Выбор алгоритма прогнозирования для разработки аналитического программного обеспечения

В.М. Курейчик, С.Б. Картиев
Южный федеральный университет, Таганрог

Аннотация: В данной работе производится выбор алгоритма обработки данных, необходимый для разработки аналитического ПО в комплексном проекте «Разработка и создание высокотехнологичного производства инновационной системы комплексного учета, регистрации и анализа потребления энергоресурсов и воды промышленными предприятиями и объектами ЖКХ». Произведен обзор существующих алгоритмов и методов прогнозирования в системах с большим числом параметров и большой эпохой анализа. Конкретным приложением для искомого алгоритма является прогнозирование потребления энергоресурсов и воды. На основании обзора алгоритмов выбраны алгоритмы, наиболее подходящие для этой задачи. Рассматривается тандемное использование построения дерева решений и генетического алгоритма прогнозирования. Сформулированы дальнейшие задачи, которые необходимо решить для эффективного внедрения данных алгоритмов при разработке аналитического программного обеспечения.
Ключевые слова: Аналитическое программное обеспечение, прогнозирование, генетический алгоритм.

Аналитическое программное обеспечение (АПО) является необходимой частью общего программного комплекса системы комплексного учета, регистрации и анализа потребления энергоресурсов и воды, но в тоже время может функционировать, как отдельный программный модуль. Основными функциями АПО является контроль достоверности данных, получаемых с абонентских приборов учета энергоресурсов, проведение краткосрочного и долгосрочного прогнозирования потребления ресурсов, анализ корректности информации, получаемой от абонентских приборов учета, на основе прогнозных значений и результатов анализа параметров окружающей среды [1].

Съем показаний приборов учета может производиться либо периодически, через определенные промежутки времени, либо постоянно. Для обеспечения бесперебойной работы систем учета энергоресурсов и предотвращения нештатных ситуаций используются методы прогнозирования. Для данной задачи целесообразным является применение методов анализа временных рядов. Для решения подобной задачи временной

ряд разбивают на части, соответствующие равным временным промежуткам. В основном, задача прогнозирования временных рядов сводится к сравнению одной последовательности с другой [2].

Предлагается гибридный подход, основанный на принципах классификации, регрессионного анализа и эволюционного моделирования. Данный подход позволяет получать более качественные решения в задачах, которые имеют количество параметров более одного исследуемого процесса. Для данного подхода необходимо для каждой конкретной задачи осуществить поиск оптимальных параметров модели, в каждом случае существуют индивидуальные особенности реализации генетических алгоритмов (ГА). ГА позволяют существенно сократить время подбора неопределяемых автоматически параметров для регрессионно - классификационных деревьев [3,4].

Постановка задачи

$x_0, x_1 \dots\dots\dots x_n$; - временной ряд значений параметров исследуемого процесса, $x_i \in R$; $x_{t+d}(W) = f_t(x_1, \dots\dots, x_t; w)$ – модель временного ряда, где d= 1,…..,D, D-горизонт прогнозирования. W – вектор параметров модели, x – единица статистического материала. Традиционно данная задача решается при помощи метода наименьших квадратов (МНК). МНК - метод нахождения оптимальных параметров линейной регрессии таких, что сумма квадратов ошибок минимальна [5]. Метод заключается в минимизации Евклидова расстояния между двумя векторами – вектором восстановленных значений зависимой переменной и вектором фактических значений зависимой переменной.

$$Q_t(W) = \sum_{i=t_0}^{t}(x_i(W) - x_i)^2 \rightarrow \min(w). \qquad (1)$$

Применение МНК имеет ограничения для решения данной задачи, т.к. имеются следующие недостатки [6]: большие затраты процессорного

времени (долгое время выполнения программы, реализующей прогнозирование); количество рядов может быть очень большим; поведение рядов не может регулироваться; функция потерь может быть неквадратичной.

Основные методы прогнозирования

Рассмотрим регрессионные модели прогнозирования [7]: простая линейная регрессия, множественная регрессия, нелинейная регрессия.

