбище г. Москвы.

Бронзовый бюст работы скульптора К.И. Чеканева и архитектора А.И. Супонина был установлен еще при жизни ученого 7 января 1984 г. на улице К.Э. Циолковского в Рязани. МонуMENT в честь академика Уткина работы архитектора Б.С. Горбунова открыт в Касимове 14 сентября 2002 г. рядом со школой № 2 им. В.Ф. Уткина, в которой создан музей братьев Уткиных. В музее К.Э. Циолковского средней школы № 16 г. Рязани Уткину посвящена экспозиция. Рязанский городской Совет принял решение от 17 января 2002 г. № 16 о присвоении скверу напротив Рязанского колледжа отраслевых технологий, экономики и права его имени.

В память выдающегося российского ученого 27 апреля 2001 г. была учреждена Золотая медаль имени академика В.Ф. Уткина, вручаемая в двух номинациях «За вклад в развитие экономики и укрепление обороноспособности страны» и «За вклад в развитие экономики Рязанской области». Серебряная медаль имени академика В.Ф. Уткина присуждается в трех номинациях: «За активную общественную и публицистическую деятельность по проблемам космонавтики и оборонно-промышленного комплекса страны», «За достижения в научной и практической деятельности по космонавтике» и «За разработку и внедрение новой техники и технологий, создание современных высокоэффективных производств, реализацию социально значимых проектов в интересах развития Рязанской области».

21 августа 2002 г. было принято постановление Рязанской областной Думы «Об учреждении премий Рязанской области по науке и технике имени академика В.Ф. Уткина» в целях увековечения его памяти и стимулирования развития научно-технического и производственного потенциала области в сфере высоких технологий. Мемориальная доска установлена на фасаде административного корпуса «ЦНИИмаш», на здании ГКБ «Южное» (г. Днепропетровск), на доме, в котором жил Уткин, в посёлке



Лашма Рязанской области. Его именем названы улицы в г. Касимове и поселке Лашма. В его честь был назван астероид 13477 Utkin.

В Рязанском радиотехническом университете появился памятный бюст Владимира Федоровича Уткина. Автор бюста – скульптор Раиса Лысенина.

Академик В.Ф. Уткин является Почетным гражданином Рязанской области, городов Рязани и Касимова. Его разработки, ставшие основой ракетно-ядерного щита России, хранят мирное небо над нашей Родиной уже почти 50 лет.

УДК 519.95

Козанков И.Ю.

МЕТРИКИ ДЛЯ ТЕСТИРОВАНИЯ ТОЧНОСТИ АЛГОРИТМОВ НЕЭТАЛОННОЙ ОЦЕНКИ КАЧЕСТВА ИЗОБРАЖЕНИЙ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

В данной работе рассматриваются метрики, используемые для тестирования точности алгоритмов неэталонной оценки качества изображений.

Неэталонные алгоритмы оценки качества изображений подразделяются на две группы: алгоритмы, для работы которых необходима априорная информация о типе искажения, и алгоритмы, которые являются работоспособными без дополнительной информации об искажениях [1]. Классификация неэталонных алгоритмов изображена на рис. 1.



Рис. 1. Классификация неэталонных алгоритмов

Пусть имеются $\{x_i\}^N = \{x_1, x_2, \dots, x_N\}$, $\{y_i\}^N = \{y_1, y_2, \dots, y_N\}$, где $\{x_i\}^N$ – оценки качества на выходе алгоритма неэталонной оценки качества изображений для N тестовых изображений, а $\{y_i\}^N$ – соответствующие экспертные оценки качества. Тогда оценить точность алгоритмов неэталонной оценки качества изображений можно с помощью следующих метрик: коэффициент корреляции Пирсона, коэффициент корреляции Спирмена, коэффициент корреляции Кендалла. Рассмотрим каждый из них.

1. Коэффициент корреляции Пирсона

$$r = \frac{\sum_{i=1}^N (y_i - \bar{y})(x_i - \bar{x})}{\sqrt{\sum_{i=1}^N (y_i - \bar{y})^2 (x_i - \bar{x})^2}} \quad [2], \text{ где } \bar{x} - \text{среднее значение оценки на}$$

выходе алгоритма, \bar{y} – среднее значение экспертной оценки.

2. Коэффициент корреляции Спирмена

$$\rho = 1 - \frac{6}{N(N-1)(N+1)} \sum_{i=1}^N (R_i - S_i)^2 \quad [2], \text{ где } R_i - \text{ранг наблюдения } x_i \text{ в}$$

x , S_i – ранг наблюдения y_i в ряду y .

3. Коэффициент корреляции Кендалла

$$\tau = 1 - \frac{4}{N(N-1)} R \quad [2], \text{ где } R = \sum_{i=1}^{N-1} \sum_{j=i+1}^N [[x_i < x_j] \neq [y_i < y_j]].$$

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Оценка качества цифровых изображений и видеоданных: учебно-методическое пособие / Д.В. Матвеев, А.Г. Седов, В.В. Хрящев, А.Л. Приоров; Яросл. гос. ун-т им. П.Г. Демидова. – Ярославль: ЯрГУ, 2018. – 76 с.

2. Кобзарь А.И. Прикладная математическая статистика. – М.: Физматлит, 2006. – 816 с.

УДК 004.9

Дмитриева Т.А., Мерзлякова М.Э.

ИНТЕРФЕЙС ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ СЕБЕСТОИМОСТЬЮ ПРОДУКЦИИ НА ПРЕДПРИЯТИИ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

Рассматривается интерфейс разработанной информационной системы, предназначенной для поиска резервов снижения себестоимости продукции на предприятии

Разработанная информационная система предназначена для автоматизированного анализа статей калькуляции и расчета себестоимости продукции в расчете на единицу продукции. В информационную систему можно добавлять любые необходимые характеристики, что дает возмож-

ность оценить стратегическую важность представленных проектов, а также относительную важность критериев оценки [1].

Для выполнения входа в систему для начала необходимо авторизоваться. Для этого в поле «Пользователь» необходимо выбрать тип в соответствии со своим статусом, после ввести пароль в соответствующем поле (рис. 1).

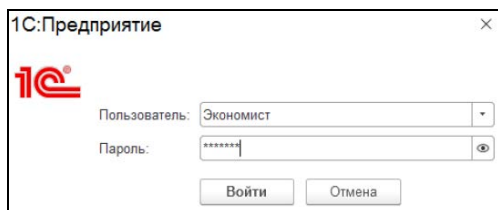
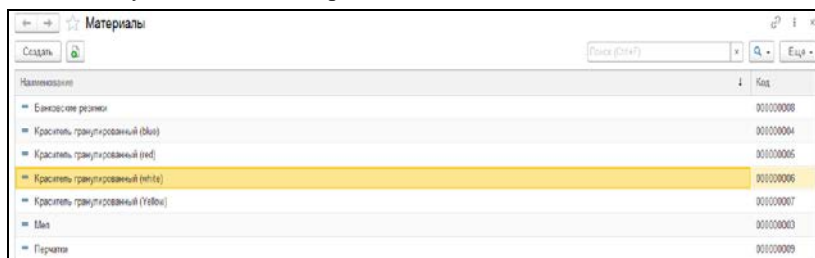


Рис. 1. Вход в информационную систему

Сначала пользователь размещает наименования материалов в справочнике «Материалы» (рис. 2) и их количества на единицу изделия в регистре «Количество материалов». Нормы расхода материала на единицу запишем в периодический регистр сведений, учитывая, что эти показатели могут изменяться со временем.

Чтобы добавить материалы, пользователю необходимо нажать на кнопку «Создать», после чего откроется окно и будут доступны поля для заполнения. Здесь необходимо добавить материал из соответствующего справочника «Материалы». Поле «Изделие» и «Количество» заполняется вручную. После внесения информации необходимо подтвердить действия, нажав кнопку «Записать и закрыть».



Наименование	Код
Бюксосоме резинки	001000008
Краситель гранулированный (blue)	001000004
Краситель гранулированный (red)	001000005
Краситель гранулированный (white)	001000006
Краситель гранулированный (Yellow)	001000007
Мыл	001000003
Перчатки	001000009

Рис. 2. Добавление наименования материалов в систему

В справочнике «Поставщики», представленном на рис. 3, записываем и группируем поставщиков по поставляемым комплектующим.

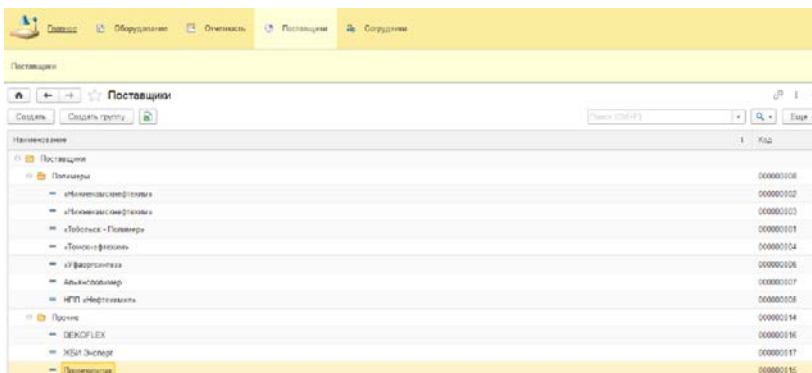


Рис. 3. Добавление поставщиков в систему

Заполнив справочник «Поставщики», пользователь переходит к документу «Оценка поставщиков» и заполняет соответствующие поля «цена», «доставка», «качество» и «риски» (см. рис. 4, 5).

Характеристики поставщиков

Период:

Поставщик:

Материал:

Цена:

Доставка:

Качество:

Риски:

Рис. 4. Добавление характеристики поставщиков в систему

№	Поставщик	Материал	Цена	Доставка	Качество	Риски
1	Леруа Мерлен г. Рязань	Аксессуары для радиатора	350 рублей	9	7	10 %
	Агротек г. Тула	Аксессуары для радиатора	380 рублей	9	7	20 %
	"101 кабатон" г. Тула	Комплект для подключения	183 рублей	6	4	24 %
	Леруа Мерлен г. Рязань	Комплект для подключения	190 рублей	9	8	10 %
	"Делсо" г. Челябинск	Комплект для подключения	206 рублей	6	6	43 %
	"Неом" г. Ростов-на-Дону	Комплект для подключения	199 рублей	4	5	15 %
	"Аматрис" г. Москва	Комплект для подключения	192 рублей	4	10	20 %
	"ГрандСтрой" г. Ярославль	Комплект для подключения	187 рублей	6	9	45 %
	"Армада" г. Нижний Новгород	Комплект для подключения	232 рублей	6	4	70 %
	"Ильмерия" г. Екатеринбург	Комплект для подключения	220 рублей	5	7	32 %
	"Кадрат" г. Москва	Комплект для подключения	196 рублей	7	9	35 %
	Агротек г. Тула	Комплект для подключения	185 рублей	7	6	56 %
	"Ориент" г. Москва	Биметаллические секции	157 рублей	8	7	60 %
	Агротек г. Тула	Биметаллические секции	215 рублей	6	7	50 %
	"Аматрис" г. Москва	Биметаллические секции	212 рублей	4	10	20 %
	"Новый век" г. Москва	Биметаллические секции	230 рублей	8	9	10 %
	"Евро Сервис" г.Тула	Биметаллические секции	210 рублей	3	4	45 %
	"Неом" г. Ростов-на-Дону	Биметаллические секции	185 рублей	4	5	15 %
	"Армада" г. Нижний Новгород	Биметаллические секции	167 рублей	6	4	70 %
	"Волна" г. Сочи	Биметаллические секции	190 рублей	9	10	30 %

Рис. 5. Справочник характеристик поставщиков

Далее пользователь переходит к документу «Анализ материалов» (см. рис. 6), где сначала выбирает материала, цену которого необходимо рассчитать, выбрав в графе «Первый поставщик» поставщика с наивысшим интегральным показателем, а в графе «Второй поставщик» соответствующие цены старого поставщика предприятия.

N	Материал	Первый поставщик	Второй поставщик	Итог оптимальный	Итог фактический	Выявл.
1	Виниловое покрытие	"Волга" г. Сочи	Старый поставщик "СЕРДИБ"	3 040	4 488	Повышение по цене
2	Клей для подложки	"Ладья Марин" г. Рязань	Старый поставщик "РАДИАТ"	190	235	Повышение по цене
3	Акселератор	"Тайм" г. Подольск	Старый поставщик "АКСЕСС"	1 300	1 600	Повышение по цене

Рис. 6. Создание документа «Анализ материалов»

Для анализа изменение себестоимости продукции при выборе нового поставщика перейдем в документ «Себестоимость» (см. рис. 7), где автоматически будут рассчитываться графы «Цены до внедрения предложения», «Цены после внедрения предложения», а также экономия на единицу изделия будет рассчитываться автоматически.

N	Статья	Цена до внедрения предложения	Цена после внедрения предложения	Экономия на одну единицу
1	Сырье и материалы	6 603,00	4 630	2 073
2	Транспортно-заготовительные расходы	1 659,75	1 132,5	518
3	Эти основные производственные расходы	610,70	610,70	0
4	Отчисления во внебюджетные фонды	103,23	103,23	0
5	Амортизация	30,69	30,69	0
6	Налоговые расходы	1 221,66	1 221,66	0

Рис. 7. Расчет себестоимости продукции и экономии на 1 единицу при выборе нового поставщика

Разработанная информационная система предназначена для анализа любой статьи затрат себестоимости. Рассмотрим на примере анализа статьи затрат «Заработная плата».

В справочник «Заработная плата» (см. рис. 8) заносятся данные с именем работника, его разрядом, трудоемкостью выполнения работы и часовой тарифной ставкой. После того, как информация внесена, необходимо нажать на кнопку «Записать и закрыть».

После заполнения справочника «Работники» пользователь может перейти к документу «Заработная плата» и начать заполнение полей.

1. Поле «Работник», «Разряд рабочего», «Трудоемкость» – заполняется автоматически.

2. Поле «Фактические часы» заполняется пользователем.

Работники (создание) *	
Код:	<input type="text"/>
Наименование:	<input type="text" value="Лучин Кирилл Викторович"/>
Должность/разряд:	<input type="text" value="I разряд"/>
Трудоемкость:	<input type="text" value="2,190"/>
Часовая тарифная ставка:	<input type="text" value="143,15"/>
Оклад в месяц:	<input type="text" value="0,00"/>

Рис. 8. Добавление работника в систему

После заполнения данных полей происходит сравнение трудоемкости фактической с оптимальной и вывод полученного расчета.

Таким образом, разработанная информационная система позволяет проводить анализ статей затрат, рассчитывать себестоимость продукции в целом, производить расчет интегрального показателя, необходимый для дальнейшего выбора оптимального поставщика для предприятия. Это позволяет выявлять проблемные места при анализе статей затрат себестоимости продукции и поможет контролировать не только рассматриваемую статью «Сырье и материалы», но и другие [2].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лыжина Н.В. Управление затратами предприятия [Электронный ресурс]: учебное пособие / Н.В. Лыжина, Р.М. Уханова. – Электрон. текстовые данные. – Казань: Казанский национальный исследовательский технологический университет, 2017. – Режим доступа: <http://www.iprbookshop.ru/80253.html> (дата обращения: 16.04.2021).

2. Зарубежный опыт управления себестоимостью продукции [Электронный ресурс]. – Режим доступа: <http://lectmania.ru/1x9e71.html> (дата обращения: 15.05.2021).

УДК 004

Благодарова Т.А., Благодаров Е.А.

АЛГОРИТМ ВЫПОЛНЕНИЯ СЦЕНАРИЯ ГРАДУИРОВКИ ПРЕОБРАЗОВАТЕЛЕЙ ДАВЛЕНИЯ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

При производстве электронных преобразователей давления (ПД) возникает потребность в автоматизации процесса их градуировки. Под градуировкой понимается сбор и обработка данных для настройки аппроксимационного полинома [1], используемого для вычисления давления контролируемой среды.

При автоматизированной градуировке ПД обычно используется датчик давления с дистанционным управлением, а также камера тепла и

холода. Датчик подает на градуируемые ПД требуемое давление. Камера тепла и холода позволяет создать требуемую температуру воздуха, в среде которой находятся градуируемые ПД.

Процесс градуировки выполняют по определенному сценарию, который последовательно задает требуемые сочетания температуры и давления для ПД. В статье предлагается алгоритм выполнения сценария градуировки ПД.

Алгоритм основан на использовании конечного автомата, поэтому его графическое изображение целесообразно представить в виде диаграммы состояний (Statechart Diagram) [2] в нотации UML (рисунок 1).

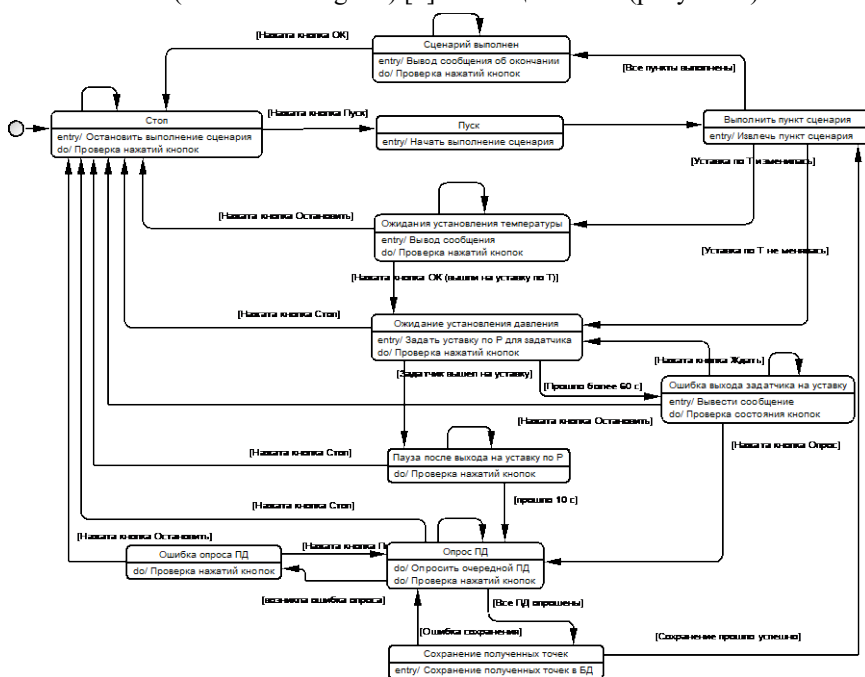


Рис. 1. Диаграмма состояний

Пункты сценария хранятся в таблице базы данных (БД). Для каждого пункта указаны значения уставок температуры T и давления P , признак выполнения, а также номер по порядку выполнения. Результатом работы алгоритма является множество градуировочных точек для каждого из градуируемых ПД. Каждая точка содержит следующие данные: значение температуры, значение давления, код аналого-цифрового преобразователя (АЦП) с выхода тензопреобразователя, сопротивление тензопреобразователя. Первые два значения получают по сценарию, а вторые два – от градуируемого ПД. Полученные точки помещают в таблицу БД.

Переходы между состояниями выполняются по таймеру с дискретностью 100 мс. Значение дискретности по времени выбрано из соображений достаточной точности отсчета потребных временных промежутков при переходах между состояниями, разумной нагрузки на центральный процессор компьютера, а также «отзывчивости» программы на действия пользователя.

В начале работы алгоритма конечны автомат находится в состоянии **«Стоп»**. При заходе в это состояние выполняются необходимые действия по остановке выполнения сценария. При нахождении в состоянии проверяется нажатие кнопки **Пуск** в главном окне приложения. Если кнопка **Пуск** нажата, выполняется переход в состояние **«Пуск»**.

При заходе в состояние **«Пуск»** производится подготовка к выполнению сценария, в том числе, выявляется количество невыполненных пунктов сценария. Далее происходит переход к состоянию **«Выполнить пункт сценария»**.

При заходе в состояние **«Выполнить пункт сценария»** из БД извлекается информация об очередном по счету пункте сценария, а именно: значения уставок по температуре T и давлению P . Если все пункты сценария выполнены, то происходит переход к состоянию **«Сценарий выполнен»**. В противном случае проверяется значение уставки по T . Если новая уставка по T изменилась, т.е. отличается от той, которая была ранее, то происходит переход к состоянию **«Ожидание установления температуры»**. В противном случае происходит переход к состоянию **«Ожидание установления давления»**.

При заходе в состояние **«Ожидание установления температуры»** выводится окно с предложением установить в камере тепла и холода требуемую температуру. После того, как температура будет установлена (это может занять много времени), пользователь нажимает кнопку **ОК**, в результате чего происходит переход к состоянию **«Ожидание установления давления»**. Если же пользователю требуется остановить выполнение сценария, он нажимает кнопку **«Остановить»** – автомат переходит в состояние **«Стоп»**.

При заходе в состояние **«Ожидание установления давления»** производится задание уставки по P для задатчика давления. После этого производится периодический опрос состояния задатчика. Если задатчик вышел на уставку, то производится переход к состоянию **«Пауза после выхода на уставку по P »**. Если же задатчик не выйдет на уставку в течение 60 с, то происходит переход к состоянию **«Ошибка выхода задатчика на уставку»**. Также пользователь может нажатием кнопки **«Стоп»** перевести автомат в состояние **«Стоп»**.

При заходе в состояние **«Ошибка выхода задатчика на уставку»** выдается окно с сообщением об ошибке. Если пользователь нажмет в этом окне кнопку **«Ждать»**, то произойдет возврат к состоянию **«Ожидания»**.

ние установления давления». При нажатии кнопки «Остановить» выполнение сценария остановится переходом в состояние «Стоп». Нажатие кнопки «Опрос» позволит принудительно перейти к состоянию «Опрос ПД», не дожидаясь выхода на уставку.

В состоянии «Пауза после выхода на уставку по Р» автомат ждет истечения 10 с, которые требуются для получения стабильных и качественных результатов опроса ПД. При этом пользователь может нажатием кнопки «Стоп» прервать выполнение сценария. По истечении 10 с происходит автоматический переход к состоянию «Опрос ПД».

В состоянии «Опрос ПД» производится сбор данных со всех ПД в группе. Собираются данные о текущих значениях кода АЦП с выхода тензопреобразователя и сопротивлении тензомоста. Также фиксируется текущее значение давления задатчика и температуры в камере тепла и холода. После сбора данных от всех ПД происходит переход к состоянию «Сохранение полученных точек». Если не удалось собрать данные со всех ПД, происходит переход к состоянию «Ошибка опроса ПД». Также при необходимости пользователь может нажатием кнопки «Стоп» прервать выполнение сценария.

При заходе в состояние «Ошибка опроса ПД» выводится окно с сообщением об ошибке. При нажатии кнопки «Повтор» будет произведен возврат в состояние «Опрос ПД». При нажатии кнопки «Остановить» выполнение сценария остановится переходом в состояние «Стоп».

При заходе в состояние «Сохранение полученных точек» производится запись в БД накопленных данных по всем ПД для текущего состояния давления и температуры. При этом используется механизм транзакций. Если в процессе сохранения возникли какие-либо ошибки, происходит откат транзакции и возврат в состояние «Сохранение полученных точек». Это позволяет избежать потери данных при редких сбоях. Если же сохранение прошло успешно, то происходит переход к состоянию «Выполнить пункт сценария».

При заходе в состоянии «Сценарий выполнен» показывается окно с соответствующим сообщением, а при нажатии кнопки ОК в нем, происходит возврат в состояние «Стоп».

Заметим, что в случае, если автомат переведен в состояние «Стоп», но еще не все пункты сценария выполнены, можно продолжить выполнение сценария, нажатием кнопки «Пуск». Это позволяет выполнять сценарий частями, например, за несколько рабочих дней.

Для удобства пользователя в процессе выполнения сценария в строке состояния главного окна выводятся сообщения, поясняющие ход выполнения сценария, а также индикатор процентов выполнения сценария.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Благодаров А.В., Виноградов А.Л. Программная реализация аппроксимации функции двух переменных при использовании двумерных

полиномов // Проблемы развития корабельного вооружения и судового радиоэлектронного оборудования.– 2018, №3.– С.20–24.

2. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка. 2-е изд.– СПб.: Питер, 2007.–544с.

УДК 004

Кирилова Л.М.

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПОИСКА НЕДОСТОВЕРНОЙ ИНФОРМАЦИИ В ИНТЕРНЕТЕ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

В современном мире СМИ приобретают очень большой вес. Люди составляют своё мнение о происходящем в мире, зачастую не зная самой ситуации, а смотря на то, как это освещается в СМИ. Таким образом, СМИ стали институтом, формирующим мнения людей, выбирая как осветить то или иное событие.

В ранее выпущенной статье [1] рассмотрены существующие программные продукты для поиска достоверной информации. Выделяя общие недостатки, можно заметить, что большинство решений на рынке являются либо узкоспециализированными, либо являются платными, их стоимость может быть затруднительной для рядовых пользователей для применения в личных нуждах. Говоря о распространенном открытом программном обеспечении, в качестве его недостатков следует выделить, что, зачастую, неизвестен алгоритм, подход к решению поставленной задачи, в связи с чем нельзя однозначно гарантировать достоверность результатов его работы (это в той же мере равносильно и для проприетарных аналогов), а также то, что поиск достоверной информации осуществляется за ограниченные временные периоды, при этом не учитывается количество аналогичных новостей за период.

Поэтому целесообразно разработать новую программу, которая будет отвечать всем необходимым требованиям. Она будет полезна как в профессиональной сфере, например, политикам или бизнесменам, так и рядовым пользователям. При этом, опираясь на математический аппарат метода поиска достоверной информации, можно с определенной долей вероятности говорить о достоверности интерпретации получаемых результатов.

Необходимо автоматизировать существующий процесс поиска фейковой информации с помощью разработки программного обеспечения (рис. 1). В предлагаемом подходе основным является снятие субъективизма принимаемого решения благодаря внедрению расчетных коэффициентов для анализируемых показателей. На основании которых, в свою очередь, рассчитывается общий коэффициент достоверности всей интернет-новости.

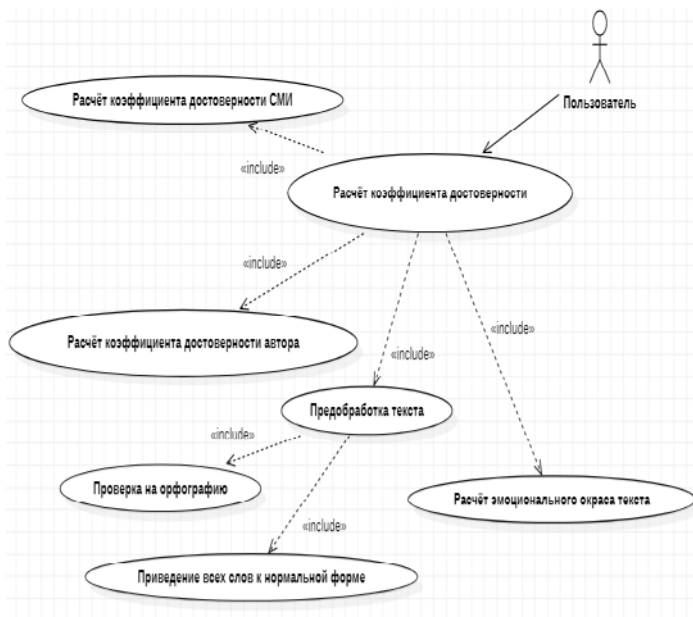


Рис.1. Варианты использования модели TO BE

При разработке интерфейсной-части приложения необходимо учитывать основные этапы работы реализуемой информационной системы.

Основные этапы:

- 1) ввод данных пользователем;
- 2) вывод данных пользователю.

Окно ввода данных необходимо спроектировать таким образом, чтобы была возможность ввести все необходимые для анализа данные. Данные, которые учитываются при анализе:

1. ФИО автора.
2. Наименование СМИ.
3. Заголовок материала.
4. Дата публикации материала.
5. Содержание материала (текст).

Кроме того, нужно учесть тот факт, что есть данные, необязательные для ввода. Например, дата публикации материала влияет на расчёт достоверности новостного материала, но ввод этой информации необязателен. Соответственно, данный факт необходимо учесть при проектировании интерфейса и написания логики приложения.

В окне вывода данных должны содержаться данные отчёта по анализу определённого новостного материала. Основная особенность интерфейса заключается в том, что необходимо вывести коэффициент по каж-

дому параметру для прозрачности расчёта достоверности. Разберём, из каких частей состоит отчёт:

1. Информация по автору.
2. Информация по СМИ.
3. Информация по орфографии заголовка.
4. Информация по орфографии содержания новостного материала.
5. Информация по дате публикации.
6. Анализ эмоционального окраса заголовка.
7. Анализ эмоционального окраса содержимого новостного материала.