К авторегрессионным моделям относят: методологию Бокса-Дженкинса(ARIMA), модель условной гетероскедатичности.

К моделям экспоненциального сглаживания относят: экспоненциальное сглаживание, модель Хольта или двойное экспоненциальное сглаживание, модель Хольта-Винтерса или тройное экспоненциальное сглаживание, методы, основанные на нейросетевых моделях, моделях марковских цепей, классификационно-регрессионных деревьях.

Рассмотрим методы, основанные на нейронных сетях. Роль нейронных сетей в решении задачи прогнозирования временных рядов состоит в предсказании будущей функциональности системы по ее предыдущему состоянию:

$$(x_{t-1}, x_{t-2}, \dots \dots x_{t-n}). \qquad (15)$$

Имеется множество элементов информации о значениях состояния системы, которые предшествуют моменту прогнозирования. На основании данной информации сеть вырабатывает решение - каким будет наиболее вероятным текущее значение x_t.

Для подобной задачи оптимальной архитектурой является многослойный персептрон с обратным распространением ошибки. Для

обучения подобной сети обычно используется алгоритм обратного распространения ошибки [8].

Модели классификационно-регрессионных деревьев

Классификационно-регрессионные деревья используются в областях статистики и анализа данных для моделей прогнозирования временных рядов. Основной алгоритм для построения данных типов деревьев является алгоритм CART (Classification and regression trees) [5].

Основные особенности алгоритма CART: функция оценки качества разбиения; механизм отсечения дерева; алгоритм обработки пропущенных значений; построение деревьев рекурсии. Алгоритм строит последовательность деревьев. Модель для решения задачи прогнозирования состоит из K последовательностей деревьев решений заданной длины M, где

K – число категорий, к которым может относиться выходная переменная. Шаг 1. Для всех m =1,…..,M. Шаг 2. Для всех k=1,…,K. Шаг 3. Вычислить антиградиент функции потерь. Шаг 4.Обучить дерево решений. Шаг 5. Обновить модель, добавив деревья в соответствующие последовательности.

Приведем постановку задачи классификации: имеется множество классов F = $\{f_1, f_2 \ldots. f_n\}$, где n – является мощностью данного множества, что

характеризует число классов, допустимых для данной задачи. Пространство признаков класса является N-мерным векторным пространством. Для некоторого вектора $\vec{x} = \{x_1, \ldots, x_n\}^T$ необходимо провести процедуру

принятия решения о принадлежности данного вектора к некоторому классу из заданного множества F. Для решения данной задачи мы будем использовать алгоритм RF. Рассмотрим основные методы, которые используются в модифицированном алгоритме RF. Алгоритм RF был представлен Лео Брейманом и Адель Катлер [9]. Он основан на построении ансамбля классификационно-регрессионных деревьев, каждое из которых

строится по выборке, получаемой из исходной обучающей выборки с использованием бэггинга. Бэггинг – метод формирования ансамблей классификаторов. Данный метод позволяет решить как проблему классификации, так и проблему регрессии.

Основной идеей RF является объединение результатов работы базовых классификаторов и выделение неточностей при их работе. Описание алгоритма будет производиться с использованием методики Дональда Кнута для описания функционала в виде кода и документирования [10]:

Функция Ensemble RandomForest(){

For b = 1 to B {

Вывести образец *Z* размера *N* из обучающей выборки.

Увеличиваем дерево T_b на данные, полученные при помощи бэггинга, далее рекурсивно повторяя следующие шаги для каждой вершины дерева, пока не будет достигнут n_{min}.

Выбрать m переменных при помощи uniform_distribution из p переменных.

Выбрать лучшую переменную среди m.

Разделить узел на два дочерних узла.

}

Return Ансамбль деревьев $\{T_b\}_1^B$

}

где B – количество деревьев. Алгоритм позволяет построить ансамбль деревьев решений для выработки наиболее качественного решения.

Для выполнения прогнозирования для следующей точки X: оценка регрессии высчитывается по следующей формуле: $f_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$. Пусть $C_b(x)$ будет классом прогнозирования b-го

дерева из ансамбля деревьев. Тогда оценка качества классификации будет рассчитываться по формуле $C_{rf}^{B}(x) = \{C_b(x)\}_1^B$.