Таким образом, можно сделать вывод о том, что при разрабатываемое приложение удалит человеческий фактор из анализа статьи и даст оценку данных по различным параметрам.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Фейковые новости в современном мире/ Алексахин В.В., Баконин Р.В., Кирилова Л.М. // Математическое и программное обеспечение вычислительных систем: межвуз. сб. науч. тр./ под ред. А.Н. Пылькина – Рязань, декабрь 2020.

УДК 004

Латахин А.

ВОССТАНОВЛЕНИЕ НЕСФОКУСИРОВАННЫХ ИЗОБРАЖЕНИЙ МЕТОДОМ ДВУМЕРНОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ И РЕГУЛЯРИЗАЦИИ ТИХОНОВА

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

Рассматривается задача реконструкции несфокусированных изображений. Для решения прямой задачи (моделирования несфокусированного изображения) используется двумерное преобразование Фурье. Для решения обратной задачи (реконструкции несфокусированного изображения) используется метод преобразования Фурье и метод регуляризации Тихонова.

Введение. В данной работе рассматривается задача восстановления несфокусированных изображений. Под изображением подразумевается фотоснимок или оптико-электронное воспроизведение объекта, текста, человека, здания, самолета, автомобиля, космического объекта, наземного объекта из космоса и т.д. Реконструкция несфокусированных изображений является одной из актуальных задач цифровой обработки изображений.

Одним из распространенных способов решения этой некорректной задачи является использование преобразования Фурье (ПФ) с фильтрацией Винера, Тихонова. При этом края несфокусированного изображения обычно формируются с использованием так называемых граничных условий. Их введение обусловлено тем, что на практике для регистрации

изображений применяются системы с матрицами, размер которых совпадает с размером области фокусировки.

Математическое описание задачи получения нефокусированного изображения

Математически задача восстановления изображений, искаженных линейной пространственно-инвариантной системой, описывается интегральным уравнением Фредгольма I рода типа свертки:

$$Af \equiv \int_{-\infty-\infty}^{+\infty+\infty} \int k(x-\xi, y-\eta) f(\xi, \eta) d\xi d\eta = g(x, y), \quad -\infty < x, y < +\infty,$$

где $g(x, y)$ – искаженное изображение на выходе системы; $f(\xi, \eta)$ – исходное изображение на входе системы; $k(x-\xi, y-\eta)$ – импульсная характеристика системы, или функция рассеяния точки (ФРТ), являющаяся трансляционно-инвариантной (разностной)

Считаем, что фотографируемый объект (для простоты полагаемый плоским) и фотопленка фотоаппарата расположены параллельно апертуре линзы фотоаппарата по разные стороны от нее на расстояниях соответственно, f_1 и $f_2 + \delta$ от линзы (δ – погрешность фокусировки изображения),

причем: $\frac{1}{f_1} + \frac{1}{f_2} = \frac{1}{f}$, где f – фокусное расстояние линзы. В результате на фотопленке возникнет перевернутое изображение (см. рис. 1).

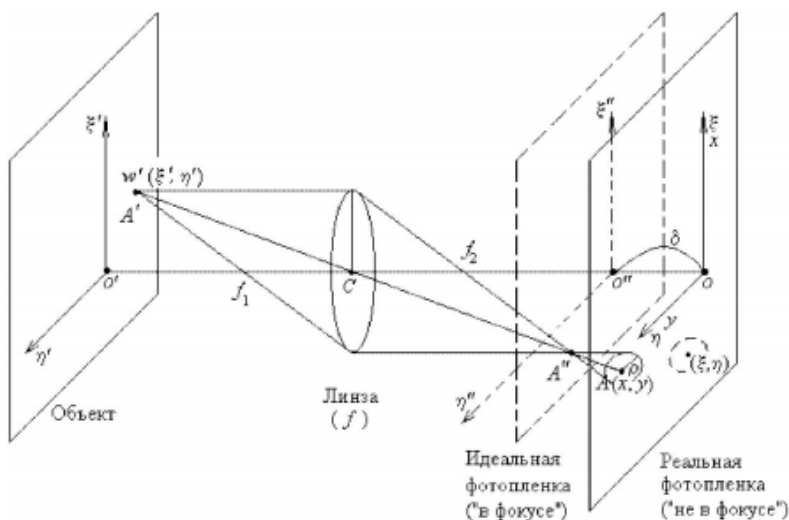


Рис. 1. Модель расфокусировки изображения

Опишем математически задачу дефокусировки. Рассмотрим, помимо круга с центром в точке $A(x, y)$, некоторый другой круг с центром в точке (ξ, η) (рис. 1). Радиусы этих (и других) кругов одинаковы и равны ρ , а их площади равны $S = \pi\rho^2$. В результате некоторая интенсивность $w(\xi, \eta)$, соответствующая точке (ξ, η) , будет «размазана» по кругу радиуса ρ и площади $S = \pi\rho^2$ с плотностью интенсивности $w(\xi, \eta) / \pi\rho^2$. Интенсивность в точке $A(x, y)$ будет результатом суммирования (интегрирования) по всем тем кругам, которые накрывают точку $A(x, y)$.

Реконструкция нефокусированного изображения методом преобразования Фурье и регуляризации Тихонова

В методе регуляризации Тихонова, регуляризация задачи достигается заменой исходной задачи на задачу нахождения экстремума (минимума) некоторого сглаживающего функционала.

В методе регуляризации Тихонова решение находится из условия минимума сглаживающего функционала:

$$\int_{-\infty}^{\infty} [Hw - f(x)]^2 dx + \alpha \int_{-\infty}^{\infty} M(\omega) |Y(\omega)|^2 d\omega = \min_w.$$

Таким образом, $W(\omega_1, \omega_2) = \frac{G(\omega_1, \omega_2)}{K(\omega_1, \omega_2)}$, где

$$G(\omega_1, \omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{i(\omega_1 x + \omega_2 y)} dx dy,$$

$$K(\omega_1, \omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k(x, y) e^{i(\omega_1 x + \omega_2 y)} dx dy,$$

где $G(\omega_1, \omega_2)$ и $K(\omega_1, \omega_2)$ – преобразования Фурье (спектры) правой части и ядра интегрального уравнения.

Однако задача решения уравнения является некорректной. Это связано с тем, что функция измеряется с погрешностью, и это ведет к сколь угодно большим погрешностям решения $w(\xi, \eta)$. Поэтому формулы не годятся для устойчивого решения уравнения. Устойчивое решение уравнения методами двумерного ПФ и регуляризации Тихонова имеет вид

$$w_\alpha(\xi, \eta) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_\alpha(\omega_1, \omega_2) e^{-i(\omega_1 \xi + \omega_2 \eta)} d\omega_1 d\omega_2.$$

Выбор параметра регуляризации

Важным является вопрос о выборе значения параметра регуляризации α . Разработан ряд способов выбора в методе регуляризации Тихонова. Выбор можно осуществлять, например, способом невязки или обобщенным

принципом невязки. Разработаны также следующие способы выбора параметра регуляризации: способ квазиоптимального (квазинаилучшего), способ отношения, способ независимых реализаций, способ перекрестной значимости, способ моделирования и др. Однако для задачи реконструкции изображений, как показала практика, более эффективен способ подбора.

Согласно ему, для ряда значений α вычисляются решения по вышеизложенным формулам, они выводятся на дисплей в графической форме и выбирается значение, дающее наилучшее восстановление изображения с точки зрения визуальных, физиологических (но не математических) критериев восприятия. Этот способ аналогичен способу настройки контраста телеизображения. Способ подбора можно назвать также визуальным критерием, или критерием качественной оценки. Этот способ эффективен при реконструкции реальных нефокусированных изображений, когда истинное изображение w неизвестно.

Сравнение методов реконструкции

В основном используются такие методы реконструкции нефокусированных изображений, как метод фильтрации Винера, метод регуляризации Тихонова и др. При этом как прямая, так и обратная задачи в них решаются с использованием преобразования Фурье. Однако аппарат ПФ неадекватен физической сути задачи дефокусировки, в которой сама природа использует лишь операцию накопления (суммирования) в пределах ФРТ. Делается вывод, что наилучшие результаты должны давать методы, в которых как прямая, так и обратная задачи решаются с использованием лишь операций суммирования. Это – методы квадратур, кубатур, итераций и т.п.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бейтс Р., Мак-Доннелл М. Восстановление и реконструкция изображений. – М.: Мир, 1989. – 336 с.
2. Грузман И.С., Киричук В.С., Косых В.П. и др. Цифровая обработка изображений в информационных системах. – Новосибирск: Изд-во НГТУ, 2000. – 168 с.
3. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Технофера, 2006. – 1072 с.

УДК 004

Швечкова О.Г., Елисеев В.А.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПОДБОРА КОМПЛЕКТУЮЩИХ ДЛЯ КОМПЬЮТЕРА НА ОСНОВЕ МНОГОКРИТЕРИАЛЬНОЙ ВЫБОРКИ

Рязанский государственный радиотехнический университет имени В.Ф. Уткина, г. Рязань

Рассматривается процесс разработки автоматизированной системы для подбора комплектующих для компьютера на основе многокритериальной выборки. Приводятся основные особенности существующих методов подбора, их достоинства и недостатки, а также обоснование необходимости автоматизации данного процесса.

В настоящее время подбор комплектующих для компьютера является весьма важной задачей, т.к. текущее положение дел в стране и в мире в целом вызвал бурный, можно сказать лавинообразный рост спроса на средства вычислительной техники. Увеличение числа рабочих мест и объема дистанционного обучения стимулировал спрос на персональные компьютеры (ПК) как со стороны бизнеса, так и со стороны потребителей. За 2020 год объем продаж персональных компьютеров увеличился на 13,1 % и составил 301,2 млн штук. По мнению компетентных источников, к концу 2020 года спрос на ПК превышал объемы производства, а каналы поставок работали на максимуме своих возможностей. Бренды и контрактные производители столкнулись с нехваткой комплектующих, также наблюдались проблемы с логистикой. Поставщики были вынуждены прибегать к авиаперевозкам, что привело к увеличению затрат, но позволило ускорить доставку. Все эти факторы привели к острому дефициту на некоторые виды комплектующих, что привело к их подорожанию, которое в отдельных случаях составляло 150-200 %.

На данном этапе развития информационного общества трудно найти ту сферу деятельности человека, которая могла бы обойтись без применения ЭВМ. Отсюда следует вывод, что объемы производства и применения средств вычислительной техники во всех сферах деятельности человека настоятельно требуют не просто универсальных вычислительных устройств, а компьютеров, ориентированных на решение определённого класса задач. Выбор вычислительного устройства при таком подходе сводится к применению ЭВМ с набором функциональных возможностей, определяемым набором комплектующих компонентов, сформированным по определённым критериям эффективности.

Эффективность заключается в максимально быстром решении той задачи, для которой требуется компьютер. Так для одних сфер деятельности наиболее важной является производительность процессора, которая в свою очередь тоже имеет ряд параметров (производительность на одно

ядро, объемы кеша, многоядерная производительность). Для работы с графикой мощность процессора отходит на второй план и важнейшей является производительность видеокарты (производительность ГПУ, объемы видеопамати и поддерживаемые видеоинтерфейсы). Затрагивая понятия эффективности, нельзя обойти стороной и количество потребляемой энергии, которая затрачивается для решения определенных задач. Существуют комплектующие, которые имеют схожую производительность, но количество энергии, которую они затратят на одну и ту же задачу может существенно отличаться. Этот параметр зависит прежде всего от техпроцесса производства, а также от эффективности архитектуры, на основе которой разрабатывается чип. Для других типов задач высокая производительность не требуется, а важнейшим становится надежность и бесперебойная работа. Для повышения надежности используются более производительные системы охлаждения, которые позволяют работать комплектующим при более низкой температуре, а также качественные блоки питания, которые позволят бесперебойно работать всей системе даже при скачках напряжения или других неполадках в сети.

Учет всех требований пользователя при подборе комплектующих представляется трудоёмкой задачей. В настоящее время эта проблема решается покупкой готовых решений в магазинах бытовой электроники или специализированных компьютерных магазинах, а также путем обращения к специалистам, готовым подобрать, подходящее для ваших задач решение.

Однако, данные подходы и применяемые методы не могут считаться универсальными и эффективными, так как имеются очевидные проблемы. В случае покупки готовых сборок в магазинах электроники возникает сразу несколько проблем.

Первая – это ограниченность выбора среди того количества собранных ПК, которое есть в магазинах. Это в свою очередь приводит к тому, что ни один из предложенных вариантов не будет сбалансированным и всецело подходящим под требуемые задачи.

Вторая проблема не менее, а возможно, и более критическая. Ее суть заключается в профессионализме консультанта, который будет помогать вам определяться с выбором компьютера. Так, во многих магазинах (даже в тех, которые ориентированы непосредственно на продажу компьютеров) берут на работу специалистов без каких-либо профильных знаний, не говоря уже про образование. А человеку, который слабо разбирается в теме компьютерных комплектующих, будет тяжело отличить профессионала, от неопытного консультанта.

Третья проблема состоит в том, что консультанты, даже если они и обладают достаточным уровнем знаний, зачастую могут рекомендовать не тот продукт, который является лучшим, а тот который им выгоднее продать.

В случае обращения к специалистам по подбору ПК также возникает ряд проблем, основной из которой также является его профессионализм.

Конечно, в современном мире с помощью интернета можно посмотреть отзывы и мнения о качестве работы компаний, но существует также и вероятность того, что негативные отзывы были удалены, а позитивные были накручены.

Для решения перечисленных проблем и устранения имеющихся недостатков существующих методов мною предложен алгоритм автоматизированного подбора комплектующих на основе многокритериальной выборки, который позволит учесть специфику требуемых задач и выполнить подбор комплектующих, наиболее соответствующих решению поставленных задач.

Суть предлагаемого метода состоит в том, что разрабатываемая система будет определять комплектующие, которые имеют лучшие показатели для данного вида задач, учитывая при этом бюджет, который пользователь готов потратить на покупку ПК.

В зависимости от типа решаемых задач, траты на отдельный вид комплектующих (процессор, материнская плата, видеокарта и т.д.), относительно всего компьютера, могут быть достаточно сильно колебаться. Путем анализа существующих цен на комплектующие и их влияние на производительность в тех или иных задачах, были получены данные о процентном соотношении конкретного вида комплектующих к общей сумме трат на ПК. Полученные проценты не являются фиксированными и могут незначительно отличаться в зависимости от бюджета и специфических требований.

Данные сведения представлены в табл. 1.