Основным недостатком RF является большой размер структуры данных. Это приводит к большому расходу памяти. Временная сложность алгоритма – *O(n * K),* где *K* – количество деревьев. Достоинствами применения RF является способность к параллелизации вычислений. Проектируя класс на языке Java, получим следующую структуру:

```
Class ModRandomForest {
    BoolTrain(); //Обучение дерева
    FloatPredict(); //Прогнозировать
    FloatFuzzyPredict(); //Нечеткий прогноз
}
```

где, BoolTrain() является функцией обучения самого RF, FloatPredict(inputData) позволяет на основе входных данных прогнозировать значения. Также присутствует функция нечеткого прогноза, называемая FloatFuzzyPredict().

Проблемы классификации могут быть рассмотрены как задачи оптимизации для нахождения лучшей целевой функции. Для решения данной задачи будем использовать алгоритм случайного леса.

Алгоритм обучения случайного леса:

1) Имеется определенная обучающая выборка, состоящая из элементов, вес которых в корневой вершине равен 1.

2) В любом корне N которое может быть расширено, вычислить количество образцов каждого класса. Классы распределены по частям или ответвлениям.

3) Выбрать подходящие образцы из обучающей выборки.

4) На каждой вершине вариантов поиска:

a. Выбрать некоторый критерий, который является атрибутом для разделения узла.

b. Провести функцию обучения

c. Выбрать искомое значение атрибута в котором есть подходящий критерий по информационному наполнению.

5) Разделяем N на поддеревья в соответствии с возможными выводами атрибута, выделенного на предыдущем шаге.

6) Повторять 2-5 шаги до тех пор, пока поиск в дереве не будет произведен.

Задача прогнозирования представлена так: дан временной ряд значений параметров исследуемого процесса, например работы программной системы.

$$x_0, x_1 \dots\dots\dots x_n; \quad (13)$$

где $x_i \in R$.

$$x_{t+d}(W) = f_t(x_1, \dots\dots, x_t; w). \quad (14)$$

Формула (2) является моделью временного ряда, где $d = 1, \dots\dots, D$, D - горизонт прогнозирования. W – вектор параметров модели, x – единица статистического материала. Иными словами, АПО получает в реальном времени некоторую последовательность числовых значений исходов. Разработанный алгоритм обучения RF решает задачу классификации и регрессии с минимальным процентом ошибок. Это может позволить применять данный алгоритм в задаче прогнозирования.

Комбинированный алгоритм прогнозирования

Разработанный алгоритм состоит из двух этапов:

1) Построение дерева решения для нахождения тех значений, которые необходимы для дальнейшего анализа. (в основном выбираются те значения, которые на $X_{тек} - 1$ шаге вызывают функции изменения).

2) Генетический алгоритм, который на основании 1 этапа реализует функцию прогнозирования.

Шаги построения дерева. Для начала, опишем принципы организации классификационно-регрессионных деревьев для первичного отбора параметров, в дальнейшем, участвующих как популяция во втором глобальном шаге алгоритма – генетического алгоритма.

Каждый узел дерева решений (DT) в CART имеет двух потомков. На каждом этапе построения DT правило, формируемое в узле, делит обучающую выборку на две части. Первая часть – соответствует выполнению правила, а вторая часть является той, в которой правило не выполняется. Для выбора оптимального правила используется функция оценки качества разбиения.

Шаги генетического алгоритма

Вход: Количество решений (n), Нижние и верхние границы решения, Максимальное число итераций.

Выход: Лучшее решение ($X_{лучш.}$)

1) Для начала, определим Целевую Функцию (далее - ЦФ) $\mathbf{f(x)}$. У нас она будет равна $f(x) = \mathbf{X_{тек.}} - \mathbf{X_{прогн.}}$

2) Сгенерируем популяцию, используя структуру данных – классификационно-регрессионное дерево. Использование дополнительной структуры данных является осмысленным ввиду необходимости отбирать лучшие, с точки зрения целевой функции, значения. В нашем случае лучшие решения – значения, имеющие наибольшее значение в метаданных. Случайно сгенерируем первую популяцию, состоящую из *n* хромосом, где для каждой четвертой новой популяции – выставим $X_{лучш.} = X_0$.

3) Оценим ЦФ каждой хромосомы с текущей популяцией.

4) Выполнение генетических операторов.

4.1) Выберем две родительской хромосомы из текущей популяции, основываясь на их минимальных значениях ЦФ.

4.2) Выполнить процедуру кроссинговера между двумя родителями для формирования двух детей.

4.3) Случайно выбрать хромосому текущей популяции и выполнить процедуру мутации в случайно выбранном месте в популяции (локус).

4.4) Найти лучшие хромосомы этой популяции в соответствии с их ЦФ.

4.5) Если $X_{\text{тек.}} > x_{\text{лучш.}}$, **задать** $X_{\text{лучш.}} = x_{\text{тек.}}$.

5) Если условия завершения выполняется, то остановить работу алгоритма и возвратить $X_{\text{лучш.}}$.

Параллельный алгоритм прогнозирования

Используя принцип параллелизма можно получить максимальное ускорение эквивалентное значению, полученному по формуле, полученной из закона Амдала [3].

$$S = \frac{1}{(1-W)+\frac{W}{N}} \quad (15)$$

где W – доля работ, которые подлежат распараллеливанию, N – количество параллельно работающих процессоров (ядер).

Для построения модели параллельного алгоритма необходимы следующие входные данные:

1) Множество переменных, используемых в алгоритме.

2) Множество выполняемых операций.

3) Служебная информация и метки (идентификатор потока и.т.д.), которые предоставляют информацию о синхронизации параллельных вычислений.

Внутренняя структура параллельной программы представляет собой ориентированный граф, где параллельно выполняемые потоки выделяют как отдельные подграфы. В отдельных потоках содержится совокупность множества переменных и выполняемых операций.

Дан ориентированный ациклический граф $G = (V,E)$ где V - множество вершин, а E - множество ребер графа. Каждая вершина графа представляет собой совокупность данных и операции вычислительного процесса, представлен на рис.1.

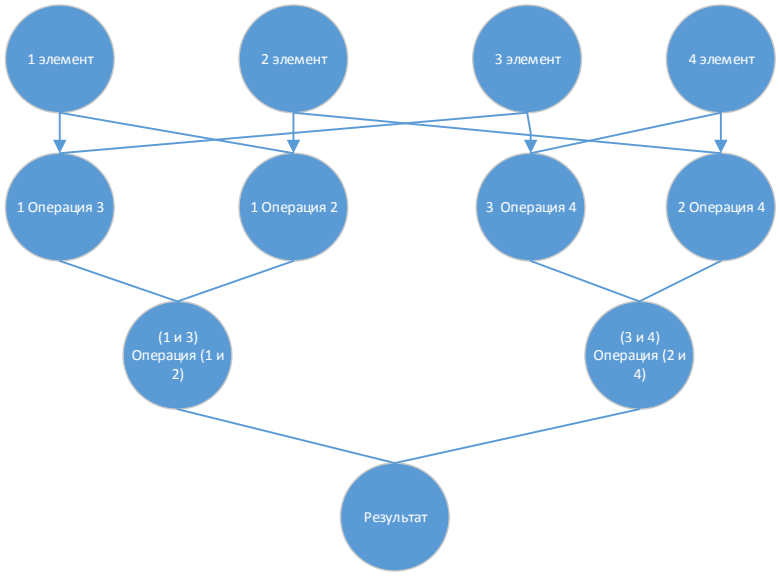

Рис. 1 – Примерная схема вычислительного процесса в параллельной программе

При разработке алгоритмов прогнозирования для применения в АПО является способность алгоритма к распараллеливанию. Данная способность потенциально связана с одним из двух внутренних свойств, таких как параллелизм задач и параллелизм данных.