Тип задачи	Стоимость процессора (в %, от общей суммы)	Стоимость видеокарты (в %, от общей суммы)	Стоимость Материнской платы (в %, от общей суммы)	Стоимость оперативной памяти (в %, от общей суммы)	Стоимость остальных комплектующих
Офисные/мультимедиа	35	0*	15	20	30
Игры	20	35	10	12	23
Работа с графикой/Моделирование	17	50	10	7	16
Работа с большими объемами вычислений	35	24	14	9	18

*-Для офисных задач отдельная видеокарта обычно не требуется. Для работы с графикой используется видеодаптер, встроенный в процессор.

Для выбора лучшего из возможных процессоров и видеокарт будут применяться их результаты в нескольких бенчмарках (программах, тестирующих производительность). Все результаты будут складываться и де-

литься на определенный коэффициент. По итогу получится балл, в соответствии с которым будет происходить определение лучшего. При схожей или равной производительности возможно также сравнение по непосредственным характеристикам, которые наиболее сильно влияют на результат выполнения требуемых задач. К таким характеристикам относятся:

- для процессоров – максимальная частота ядра, количество ядер, техпроцесс, тепловыделение, размер кеша и поддерживаемые инструкции.
- для видеокарт – частота видеопроцессора, объем видеопамати, тип видеопамати, ширина шины.
- для оперативной памяти – частота, объем, стандарт, тайминги.
- для материнских плат – чипсет, возможность разгона процессора, количество слотов для оперативной памяти, видеокарт и других устройств.

Программная система будет учитывать следующие параметры для выбора ПК: диапазон бюджета, назначение (которое в свою очередь будет подразделять на несколько пунктов, таких как игры, работа, мультимедиа и другие), предпочтительный производитель комплектующих, размеры корпуса, объем необходимой памяти и другие.

При внедрении разработанной программной системы предполагается достичь следующих результатов:

- облегчение и автоматизация подбора аппаратных компонентов ПК;
- повышение скорости выполнения задач для сотрудников предприятий и пользователей;
- предоставление данных о самых выгодных ценах в различных магазинах;
- рост экономической выгоды пользователям и компаниям.

После поведения этапа тестирования предполагается этап разработки инструкций пользователя, а затем размещения программной системы в открытом доступ в виде веб-сайта, где любой пользователь сможет самостоятельно выполнить подбор по множеству критериев.

Также существует возможность внедрения программы на различного рода предприятия. Для этого могут потребоваться незначительные доработки программы под специфические профессиональные задачи фирмы.

УДК 004.421

Иванова Е.С.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ АСИНХРОННЫМ ДВИГАТЕЛЕМ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

Рассматривается целесообразность реализации программного обеспечения. Приводится сравнительная характеристика алгоритмов управления асинхронным двигателем, а также перечень функциональности, обеспечиваемый разрабатываемым программным обеспечением.

Работа трехфазных асинхронных двигателей заключается в преобразовании электрической энергии в механическую энергию. По причине высокой стоимости и, в свою очередь, высокой надежности данный тип двигателей получил широкое распространение. Они используются повсеместно, являются одними из самых распространенных двигателей на рынке, их доля может достигать до 90 % от общего числа мирового производства.

Отметим, что на сегодняшний день метод частотного управления двигателями является неким техническим стандартом, при этом такие методы, как введение добавочных сопротивлений в цепи статора и ротора, а также изменение числа пар полюсов, стремительно теряют свою популярность и уже не используются в современных разработках [1].

С активным развитием микропроцессорной техники для управления двигателями все большую популярность приобретают инверторы, которые предназначены для реализации частотного управления. Для формирования выходного напряжения инвертора используется принцип широтно-импульсной модуляции. Сигналы, подающиеся на затворы транзисторов инвертора, формируют напряжение, подаваемое на двигатель.

Принято различать синусоидальную и векторную ШИМ для управления трехфазными асинхронными двигателями [2].

Если линейные напряжения трехфазной системы формировать из напряжения звена постоянного тока с помощью ШИМ по синусоидальному закону, то их максимальная амплитуда получается меньше значения напряжения в звене постоянного тока. Напряжения, формируемые синусоидальной ШИМ, проиллюстрированы графиком на рис. 1.

Чтобы полностью использовать напряжение источника постоянного тока можно использовать так называемую векторную ШИМ. При этом линейные напряжения:

1. Увеличивают свою амплитуду до 1.
2. Остаются синусоидальными и сдвинутыми на 120° .

Достоинством же синусоидальной ШИМ по сравнению векторной ШИМ является лишь простота программной реализации.

При использовании векторной ШИМ, во-первых, генерируется меньше гармонических искажений, во-вторых, обеспечивается более эффективное использование напряжения источника постоянного тока.

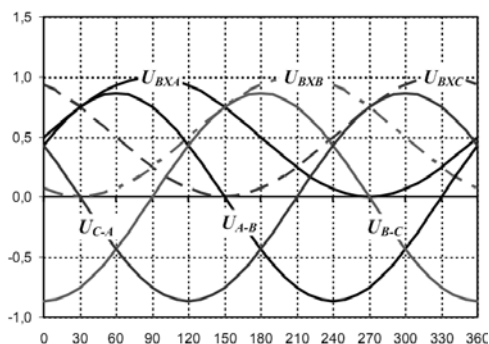


Рис. 1. Напряжения, формируемые синусоидальной ШИМ

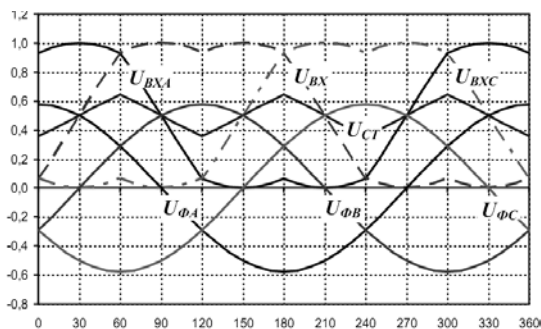


Рис. 2. Входные и фазные напряжения (векторная ШИМ)

На рис. 2 видим, что выходные напряжения фаз инвертора (U_{BXA} , U_{BXB} , U_{BXC}) имеют форму отличную от синусоидальной, а средняя точка смещена.

Но при этом фазные напряжения ($U_{\Phi A}$, $U_{\Phi B}$, $U_{\Phi C}$) синусоидальны, сдвинуты на 120° , а также имеют амплитуду равную $\frac{1}{\sqrt{3}}$. Отсюда можно сделать вывод, что напряжение звена постоянного тока используется полностью.

Современные системы управления электродвигателями [3, 4], которые позволяют регулировать частоту вращения вала двигателя, а также включают в себя систему защиты, несомненно, имеют массу достоинств. При этом необходимо выделить основные недостатки, подводящие нас к необходимости разработки новой системы.

1. Нет таких устройств, которые работали бы от АКБ с номинальным напряжением 118,4 В, все из них питаются именно от промышленных сетей.

2. Данные аналоги не сертифицированы для использования в государственных нуждах, в том числе для нужд обороны. Для разработки устройств в военном исполнении необходимо использовать комплектующие с военной приемкой.

Таким образом, необходимо разработать систему управления двигателем, которая лишена недостатков аналогичных разработок и обладает следующей функциональностью:

1. Пуск и останов двигателя.
2. Плавное регулирование частоты вращения двигателя в диапазоне - 200 до 2000 об/мин.
3. Стабилизация заданной частоты вращения.
4. Регистрация основных параметров двигателя.
5. Передача сигналов управления.
6. Индикация перегрузок и неисправностей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Усольцев А.А. Частотное управление асинхронными двигателями / Учебное пособие. СПб: СПбГУ ИТМО, 2006, – 94 с.

2. Калачев Ю.Н. Векторное регулирование (заметки практика) [Электронный ресурс] – Режим доступа: http://privod.news/files/kniga_www_3.pdf, свободный (дата обращения: 05.05.2021).

3. Модуль управления SINAMICS DCM [Электронный ресурс]. – Режим доступа: https://siemens-drives.ru/files/lists/s_Directories/72_Files_1351230193_intsruksiya-po-ekspluatatsii.pdf, свободный (дата обращения: 07.05.2021).

4. Toshiba – Стандартные преобразователи частоты / Серия S15 [Электронный ресурс]. – Режим доступа: <http://www.toshiba.ru/products/products-for-business/promyshlennoe-oborudovanie/standartnye-preobrazovately-chastoty/seriya-s15/>, свободный (дата обращения: 07.05.2021).

УДК 519.95

Бабкин Д.И.

АЛГОРИТМЫ И МЕТОДЫ ОБРАБОТКИ СИГНАЛОВ С СЕРДЦА

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

В данной работе рассматриваются и анализируется метод записи и оценки сигналов с сердца, а также разработан виртуальный блок считывания данных.

Методика проведения фонокардиографических исследований заключается в следующем. Фонокардиограмму регистрируют в помещении, изолированном от посторонних шумов. С помощью предварительной

аускультации и клинических данных определяют выбор точек записи и положение пациента (лежа, на боку, сидя, стоя, после физической нагрузки), а иногда применение фармакологических средств (амилнитрата, нитроглицерина, эуфиллина). Фонокардиограмму записывают обычно на выдохе, а при необходимости – на высоте вдоха и при свободном дыхании. Для получения качественной фонокардиограммы большое значение имеет фиксация микрофона - микрофон должен плотно, но не сильно прилегать к поверхности грудной клетки. После записи исследуются временные характеристики звуков сердца: их последовательность, продолжительность, конфигурация (характер изменения во времени амплитуды звукового сигнала – время нарастания, максимума, убывания и т.д.), а также наличие и длительность интервалов между ними.

Перспективным направлением является использование адаптивных методов фильтрации ФКГ, а также разработка эвристических алгоритмов, автоматически выбирающих оптимальные алгоритмы обработки результатов фонокардиографии; ФКГ имеет большие перспективы применения веб-технологий в системах дистанционного биомониторинга, позволяющих осуществлять удаленное наблюдение за пациентами, и автоматическую передачу информации о их состоянии в телемедицинский центр с последующим автоматическим анализом информации и формированием сообщений врачам.

Часто используется одновременная запись ЭКГ и ФКГ, при этом сигнал ЭКГ служит для разбиения ФКГ сигнала на систолическую и диастолическую части. Сигналы ЭКГ и ФКГ различаются по своей физической природе: первый из них электрический сигнал, а второй механический (звук, вибрация). На рисунке 1 представлена одновременная запись ФКГ и ЭКГ, а также показана разметка систолической (SYS) и диастолической (DIAS) частей сигнала ФКГ с использованием QRS-комплекса ЭКГ.

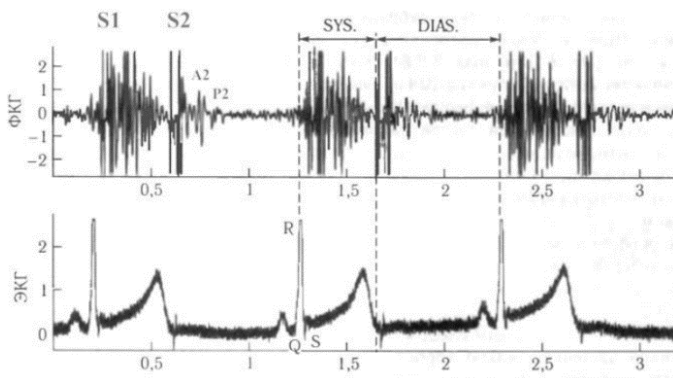


Рис. 1 Одновременная запись ЭКГ и ФКГ
Разработка блок-диаграммы виртуального прибора

Вывод исходного сигнала

Исходный сигнал считывается с помощью элемента «Path», преобразуется в формат понятный для программы и выводится на график ,сам файл задается через лицевую панель рисунок 2

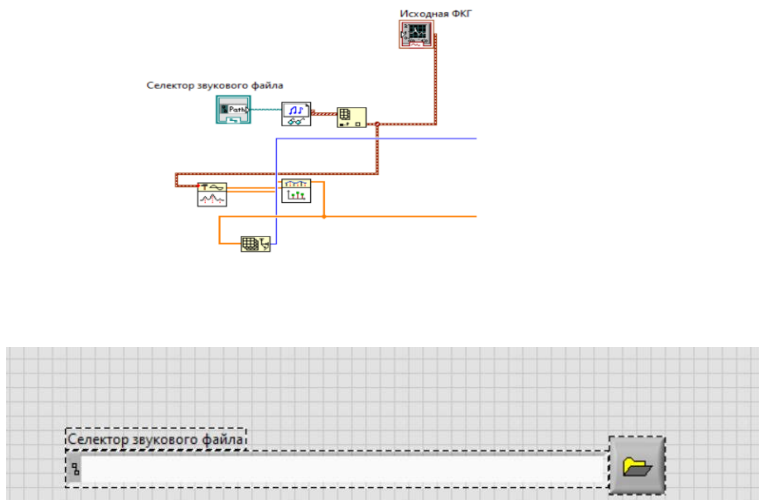


Рис. 2. Считывание звукового файла и преобразование исходного сигнала

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Краснов Л.А. Фонокардиография. Технические средства электронной и компьютерной диагностики в медицине: учеб.пособие/ Л.А. Краснов, В. П. Олейник. Х. : Нац. аэрокосм. ун-т им. Н.Е. Жуковского «Харьк. авиац. ин-т». 2013. 64 с.
2. Программно-аппаратный комплекс для фонокардиографических исследований. В.М. Никитин, В.В. Муромцев, Д.А. Анохин. Белгородский государственный национальный исследовательский университет. 2012. №13 (132). Выпуск 23/1 УДК 004.3
3. Масленников М.Ю, Соболев Е.А. Справочник разработчика и конструктора РЭА. Элементная база. М.: ТОО «Прибор», 1993.-158 с.

УДК 004.432

Агафонов П.А., Асташкин М.С.

КОМПЬЮТЕРНАЯ РЕАЛИЗАЦИЯ НАСТОЛЬНОЙ ИГРЫ
«СОПРОТИВЛЕНИЕ» НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ KOTLIN

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

Рассматриваются возможности языка Kotlin для разработки web-приложений и его основные преимущества. Кратко описывается архитектура разрабатываемой игры, а также используемые инструменты и средства.

Введение. Настольные игры – игры, основанные на разного рода манипуляциях относительно небольшим набором предметов, например, карты, кости, домино, которые могут целиком разместиться на столе (на специальном поле) или в руках играющих. В отличие от многих игр, данный тип игры не требует активной физической деятельности, наличия сложного технического оборудования, в них не существует рисков при проигрыше.

Недостатками настольных игр являются: габариты коробки, в которой хранятся основные элементы для игры; рутинная работа, которая возникает при подготовке начала игры; возможное наличие ведущего, который не может играть в игру, которую он ведет.

Создание компьютерных реализаций настольных игр актуально, поскольку они избавят от перечисленных проблем. Было предложено реализовать компьютерную версию настольной игры «Сопrotивление», а для создания игры использовать язык программирования Kotlin и различные фреймворки и библиотеки.

Kotlin – лаконичный, безопасный и прагматичный язык, совместимый с Java. Одним из основных разработчиков языка является русский офис компании JetBrains, расположенный в Санкт-Петербурге. Его можно использовать практически везде, где применяется Java: для разработки серверных приложений, приложений для Android и многого другого. Kotlin прекрасно работает со всеми существующими библиотеками и фреймворками, написанными на Java, не уступая последнему в производительности.

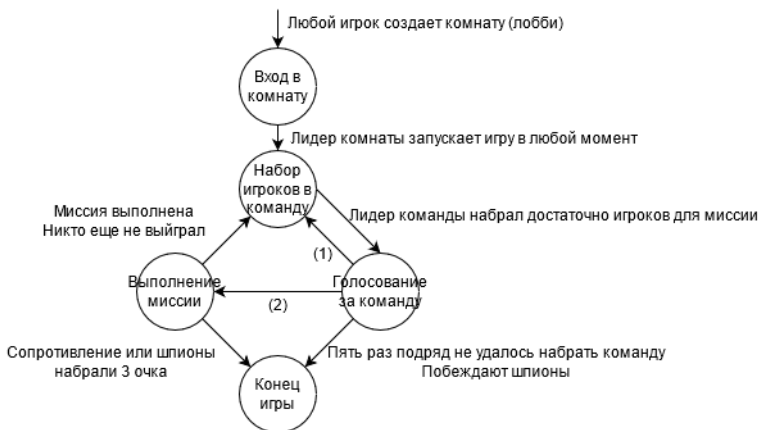
Настольная игра «Сопrotивление»

«Сопrotивление» – дедуктивная психологическая игра. Игроки либо принадлежат к сопротивлению, целью которого является свержение деспотичного правителя, либо становятся шпионами, стремящимися расстроить планы сопротивления.

Сопrotивление побеждает в случае успеха 3 миссий. Победа достается шпионам, если 3 миссии были провалены. Шпионы также могут победить на любом этапе игры, если сопротивлению не удаётся организовать команду по выполнению миссии (5 срывов голосования за выполнение миссии) [1].

Основным правилом данной игры является возможность игроков в любое время высказать свое мнение. Дискуссии, ложные обещания, интуитивные озарения, социальное взаимодействие и логические умозаключения важны для победы в этой игре.

По правилам игры можно составить граф, представленный на рис. 1, в котором вершины – состояния игры (определенный этап), а дуга – условия перехода между состояниями.



1) Большинство проголосовало ПРОТИВ

2) Большинство проголосовало ЗА

Рис. 1. Граф состояний игры и условий перехода между ними

Язык программирования Kotlin. Основная цель языка Kotlin – предложить более компактную, производительную и безопасную альтернативу языку Java, пригодную для использования везде, где сегодня применяется Java. Java – чрезвычайно популярный язык, который используется в самых разных окружениях, начиная от смарт-карт (технология Java Card) до крупнейших вычислительных центров таких компаний, как Google, Twitter и LinkedIn [2]. В большинстве таких окружений применение Kotlin способно помочь разработчикам достигать своих целей меньшим объемом кода и избегая многих неприятностей.

Наиболее типичные области применения Kotlin: разработка кода, работающего на стороне сервера (как правило, серверной части веб-приложений); создание приложений, работающих на устройствах Android.

Также существуют технологии Kotlin/JS и Kotlin/Native, которые позволяют компилировать код в JavaScript и в исполняемый файл определенной операционной системы без использования виртуальной машины.

Язык программирования Kotlin – статически типизированный язык программирования. Это означает, что тип каждого выражения в програм-

ме известен во время компиляции, и компилятор может проверить, что методы и поля, к которым вы обращаетесь, действительно существуют в используемых объектах.

В отличие от Java, Kotlin не требует явно указывать тип каждой переменной. В большинстве случаев тип переменной может быть определен автоматически из контекста. Благодаря поддержке вывода типов в Kotlin исчезает излишняя избыточность статически типизированного кода, поскольку больше не нужно объявлять типы явно.

Наиболее важным нововведением в Kotlin является поддержка типов, допускающих значения null (nullable types), которая позволяет писать более надежные программы за счет выявления потенциальных ошибок обращения к пустому указателю на этапе компиляции.

Язык программирования Kotlin является мультипарадигмальным языком программирования, поэтому он обладает возможностями для поддержки функционального программирования: функциональные типы, позволяющие функциям принимать или возвращать другие функции; лямбда-выражения, упрощающие передачу фрагментов кода; классы данных, предоставляющие емкий синтаксис для создания неизменяемых объектов-значений; обширный набор средств в стандартной библиотеке для работы с объектами и коллекциями в функциональном стиле.

Реализация настольной игры. Клиентская часть игры реализована в виде telegram-бота, то есть всё управление происходит через сообщения и кнопки. Для хранения информации об игроках и играх используется СУБД PostgreSQL. При разработке были выделены следующие модели: игроки (players), игры (games), история игр (games_history). На рис. 2. Представлена диаграмма базы данных.

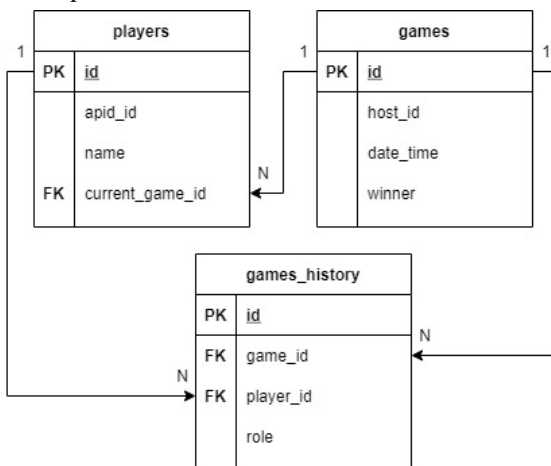


Рис. 2. Диаграмма база данных

Основные библиотеки и фреймворки, используемые для реализации:

1. ktor – асинхронный веб-фреймворк (с его помощью развернуты веб-сервера для сервиса, управляющего игрой, и telegram-бота, а также http клиент для организации связи между клиентской и серверной частями);
2. exposed – используется для работы с элементами СУБД в объектном виде (ORM);
3. flyway – библиотека для миграций базы данных;
4. kodein – используется для Dependency Injection;
5. kotlin Telegram Bot – обертка для удобного взаимодействия с telegram API через Kotlin Domain-Specific Language (DSL).

Архитектуру приложения можно представить в виде трёх модулей: клиентская часть (telegram-бот), веб-сервер, игровая логика. Визуальное представление архитектуры приложения показано на рис. 3.

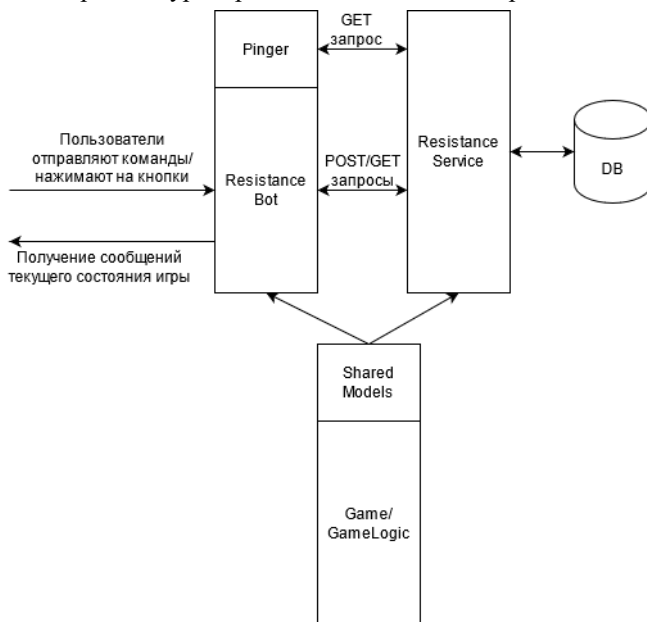


Рис. 3. Архитектура приложения

Также была предусмотрена ситуация, когда один из игроков блокирует telegram-бота, поэтому был разработан небольшой модуль, который с определенным интервалом отправляет сообщение всем игрокам, если при отправке сообщения произошла ошибка, то игрок заблокировал telegram-бота или удалил его, следовательно игру можно досрочно завершать.

Пример диалога с telegram-ботом представлен на рис. 4.

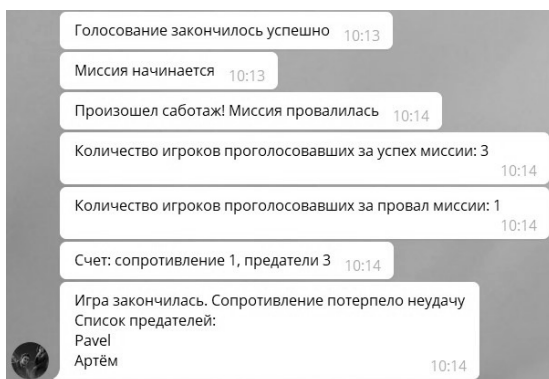


Рис. 4. Диалог с telegram-ботом

Заключение. В результате работы была реализована компьютерная версия настольной игры «Сопротивление». Веб-приложение, база данных и telegram-бот были развернуты на облачном сервисе Heroku с использованием Docker-контейнеров. Приложение доступно в телеграмм-канале по тегу @the_resistance_game_telegram_bot. Также были приобретены навыки работы с современными фреймворками для разработки веб-приложений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. https://hobbygames.ru/download/rules/Soprotivlenie_2_izdanie_Rules.pdf (дата обращения 10.06.2021)
2. <https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/> (дата обращения 10.06.2021)

УДК 004 ГРНТИ 28.23

Малистова М.Р.

ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРИМЕНЯЮЩЕГО АНАЛИЗ ТЕКСТОВЫХ ДАННЫХ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

В работе рассматриваются основные особенности программного обеспечения для анализа текстовых данных. Приводятся также их достоинства, недостатки, а также виды программ и их основные различия.

Данные являются неотъемлемой частью нашей жизни, но сами по себе неструктурированные данные несут мало ценности, поэтому чтобы использовать данные для своих целей и нужд, их нужно структурировать и анализировать. Но анализировать данные вручную довольно сложно, долго и можно сделать при этом довольно много ошибок. Для того, чтобы облегчить эту задачу, используется различное программное обеспечение для анализа текстовых данных. У таких программ есть ряд особенностей,

преимуществ и недостатков, которые надо рассмотреть, чтобы выбрать наиболее подходящую для цели анализа программу.

Интеллектуальный анализ текста - это процесс сортировки и анализа необработанных текстовых данных для получения практических сведений. Он включает извлечение значимой информации из больших объемов неструктурированных данных, таких как обзоры продуктов, электронные письма, твиты, заявки в службу поддержки и результаты опросов [1].

Анализ включает набор методов, которые структурируют поступающую информацию в текстовом формате, например, обратная связь с клиентами в виде произвольного текста. Цель состоит в том, чтобы преобразовать неструктурированный текст в значимые структурированные данные для поддержки бизнес-анализа и принятия решений. Структура, созданная программным обеспечением, создается с помощью таких методов, как анализ тональности, языковой анализ или просто просмотр ключевых фраз.

Существует масса текстовой информации, но текст – это неструктурированные данные, и для извлечения полезной информации из этих данных требуются средства, так как обработка и систематизация огромных объемов текстовых данных вручную не только требует много времени, но это также дорого, ненадежно и утомительно. Компании получают огромные объемы неструктурированных данных в виде текста (электронные письма, разговоры в социальных сетях, чаты), анализ которых может быть чрезвычайно сложным и долгим.

Большинство инструментов используют обработку естественного языка и / или методы машинного обучения для анализа текста. Также используются статистические методы, нейронные сети, методы классификации текста и документов, лингвистическое обучение, семантический анализ и прогнозное моделирование. Методы анализа текста и частота их использования представлены на рис. 1.

Программное обеспечение для анализа текста используется для анализа всех видов текстов. Существуют разные виды такого программного обеспечения: собственные коммерческие инструменты для анализа данных, инструменты интеллектуального анализа текста с открытым исходным кодом, инструменты интеллектуального анализа текста в Интернете, которые можно запустить с самого веб-сайта [2].

Программы для анализа текстовых данных обладают рядом преимуществ, которые помогают облегчить работу с данными [3]. Различные программные обеспечения обладают разными преимуществами, но некоторые присутствуют почти во всех программах для анализа текста, например:

- Быстрая интерпретация и анализ больших объемов необработанных данных, таких как электронные письма, чаты, твиты, сообщения и обзоры, за несколько минут. Приложения для интеллектуального анализа текста

помогают сэкономить время, поэтому можно сосредоточить свои ресурсы на более важных задачах.

- **Отслеживание комментариев и отзывов клиентов в режиме реального времени:** комментарии, отзывы и отзывы клиентов могут появляться в любом месте в Интернете, что затрудняет их отслеживание. Программное обеспечение для анализа текста может отслеживать комментарии и отзывы клиентов в режиме 24/7 и автоматически отмечать негативные твиты или отзывы, чтобы помочь предпринять необходимые действия в режиме реального времени.

- **Помощь в принятии решений:** программное обеспечение помогает определить, нужно ли вносить какие-либо изменения в существующие бизнес-операции. Можно использовать полученную информацию, чтобы подтолкнуть к изменению процесса и принимать бизнес-решения на основе данных.



Рис. 1. Методы анализа текстовых данных

Программное обеспечение для анализа текста используется для разных целей. Например, некоторые используют его в исследовательских целях, когда есть много расшифрованных данных, а программное обеспечение помогает структурировать эти данные [4].

Программное обеспечение для текстовой аналитики также используется в коммерческих целях. Компании, которые получают отзывы клиентов в открытом текстовом формате, являются идеальными кандидатами на такое программное обеспечение и могут получить много преимуществ, используя его для своих целей.

Программы для анализа текстовых данных бывают разными и могут сильно отличаться друг от друга, но у всех у них есть общие особенности [5], перечислим самые основные из них:

- **Извлечение текстовых данных из таких каналов, как социальные сети, онлайн-чат и электронные письма, которые являются хранилищами взаимодействий с клиентами.**

- Анализ отзывов и комментариев клиентов с помощью механизмов оценки, которые классифицируют настроения клиентов как положительные, отрицательные или нейтральные.
- Отчетность: использование диаграмм, графиков и других визуальных элементов для создания отчетов о настроениях клиентов, категоризации ключевых слов, результатах опросов и многом другом.