Параллелизм применяется для алгоритма обучения, который решает данную задачу с использованием классификационно-регрессионных моделей и нейронных сетей. Идея алгоритма состоит в том, что для прогнозирования, на каждом шаге вырабатывается обучающая выборка, которая представляет собой предыдущие состояния системы, которые влияют на N + 1 элемент

временного ряда. На первом шаге, все предыдущие состояния расположены в регрессионно-классификационном дереве, откуда производится выборка данных о тех состояниях, которые влияют на текущее. После этого нейронная сеть обучается на основе обучающей выборки.

Допустим, мы имеем историю состояний системы $S = h_0, h_1, h_2 \dots \dots h_{n-1}$, где h – состояние системы на некоторый временной промежуток (элемент временного ряда), а n – временной промежуток прогнозируемого состояния.

Параллельный алгоритм прогнозирования будет выглядеть так:

Вход: Количество решений (n), Нижние и верхние границы решения, Максимальное число итераций.

Выход: Лучшее решение ($X_{\text{лучш.}}$).

Для начала, определим Целевую Функцию (далее - ЦФ) $f(x)$. У нас она будет равна $\sum_{K=1}^{L}(g(p_k) - f_{k+n})^2$, где p – прошедшие события, f – текущие события и L – длина секции. Сама ЦФ будет высчитываться параллельно.

1. Параллельно сгенерируем популяцию для всех особей, используя классификационно-регрессионное дерево.
2. Случайно сгенерируем первую популяцию, состоящую из n хромосом, где для каждой четвертой новой популяции – выставим $X_{\text{лучш.}} - X_0$. [5].
3. Вычислить ЦФ для каждой хромосомы.
4. Отсортировать особей согласно их целевой функции.
5. Оценим ЦФ каждой хромосомы с текущей популяцией (параллельно).

6. Выполнение генетических операторов (параллельно – ***parallel for***).

7. Выберем две родительской хромосомы из текущей популяции, основываясь на их минимальных значениях ЦФ.

8. Выполнить процедуру кроссинговера между двумя родителями для формирования двух детей.

9. Случайно выбираем хромосому текущей популяции и выполняем процедуру мутации в случайно выбранном месте в популяции (локус).

10. Найдем лучшие хромосомы этой популяции в соответствии с их ЦФ.

11. Если $X_{\text{тек.}} > x_{\text{лучш.}}$, задать $X_{\text{лучш.}} = x_{\text{тек.}}$.

12. Если условия завершения выполняется, то остановить работу алгоритма и возвратить $X_{\text{лучш.}}$.

Основным отличием данного параллельного алгоритма от последовательного является организация вычислительного процесса для более эффективного подбора подходящих решений для применения в системе учета и анализа потребления энергоресурсов. Структура АПО представлена на рис. 2. Выбранная связка алгоритмов представляется наиболее применимой в структуре АПО.

Для корректной работы генетического алгоритма необходимо на первом этапе построить деревья, используя CART-алгоритм, причем требуется сделать этот шаг до этапа работы подсистемы прогнозирования потребления энергоресурсов. Данный этап является подготовительным и может быть проведен либо подсистемой группировки данных, либо подсистемой статистической обработки. Так как CART-алгоритм имеет большую вычислительную сложность, эффективнее всего применять его в подсистеме статистической обработки, так как в отличие от подсистемы группировки

данных она не задействуется для выполнения сторонних фоновых задач, таких как поддержка обмена между различными подсистемами и консолями аналитического программного обеспечения и MS SQL Server.

Второй этап работы связан с функционированием генетического алгоритма, он может работать непосредственно в подсистеме прогнозирования потребления.