Помимо сходств, программы имеют много отличий, и в зависимости от целей, которые преследует аналитик при исследовании данных и от вида самих данных, он должен выбирать подходящее приложение для анализа данных.

Программы для анализа текстовых данных могут различаться по следующим пунктам [6]:

1) Аналитика и отчеты

Часто у компании или бизнеса есть ключевые показатели эффективности, над которыми они работают, и возможность программы для анализа данных связывать полученный результат с такими показателями очень облегчает работу.

Также важно, чтобы данные, полученные после анализа, отображались удобно и понятно, и была возможность подстраивать их отображение так, как это нужно и так, как выглядит более наглядно.

2) Действенность

Один из наиболее важных результатов, который можно получить от программного обеспечения для анализа текста, - это знать, нужно ли вносить какие-либо изменения в текущие способы работы.

Программное обеспечение для анализа текста не только классифицирует и структурирует данные, некоторые могут даже сказать, что нужно делать дальше.

На основе данных, полученных с помощью программного обеспечения для анализа текста, необходимо подумать, насколько легко определить приоритеты и принять меры по только что обнаруженным проблемам. Программа может расставить эти приоритеты и указать, какие проблемы нужно исправить и улучшить.

3) Удобство пользователю

Некоторое программное обеспечение для анализа может быть более сложным с технической точки зрения, чем другое, а для некоторых может даже потребоваться, чтобы пользователь программы умел выполнять базовое кодирование. Приложения разной технической сложности подходят для аналитиков с разным уровнем подготовки и разным уровнем технических знаний.

4) Анализ текста на разных языках

Большинство программ для анализа текста могут анализировать текст на русском или английском языке. Но иногда этого бывает недостаточно и

может потребоваться программное обеспечение, способное анализировать текст на некоторых дополнительных языках.

Некоторое программное обеспечение может предлагать текстовую аналитику на более популярных языках, таких как английский, французский и испанский.

Наличие анализа текстовых данных на многих языках особенно важно, если речь идёт о международной компании, которая может получать информацию, отзывы и сообщения от клиентов из разных стран, говорящих на разных языках.

5) Сравнение фильтров

В бизнесе важно иметь возможность измерять свои результаты и сравнивать их с предыдущими результатами или результатами из другой области бизнеса или типа продукта, чтобы можно было искать тенденции. Функции сравнения добавляют значительную ценность программному обеспечению для анализа текста.

6) Время установки

Установка программы и получение первых результатов может занимать разное время у разных приложений. В одних случаях требуется получить и проанализировать данные немедленно, тогда как в других случаях промедление в несколько дней или даже недель не играет большой роли.

7) Настройка тэгов

Чтобы сделать результат оптимальным для отрасли или конкретной компании, большинство сервисов требует, чтобы определенные слова, пары слов и более длинные словосочетания были связаны с категориями (это называется тегированием), чтобы машина начала обучение.

Как видно, данные являются очень важной частью современного мира, и для извлечения полезной информации из этих данных, их надо анализировать. Без специального программного обеспечения это сделать сложно. В то же время программное обеспечение для анализа текстовых данных имеет свои особенности и технологии, которые помогают сделать исследование текста проще удобнее и нагляднее. Также при выборе программного обеспечения для анализа текста в зависимости от того, для чего используется приложение, нужно учитывать и принимать во внимание многие факторы. Выбор правильного и наиболее подходящего для целей аналитика программного обеспечения поможет улучшить качество анализа.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Feldman, Ronen, and James Sanger. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, 2007.

2. Dekhtyar, Alexander, Jane Huffman Hayes, and Tim Menzies. «Text is software too». MSR 2004: International Workshop on Mining Software Repositories at ICSE'04: Edinburgh, Scotland. 2004.

3. Malhotra, Ruchika, et al. «Severity Assessment of Software Defect Reports using Text Classification». International Journal of Computer Applications 83.11 (2013).

4. Jurek, Anna, Maurice D. Mulvenna, and Yaxin Bi. «Improved lexicon-based sentiment analysis for social media analytics». Security Informatics 4.1 (2015): 1-13.

5. Gamon, M., Aue, A., Corston-Oliver, S., & Ringger, E. (2005). Pulse: Mining customer opinions from free text. In Advances in Intelligent Data Analysis VI (pp. 121-132). Springer Berlin Heidelberg.

6. Bragge, Johanna, and Jan Storgårds. «Profiling academic research on digital games using text mining tools». Proceedings of DiGRA 2007 Conference.

УДК 004.032.6; ГРНТИ 20.23

Лукьянова Т.А.

ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРИМЕНЯЮЩЕГО ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ МУЛЬТИМЕДИЙНЫХ ДАННЫХ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

В работе рассматриваются особенности программного обеспечения при интеллектуальном анализе мультимедийных данных.

Мультимедийный контент сложно анализировать с помощью классических инструментов, поэтому приходится прибегать к использованию решений, основанных на нейронных сетях и больших данных.

Для корректной и быстрой работы с большим объемом информации можно использовать графовую базу данных.

Графовые базы данных. Базы данных графов предназначены для хранения связей и навигации по ним. Такие базы данных используют узлы для хранения сущностей данных и ребра (дуги) для хранения связей между сущностями.

Как узлы, так и ребра могут иметь записанные свойства, а ребра имеют тип, определяющий характер связи. Граф может быть пройден либо по выбранным типам ребер, либо по всему графу. Проход по связям в графовых базах данных происходит очень быстро, так как взаимоотношения между узлами не вычисляются при выполнении запроса, а хранятся в базе данных. У графовых баз данных есть преимущества в таких приложениях, как системы обнаружения мошенничества, социальные сети и рекомендательные службы, где необходимо создавать связи между данными и быстро запрашивать их.

На рис. 1 показан пример графа социальной сети. Если у вас есть данные о людях (узлах) и отношениях между ними (ребрах), то можно узнать, кто такие «друзья друзей» конкретного человека.

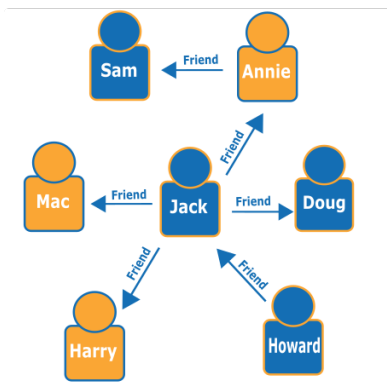


Рис. 1. Пример графа социальной сети

Есть несколько особенностей баз данных графов, которые следует учитывать при рассмотрении технологии, которую они используют [1]: принцип хранения и механизм обработки.

Достоинства графовых баз данных [1]:

1. Производительность

Одной из причин выбора графовой базы данных является большой прирост производительности при работе с взаимосвязанными данными по сравнению с реляционными базами данных. В отличие от реляционных баз данных, где учет взаимосвязей снижает производительность запросов на больших наборах данных, производительность графовых баз данных с увеличением объема хранимых данных не меняется. Это происходит потому, что запросы локализируются в определенной части графа. В результате время выполнения каждого запроса зависит от размера части графа, которую необходимо пройти, чтобы удовлетворить запрос, а не от общего размера графа.

2. Гибкость

Расширяемость, характерная для графов, означает, что можно добавлять новые типы связей, новые узлы в существующую структуру, не нарушая существующих запросов и функциональности приложения. Это хорошо влияет на производительность разработки. Благодаря гибкости графовой модели нет необходимости предварительно моделировать задачу в мельчайших деталях.

Контент мультимедиа. Мультимедийный контент может быть рассмотрен в нескольких вариантах:

1. Как информация, которая говорит о предпочтениях человека. Цель анализа – сравнение мультимедийных объектов с образцами, доступными в Интернете и в рамках социального профиля (например, определение музыкальных предпочтений по нескольким аудиозаписям).

2. Как контент, который содержит в себе информацию о данном человеке. Задачей анализа является извлечение информации из самого объекта (например, распознавание лиц и эмоций на фотографии).

Для обработки мультимедийной информации первой категории следует использовать контентный метод анализа [2], сутью которого является деление данных на составные части и непосредственное их сравнение. К примеру, следующий алгоритм будет использоваться для определения ваших музыкальных предпочтений. Аудиозаписи, которые существуют в базе социального профиля, проверяются на наличие тегов Исполнителя и Жанра музыки. Если они найдены, то записываются в таблицу предпочтений. В противном случае аудиозапись сравнивается с образцами из Интернета, а при обнаружении совпадения теги заносятся в таблицу. После того, как все имеющиеся аудиозаписи обработаны, подсчитываются теги и делается вывод о преобладании того или иного жанра или исполнителя в выборке.

Для исследования второй категории применяется контентно-интерпретационный метод, в соответствии с которым компонентам мультимедийных данных на формальном языке присваиваются понятия, а затем между ними выстраиваются связи.

Основным направлением анализа аудиозаписей является распознавание речи и ее интонации. Результатами являются: звуковые характеристики голоса; распознанные тексты, связанные с оригинальной записью.

Подходы к распознаванию графической информации делятся на категории: методы перебора, искусственные нейронные сети и поиск контуров объекта с последующим исследованием их свойств. Следующим шагом после того, как изображение будет обработано, будет сведение данных в таблицу результатов, содержащую следующие параметры: распознанные лица, их эмоции, список надписей, данные об окружении (обнаруженные объекты), информация об изображении (размер, дата создания и т.п.).

Для анализа изображений или аудиозаписей часто применяют нейронные сети.

Нейронная сеть. Нейронная сеть – это структура, которая состоит из искусственных нейронов, связанных определенным образом между собой и внешней средой посредством связей, каждая из которых имеет определённый коэффициент, на который умножается поступающее через нее значение (эти коэффициенты называют весами) [3].

Нейронные сети могут быть реализованы как программно, так и аппаратно (нейронные чипы, нейрокомпьютеры).

При работе нейронной сети происходит изменение данных, конкретный вид которых определяется весами межнейронных связей, типом активационной функции нейронов, архитектурой и конфигурацией сети.

Нейронные сети — это модели, которые основаны на машинном обучении, то есть они приобретают требуемые свойства в процессе обучения,

закрывающегося в итеративной корректировке весов сети в соответствии с определенным правилом, называемым алгоритмом обучения [4].

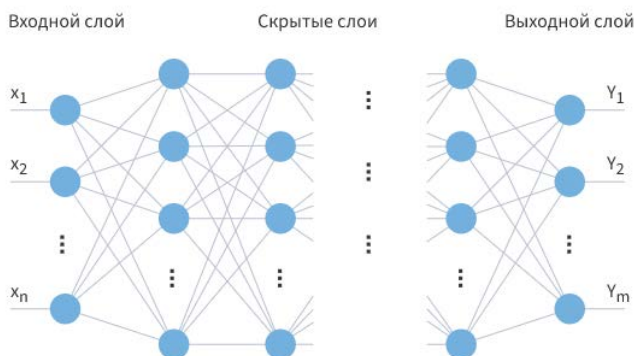


Рис. 2. Схема многослойной нейронной сети

При построении нейронных сетей может использоваться обучение как с учителем (для многослойных персептронов), так и без учителя (для сетей Кохонена).

Главные задачи, которые решают нейронные сети в анализе данных:

Аппроксимация функций – восстановление функциональных зависимостей из обучающих данных.

Классификация – определение принадлежности входного образа (объекта), представленного вектором признаков, к одному из предварительно заданных классов.

Кластеризация – группировка объектов на основе близости их свойств [5].

Прогнозирование – предсказание значения $y(t+1)$ при заданной последовательности $y(t1), y(t2), \dots, y(tn)$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Robinson, I. and Webber, J. and Eifrem, E. Graph Databases. – O'Reilly Media, 2013.
2. Чураков А.Н. Компьютерный контент-анализ. М.: ИСАН, 1996.
3. Рашид Тарик. Создаем нейронную сеть.: пер. с англ. А.Г. Гузикевич, ред. С.Н. Тригуб – СПб.: ООО «Диалектика», 2019.
4. Нейронные сети – учебник [Электронный ресурс]. – Режим доступа: <https://neuralnet.info/book/> – Дата доступа: 27.02.2021.
5. Bradley P., Fayyad U., Reina C. Scaling Clustering Algorithms to Large Databases, Proc. 4th Int'l Conf. Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, Calif., 1998.

УДК 004.023

Кузовлев Р.Г.

СРАВНЕНИЕ ПРООБРАЗОВ ВХОДНЫХ СИГНАЛОВ ДЛЯ РАСПОЗНАВАНИЯ ФИГУР ТЕХНИЧЕСКОГО АНАЛИЗА С ПРИМЕНЕНИЕМ НЕЙРОННЫХ СЕТЕЙ

Рязанский государственный радиотехнический университет им. В. Ф. Уткина, г. Рязань

В статье кратко рассматриваются преимущества применения нейронных сетей для анализа финансовых временных рядов и на их основе приводится сравнение вариантов формирования входных сигналов для решения задачи распознавания фигур технического анализа с применением нейронных сетей.

Существуют два основных направления технического анализа: распознавание фигур (изображений, паттернов, моделей), использование индикаторов.

Фигуры технического анализа делятся на три категории: фигуры разворота, продолжения тренда и неопределенные, такие фигуры, которые сложно отнести к двум первым группам из-за специфических особенностей [1]. Основная сложность использования метода распознавания паттернов, является их интерпретация на графике цен. Следует учитывать, что на одном интервале времени может формироваться фигура «флаг», а на другом, в этот же момент, формироваться «голова и плечи». Это следует обязательно рассматривать в выборе прообраза входных данных, так как важно определять модель в процессе ее формирования, а не после того, как она уже сформировалась и временной ряд начал изображать новую фигуру.

Научно-технический прогресс привел к появлению эффективного способа в области прогнозирования с помощью нейронных сетей, когда компьютер обучается сам. Основными параметрами для тренировки сети являются: время, затраченное на обучение и качество подбора весовых коэффициентов. Процесс обучение сети нужен для поиска наилучшего набора весов, так как параметры этого набора определяют точность предсказания [2].

Нейросеть является многослойной сетевой структурой нейронов, соединенных между собой и сгруппированных в слои.

Существует ряд важных преимуществ нейронных сетей перед традиционными вычислительными методами, которые следует учитывать при формировании входных сигналов:

1. Нейронные сети могут работать с неполными данными, так как благодаря обучению способны решать задачи с неизвестными закономерностями и зависимостями между данными.

2. Нейронные сети имеют устойчивость к помехам во входных данных. Сеть способна самостоятельно определять неинформативные пара-

метры для анализа и отфильтровывать их, что отбрасывает необходимость предварительного анализа входных данных.

3. Одновременная обработка входных данных многими нейронами обеспечивают нейронной сети высокую производительность, что дает преимущество в скорости решения задач, перед большинством других алгоритмов.

4. Способность обучения на основе данных, дает возможность нейронными сетям адаптироваться к изменяющейся среде, что очень важно при анализе изменений рыночной ситуации.

5. Нейронные сети имеют аффинную независимость к входным данным, в частности к смещению фигур технического анализа, масштабу, углам. Это исходит из второго преимущества, так как эти искажения можно считать шумом.

6. Способность работать с огромным количеством данных. Для курса цен может быть сформирована огромная история длительностью несколько десятков лет, что образует миллионы входных примеров. Только нейронные сети способны справиться с таким объемом информации в приемлемое время.

Основываясь на приведенных преимуществах, нейронная сеть будет предпочтительнее ряда других методов вычисления в решении задачи распознавания фигур технического анализа с огромным количеством данных, в которых скрыты закономерности.

Предобработка данных является одним из главных моментов в анализе финансовых временных рядов. Здесь главной задачей является снижение избыточности данных, что способствует успешному обучению нейронной сети, а, следовательно, повышает точность предсказаний.

Первым способом формирования входных данных, является использование «свечек» или цен бара OHLC за определенный период, где Open это цена открытия бара, High – наивысшая цена, Low – минимальная цена, Close – цена закрытия. Как правило, период обычно берется от 30 до 100 баров [3]. Преимуществами данного типа входных данных является подача полной информации на определенном участке графика. Однако существенным недостатком является серьезные вычислительные затраты, так как на вход подается массив входных данных огромного объема, следовательно, потребуется создать нейронную сеть большой размерности.

На вход также можно использовать значения популярного сигнального технического индикатора ZigZag. Он соединяет наиболее значимые экстремумы на графике прямой линией, отфильтровывая мелкие колебания [4]. В качестве входной информации можно использовать координаты начала и окончания отрезка и угол наклона отрезка. Данный индикатор наглядно отображает наиболее существенные развороты и откаты рынка. Преимуществами данного способа представления входных сигналов является значительно меньшая размерность сети, а также исключение «шума»

во входных данных. С другой стороны, избавление от шума может привести к потере значимых точек или отрезков на графике для определения фигур технического анализа. Пример индикатора ZigZag представлен на рис. 1.



Рис. 1. Технический индикатор ZigZag

Еще одним способом подачи входных данных можно выделить применение индикатора Fractals [5]. Он выполняет анализ ценовых фигур, которые образуются на свечном графике и отображают точки разворота рыночной тенденции. Главным условием применения Fractals является, чтобы центральный бар на определенном отрезке времени был выше или ниже всех остальных, что и будет являться значением фрактала. Считается, что временной отрезок должен быть не менее 5 баров и чем длиннее отрезок, тем результаты надежнее. Фрактал вверх – это серия из минимум пяти последовательных баров, в которой перед самым высоким максимумом и за ним находятся бары с более низкими максимумами. Фрактал вниз – серия баров, в которой перед самым низким минимумом и за ним находятся бары с более высокими минимумами. На графике фракталы имеют значения High и Low и отмечены стрелками вверх или вниз. В общем случае данный индикатор показывает точки экстремума на определенном периоде. Из достоинств можно отметить еще меньшую размерность сети, чем у индикатора ZigZag, так как теперь мы храним только точки экстремума, а не отрезки. Однако без сочетания с другими индикаторами или какой-либо модификации данный индикатор будет иметь довольно низкую точность, ведь гораздо сложнее определять модели технического анализа по точкам экстремума, чем по отрезкам, с учетом замеча-

ния формирования нескольких фигур одновременно на разных интервалах времени. Пример индикатора Fractals представлен на рис. 2.



Рис. 2. Технический индикатор Fractals

Таким образом, с учетом всех преимуществ и недостатков, наиболее многообещающим типом входных данных являются значения входных данных на основе индикатора ZigZag. Различные модификации данного индикатора или архитектуры сети, помогут оптимизировать проблему потери значимых вершин на графике для определения моделей технического анализа. Входные данные на основе OHLC можно использовать для анализа небольшого промежутка времени, чтобы минимизировать размерность сети. Однако следует соблюдать баланс между размерности сети и периодом анализа, так как обе эти величины влияют на точность предсказаний. Наименее перспективным видом входных сигналов, без каких-либо изменений являются данные на основе индикатора Fractals. Возможно, эффективность данного индикатора возрастет с использованием в сочетании с другими индикаторами, но скорее всего размерность входных данных станет приблизительно равной или больше, чем для индикатора ZigZag.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Тенсин П. Самые распространенные фигуры технического анализа. [Электронный ресурс] / П. Тенсин // BCS Express. – 2019. – Режим доступа: <https://bcs-express.ru/novosti-i-analitika/samyie-rasprostranennye-figury-tekhicheskogo-analiza>.

2. Малышенко К.А. Использование нейросетей для целей прогнозирования фондового рынка [Электронный ресурс] / К.А. Малышенко, М.В. Анашкина // Эффективная экономика. – 2014. – Т. 2, № 975. – Режим доступа: <http://www.economy.nauka.com.ua/?op=1&z=2744>.

3. Manurung, A. H. Algorithm and modeling of stock prices forecasting based on long short-term memory / A.H. Manurung // ICIC Express Letters. – 2018. – Т. 12, № 12. – С. 1277.

4. Chen, J. Zig Zag Indicator [Электронный ресурс] / J. CHEN // Investopedia. – 2021. – Режим доступа: https://doi.org/https://www.investopedia.com/terms/z/zig_zag_indicator.asp

5. Фрактал (Fractals) [Электронный ресурс] // Открытый журнал. – 2020. – Режим доступа: <https://journal.open-broker.ru/trading/fractals/>.

УДК 004.33(035)

Антипов В.А., Антипов О.В., Солдатов В.В.

МОБИЛЬНЫЕ АБОНЕНТЫ ВИРТУАЛЬНЫХ ОРГАНИЗАЦИЙ

Рязанский государственный радиотехнический университет им. В.Ф. Уткина, г. Рязань

Публикация/Подписка (pub/sub) является перспективной архитектурой промежуточного программного обеспечения (ПО) виртуальной организации, поддерживающей мобильное взаимодействие, реконфигурируемость и его эволюцию. Существующее промежуточное ПО Публикации/Подписки было оптимизировано для стационарных систем, где пользователи, а также базовая структура системы были фиксированными в пространстве. В работе рассматривается вопрос о том, можно ли расширить существующее промежуточное ПО Публикации/Подписки для поддержки мобильных и независимых от местоположения приложений. Анализируются требования к таким приложениям и выделяются две ортогональные формы мобильности: системно-ориентированную физическую мобильность и логическую мобильность, ориентированную на приложения, в которых пользователи знают об изменении своего местоположения. Вводятся зависимые от местоположения подписки в качестве подходящего средства для использования возможностей парадигмы, основанной на событиях в мобильных приложениях.

Введение. Публикация/Подписка (pub/sub) считается ценной архитектурой промежуточного программного обеспечения виртуальных организаций, которая распространяет свободную связь и использует реконфигурируемость и эволюцию. До настоящего времени существующее промежуточное ПО Публикации/Подписки было оптимизировано для статических систем, где пользователи, а также базовая структура системы были фиксированными. Мы изучаем вопрос о том, можно ли расширить существующее промежуточное ПО Публикации/Подписки для поддержки мобильных и зависимых от местоположения приложений.

Проблемы с мобильностью в промежуточном ПО Публикации/Подписки. Мобильные клиенты имеют множество характеристик, среди которых необходимость отключаться от сети по разным причинам, например, по географическим, административным или по соображениям энергосбережения, быть все время подключенным к одному брокеру невозможно. Следовательно, необходимо учитывать, что клиенты будут время от времени отключаться от своего пограничного брокера. Промежуточное ПО должно быть связано с перемещением клиентов, а также должно обеспечивать возможность того, что отключенный клиент снова подключится к тому же или другому брокеру.

Первым шагом на пути к мобильности является расширение существующего промежуточного программного обеспечения Публикации/Подписки, допускающего перемещение клиентов, так, чтобы существующие приложения могли использоваться в мобильных средах.

Это означает, что интерфейсы для доступа к промежуточному программному обеспечению и приложениям сверху не требуется изменять. Что еще более важно, качество обслуживания, предлагаемого промежуточным программным обеспечением, не должно существенно ухудшаться. Результирующая прозрачность местоположения необходима для того, чтобы сделать существующие приложения мобильными, например, мониторинг котировок акций, плавно переносимых с ПК на корпоративный ПК. С другой стороны, будущие приложения не нуждаются в полной прозрачности, но полагаются на поддержку мобильности. В частности, поддержка мобильности должна сочетать нежелательные явления, такие как разъединенность и принуждение к желаемому поведению, осведомленность о местоположении в службах, основанных на локализации. Поэтому, расширение интерфейса промежуточного программного обеспечения Публикации/Подписки для облегчения определения местоположения является актуальной проблемой, поскольку большинство существующих работ сосредоточено только на прозрачности.

При роуминге клиенты меняют контекст (по крайней мере, некоторую его часть), в котором они работают, и могут захотеть отреагировать на эти изменения, например, адаптировать свои подписки. Однако соответствующая инфраструктурная поддержка должна избавить приложение от необходимости реагировать «вручную» ко всем изменениям. Промежуточное ПО скорее должно предлагать автоматическую адаптацию к изменениям контекста, то есть облегчать зависимость местоположения. Это приводит к другому понятию мобильности. Таким образом, мы различаем:

1. Физическую мобильность: клиент, который физически мобилен в течение определенного периода времени и имеет различных пограничных брокеров вдоль маршрута через инфраструктуру системы. Основной проблемой физической мобильности является прозрачность местоположения.

2. Логическую мобильность: клиент, который логически мобилен, знает об изменениях своего местоположения. Чтобы избавить клиента от ручной адаптации к новым местоположениям, основной задачей логической мобильности является автоматическое определение местоположения в промежуточном программном обеспечении Публикации/Подписки.

Физическая и логическая мобильность – это два ортогональных аспекта мобильности. Поскольку физическая компоновка систем Публикации/Подписки обычно фиксируется, и ее схема обычно не соответствует географическим реальностям, представляется разумным разделить два понятия мобильности. В этой статье мы предполагаем, что логическая мобильность является уточнением физической мобильности в том, что клиент остается подключенным к одному и тому же брокеру при логическом роуминге. Эти два понятия имеют разные требования к качеству обслуживания, поэтому разрабатываются различные решения для соответствия им.

Физическая мобильность. Физическая мобильность похожа на то, что в области мобильных вычислений называется терминальной мобильностью или роумингом. Клиент обращается к системе через определенное количество точек доступа (базовые станции GSM, точки доступа WLAN или пограничные брокеры). При физическом перемещении клиент может оказаться вне досягаемости одной точки доступа и перейти в зону доступа второй, которая не обязательно перекрывается. Мы не можем рассчитывать на полностью беспрепятственный доступ к брокерской сети, однако можно добиться более согласованных фаз связности, например, на ежедневном маршруте между домом и офисом. В этой установке мы анализируем качество требований к услугам с точки зрения клиентов роуминга:

1. *Интерфейс.* Очевидно, что интерфейс в системе Публикации/Подписки не должен меняться, поскольку устаревшие приложения не знают о мобильности.

2. *Полнота.* Несмотря на прерывистые разъединения, в конечном счете, промежуточное ПО Публикации/Подписки доставляет все уведомления для клиента. Это основное требование прозрачности.

3. *Заказ.* Адресант отправляет уведомления (FIFO).

4. *Ответная реакция.* Задержка перемещения клиента должна быть минимальной, чтобы максимизировать оперативность системы. Это необходимо учитывать при разработке протокола перемещения.

Возможные решения. Одно из решений – полагаться на мобильный IP [1] для подключения клиентов к пограничным брокерам, скрывая физическую мобильность на сетевом уровне. Однако недостатком этого является то, что связь также скрыта от промежуточного программного обеспечения Публикации/Подписки, которая далее не может извлекаться из любых мест доставки уведомлений или оптимизации маршрутизации. Тем самым, нарушается требование ответной реакции. Такой подход может быть осуществим только в том случае, если физическое и логическое рас-

положение данной системы полностью ортогонально. Другим решением для реализации физической мобильности может быть использование последовательностей sub-unsub-sub-вызовов для регистрации клиента у нового брокера. Когда клиент переходит от пограничного брокера B_1 в B_2 , он просто отписывается от B_1 и повторно подписывается на B_2 без поддержки в промежуточном ПО. Но клиент может не обнаружить, что покидает диапазон брокера и в этом случае не сможет отказаться от подписки в своем прежнем месте. Более того, клиент в течение всего времени отключения может пропустить несколько уведомлений или получить дубликаты, даже если в сети залиты уведомления, а изменение местоположения мгновенно. Эта проблема изображена на рис. 1. Следовательно, это решение не является полным [2].



Рис. 1. Отсутствие уведомлений в сценарии наводнения

Хотя физическая мобильность является скорее техническим нюансом, невидимым для приложения, логическая мобильность включает в себя информацию о местоположении.

Примером логической мобильности является перемещение клиента по дому или зданию, обслуживаемому только одним пограничным брокером. В этом случае пользователю должно быть интересно получать только те уведомления, которые относятся к комнате, в которой он находится в настоящее время. Стоит обратить внимание, что клиент может одновременно быть логически и физически мобильным.

Логически мобильный клиент, перемещающийся из одного места в другое, будет ожидать беспрепятственное изменения местоположения без заметного времени настройки после того, как он переместился, например, из своего офиса в конференц-зал по соседству. Адаптация какой-либо привязки с определением местоположения должна происходить «мгновенно». Интуитивно мы хотели бы испытать понятие подписки на «все, повсюду, все время» и повысить реактивность системы на перемещение клиентов.

Фильтры, зависящие от местоположения. Система Публикации/Подписки, которая предлагает фильтры, зависящие от местоположе-

ния, имеет тот же интерфейс, что и обычная система Публикации/Подписки (т.е. она предлагает `pub`, `sub`, `unsub`, примитивы уведомления). Однако при указании фильтров подписки для пар имя/значение, ссылающихся на «местоположение», она поддерживает новый примитив для указания таких вещей, как «все уведомления, где местоположение атрибута равно моему текущему местоположению». Точнее, мы постулируем конкретный маркер `myloc`, который можно использовать в подписке. Маркер обозначает определенный набор местоположений, зависящих от текущего положения клиента. Например, клиент может оформить подписку на все свободные парковочные места вблизи своего текущего местоположения следующим образом: `(service = "parking")`, `(location ∈ myloc)`, `(car-type ≥ "compact")`.

Набор местоположений, связанных с маркером, берется из определенного диапазона L мест. Этот набор зависит от приложения и может содержать, например, все комнаты здания, все улицы города или все географические координаты, заданные системой GPS с определенной степенью детализации. Учитывая уведомление с местоположением атрибута, подписка `(location ∈ myloc)` будет оцениваться как `true` для конкретного клиента в местоположении y тогда и только тогда, когда $x ∈ myloc(y)$, где `myloc(y)` – это определенный набор местоположений, связанных с y .

В этом случае мы говорим, что уведомление соответствует зависящему от местоположения фильтру. Простейшей формой `myloc(y)` является просто множество $\{y\}$. В этом случае уведомление соответствует подписке, если $x = y$. Но в примере автомобиля водитель, ищущий место для парковки, может захотеть указать: `(location = «не более двух кварталов от myloc»)`. В этом случае `myloc` соответствует всем элементам L , которые удовлетворяют этому требованию.

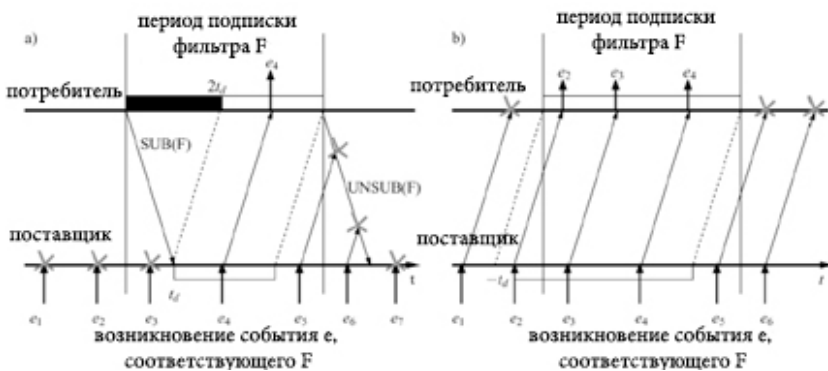


Рис. 2. Период отключения после подписки на простую маршрутизацию а) и наводнение с помощью фильтрации на стороне клиента б)

Предварительное, но неполное решение для логической мобильности. Хотя фильтры, зависящие от местоположения, напрямую не поддерживаются текущим промежуточным программным обеспечением Публикации/Подписки, можно утверждать, что в этом случае не очень сложно имитировать их поверх доступных в настоящее время систем. Идея заключалась бы в создании обертки вокруг существующей системы, которая следует за изменениями местоположения пользователей, прозрачно отписывается от старого местоположения и подписывается на новое при перемещении пользователя. Однако, в зависимости от стратегии внутренней маршрутизации системы событий, это может привести к неожиданным результатам. Стратегии маршрутизации, развернутые во многих существующих системах событий на основе содержимого, таких как Siena [3], Elvin [4], and Rebeca [5], приводят к запретным периодам, когда никаких уведомлений не поступает. Проблема в том, что для обработки новой подписки обычно требуется некоторая временная задержка. После подписки на фильтр требуется некоторое время t_d , пока подписка не будет распространена на потенциальный источник. Затем требуется как минимум другое время t_d , пока уведомление не достигнет подписчика. Это явление изображено на рисунке 2а. (Обратите внимание, что задержка t_d может отличаться для разных источников уведомлений и может меняться со временем.) Если клиент остается в любом новом месте менее $2t_d$ времени, тогда подписчик «голодает», то есть он будет получать небольшие уведомления или вообще не получать их.

Интуитивное, но неэффективное решение. Другое базовое решение, которое может быть непосредственно построено с использованием существующей технологии, основано на наводнении. Местный брокер может принять решение о предоставлении уведомления клиенту в зависимости от его текущего местоположения (рисунок 2б). Очевидно, что наводнение предотвращает запретный период, который присутствовал в предыдущем решении, но при этом, наводнение является очень дорогостоящей стратегией маршрутизации, особенно для крупных систем Публикации/Подписки [6].

Качество обслуживания логической мобильности. Интересно, что в то время как наводнение является очень дорогостоящим и, следовательно, нежелательным, оно очень близко к тому качеству обслуживания, которое мы хотели бы достичь для логической мобильности, а именно к понятию подписки на «все, везде и постоянно». Проблема в том, что трудно точно определить поведение наводнений, не обращаясь к некоторым нежелательным теоретическим построениям операционной семантики.

Однако с логической мобильностью нет опасности получать уведомление дважды, потому что потребитель остается прикрепленным к одному и тому же «пути доставки». Поэтому качество обслуживания, которое мы требуем для логической мобильности, просто указано следующим обра-

зом: при изменении местоположения от x до y все уведомления должны доставляться потребителю «как если бы» наводнение использовалось в качестве базовой стратегии маршрутизации. Это утверждение более конкретно отражено на рисунке 3, где последовательность уведомлений, генерируемых любым потребителем, делится на эпохи, которые соответствуют тому, что уведомление действительно приходит к потребителю (границы эпохи между местоположением y и z рисуются как виртуальное уведомление $ny \rightarrow z$). Мы требуем, чтобы все уведомления, соответствующие текущей локально-зависимой подписке из каждой такой эпохи, должны быть доставлены. Интуитивно, эпохи определяют семантику наводнения.

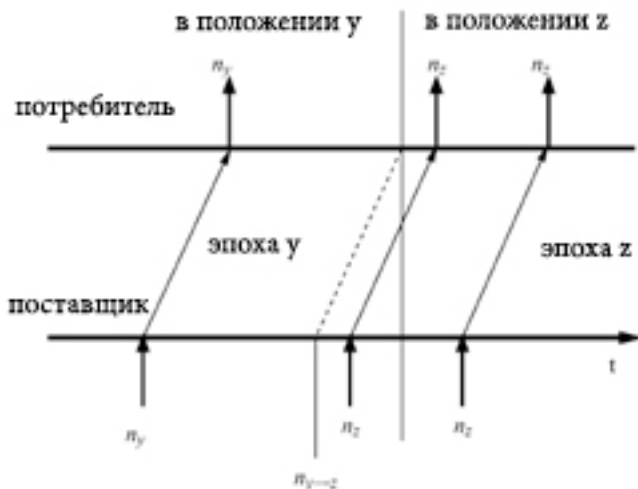


Рис. 3. Определение качества обслуживания для логической мобильности с использованием виртуальных уведомлений $ny \rightarrow z$, которые поступают к потребителю только во время изменения местоположения от y до z .

Заключение. В данной статье представлен подход к поддержке мобильности в существующем промежуточном программном обеспечении протокола Публикации/Подписки, используемого при формировании виртуальных организаций. Мы проанализировали проблему мобильности с точки зрения, основанной на событиях парадигмы, и определили два отдельных аспекта мобильности. В то время как физическая мобильность связана с понятием перефильтрации клиента для разных брокеров и может быть реализована прозрачно, логическая мобильность относится к определенной форме осведомленности о местоположении, предлагающей клиенту контроль доставки уведомлений в виде фильтров, зависящих от местоположения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. D. Johnson. Scalable support for transparent mobile host internetworking. *Wireless Networks*, 1: 311–321, Oct. 1995.
2. Zeidler and L. Fiege. Mobility support with REBECA. In *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshop on Mobile Computing Middleware*, 2003.
3. W. Segall and D. Arnold. Elvin has left the building: A publish/subscribe notification service with quenching. In *Proceedings of the 1997 Australian UNIX Users Group*, Brisbane, Australia, Sept. 1997.
4. A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3): 332–383, 2001.
5. L. Fiege and G. Mühl. Rebeca Event-Based Electronic Commerce Architecture, 2000. <http://www.gkec.informatik.tu-darmstadt.de/rebeca>.
6. G. Mühl, L. Fiege, F. C. Gärtner, and A. P. Buchmann. Evaluating advanced routing algorithms for content-based publish/subscribe systems. In A. Boukerche and S. Majumdar, editors, *The Tenth IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2002)*, Fort Worth, TX, USA, October 2002. IEEE Press.
7. V.A. Antipov, O.V. Antipov, A.N. Pytkin. Dynamic Publish/Subscribe Systems. *International conference on computer technologies in physical and engineering applications*. – SPb.: 2014, p. 11 – 11a

СОДЕРЖАНИЕ

Пылькин А.Н., Соколова Ю.С.

СОВРЕМЕННЫЙ PASCALABC.NET,
КАК УДАЧНАЯ ОТЕЧЕСТВЕННАЯ РАЗРАБОТКА
ДЛЯ ИЗУЧЕНИЯ БАЗОВЫХ ПРИНЦИПОВ ПРОГРАММИРОВАНИЯ 4

Маркова В.В.

РАЗБИЕНИЕ ГРАФА РАЗВИТИЯ МОДЕЛИРУЮЩЕЙ
АВТОМАТИЗИРОВАННОЙ ДЕЛОВОЙ ИГРЫ НА ЭТАПЫ ПО СМЫСЛУ 11

Головкин Н.В., Цуканова Н.И.

ГЕНЕРАЦИЯ ИЗОБРАЖЕНИЙ КОШАЧЬИХ МОРДОЧЕК
С ПОМОЩЬЮ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНОЙ НЕЙРОННОЙ СЕТИ DCGAN 13

Ромашкова В.В.

ЭТАПЫ РЕАЛИЗАЦИИ АЛГОРИТМА ПРИНЯТИЯ РЕШЕНИЯ
НА ОСНОВЕ ФУНДАМЕНТАЛЬНОГО АНАЛИЗА
ПРИ ИНВЕСТИРОВАНИИ НА ФОНДОВОМ РЫНКЕ 15

Шурьгина О.В., Цуканова Н.И.

ПРЕОБРАЗОВАНИЕ ИЗОБРАЖЕНИЯ В ИЗОБРАЖЕНИЕ
С ПОМОЩЬЮ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫХ НЕЙРОННЫХ СЕТЕЙ 18

Перевезенцев Е.Е.

МЕТОД ПРОГНОЗИРОВАНИЯ ЦЕН НА ФОНДОВОМ РЫНКЕ 20

Проказникова Е.Н., Калугин Д.И. ПРИМЕНЕНИЕ РАЗЛИЧНЫХ ПРИКЛАДНЫХ МЕТОДОВ ДЛЯ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ РЕЗУЛЬТАТОВ БЕЙСБОЛЬНЫХ МАТЧЕЙ.....	22
Дмитриева Т.А., Чевагин В.В., Благодаров Е.А. ОБЗОР АНАЛОГОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОРГАНИЗАТОРА ПРЕПОДАВАТЕЛЯ	25
Журавлев П.В. УЧЁНЫЙ, ИМЯ КОТОРОГО НОСИТ НАШ УНИВЕРСИТЕТ!	31
Козанков И.Ю. МЕТРИКИ ДЛЯ ТЕСТИРОВАНИЯ ТОЧНОСТИ АЛГОРИТМОВ НЕЭТАЛОННОЙ ОЦЕНКИ КАЧЕСТВА ИЗОБРАЖЕНИЙ	34
Дмитриева Т.А., Мерзлякова М.Э. ИНТЕРФЕЙС ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ СЕБЕСТОИМОСТЬЮ ПРОДУКЦИИ НА ПРЕДПРИЯТИИ	35
Благодарова Т.А., Благодаров Е.А. АЛГОРИТМ ВЫПОЛНЕНИЯ СЦЕНАРИЯ ГРАДУИРОВКИ ПРЕОБРАЗОВАТЕЛЕЙ ДАВЛЕНИЯ.....	39
Кирилова Л.М. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПОИСКА НЕДОСТОВЕРНОЙ ИНФОРМАЦИИ В ИНТЕРНЕТЕ	43
Латахин А. ВОССТАНОВЛЕНИЕ НЕСФОКУСИРОВАННЫХ ИЗОБРАЖЕНИЙ МЕТОДОМ ДВУМЕРНОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ И РЕГУЛЯРИЗАЦИИ ТИХОНОВА.....	45
Швечкова О.Г., Елисеев В.А. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПОДБОРА КОМПЛЕКТУЮЩИХ ДЛЯ КОМПЬЮТЕРА НА ОСНОВЕ МНОГОКРИТЕРИАЛЬНОЙ ВЫБОРКИ.....	49
Иванова Е.С. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ АСИНХРОННЫМ ДВИГАТЕЛЕМ.....	53
Бабкин Д.И. АЛГОРИТМЫ И МЕТОДЫ ОБРАБОТКИ СИГНАЛОВ С СЕРДЦА	55
Агафонов П.А., Асташкин М.С. КОМПЬЮТЕРНАЯ РЕАЛИЗАЦИЯ НАСТОЛЬНОЙ ИГРЫ «СОПРОТИВЛЕНИЕ» НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ KOTLIN.....	58
Малистова М.Р. ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРИМЕНЯЮЩЕГО АНАЛИЗ ТЕКСТОВЫХ ДАННЫХ	62
Лукьянова Т.А. ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРИМЕНЯЮЩЕГО ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ МУЛЬТИМЕДИЙНЫХ ДАННЫХ	67
Кузовлев Р.Г. СРАВНЕНИЕ ПРОБРАЗОВ ВХОДНЫХ СИГНАЛОВ ДЛЯ РАСПОЗНАВАНИЯ ФИГУР ТЕХНИЧЕСКОГО АНАЛИЗА С ПРИМЕНЕНИЕМ НЕЙРОННЫХ СЕТЕЙ.....	71
Антипов В.А., Антипов О.В., Солдатов В.В. МОБИЛЬНЫЕ АБОНЕНТЫ ВИРТУАЛЬНЫХ ОРГАНИЗАЦИЙ	75

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Межвузовский сборник научных трудов

Подписано в печать хх.06.2021. Формат 60x84 1/16.

Бумага офсетная. Печать цифровая.

Гарнитура «Times New Roman».

Усл. печ. л. 5,25.

Тираж _____ экз. Заказ № 4302

Издательство ИП Коняхин А.В. (Book Jet)

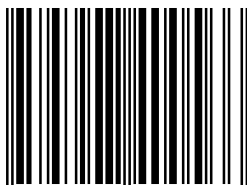
Отпечатано в типографии Book Jet
390005, г. Рязань, ул. Пушкина, д. 18

Сайт: <http://bookjet.ru>

Почта: info@bookjet.ru

Тел.: +7(4912) 466-151

ISBN 978-5-907400-56-6



9 785907 400566 >