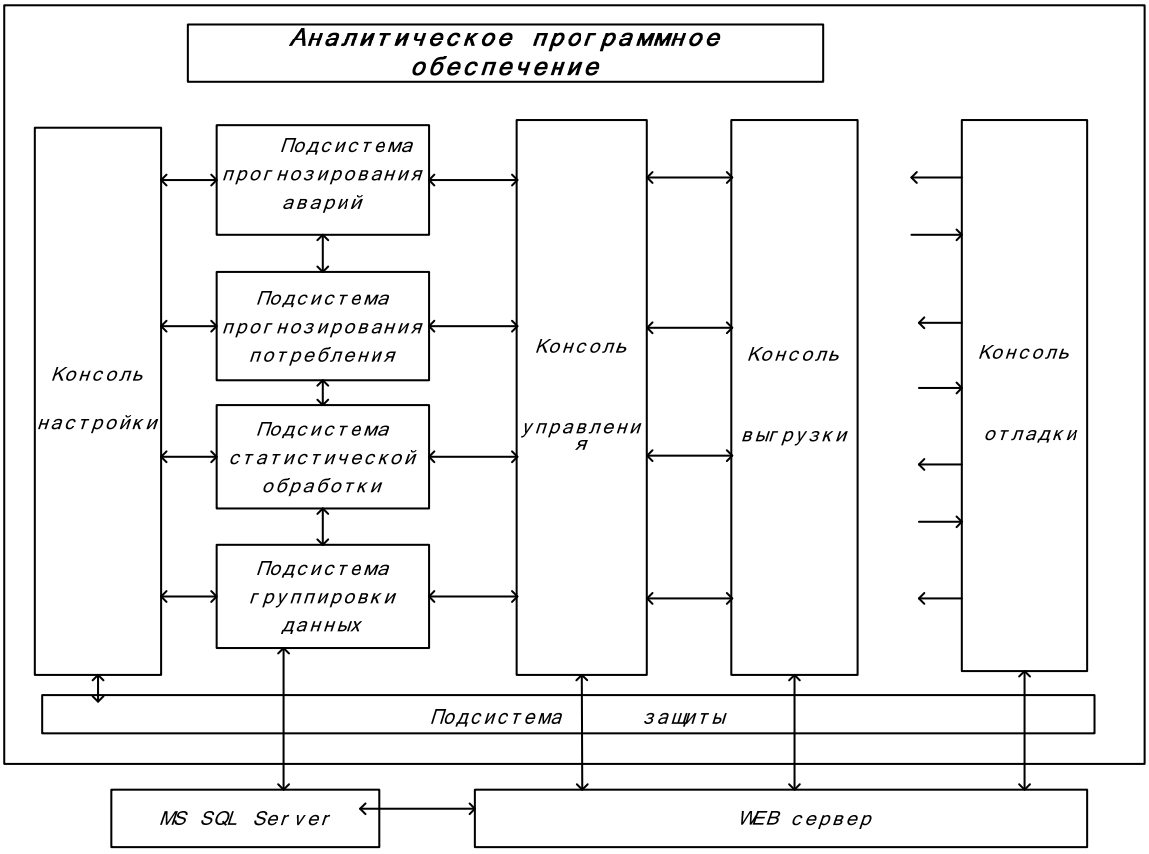

Рис.2 – Структура аналитического программного обеспечения

Заключение

В работы был произведен выбор связки алгоритмов для прогнозирования потребления энергоресурсов и воды. Для построения классификационно-регрессионных деревьев выбран алгоритм CART, результаты его работы затем используются в качестве входных данных для генетического алгоритма. Данные алгоритмы отвечают требованиям по

глубине эпохи анализа (до 3-х лет), не обладают повышенными требованиями к объему предоставляемых данных. Однако на данном этапе работы не была проведена оценка вычислительной сложности этих алгоритмов. Также необходимо отметить, что выбор трендов потребления энергоресурсов и воды будет влиять на вычислительную сложность выбранных алгоритмов и, вероятно, потребуется использовать секционное прогнозирование потребления (с разбиением на поселки, районы городов, и т.п.).

Работа выполнена при финансовой поддержке Минобрнауки РФ в рамках реализации проекта «Разработка и создание высокотехнологичного производства инновационной системы комплексного учета, регистрации и анализа потребления энергоресурсов и воды промышленными предприятиями и объектами ЖКХ» по постановлению правительства №218 от 09.04.2010 г. Работа выполнялась во ФГАОУ ВО ЮФУ.

Литература

1. Е.С.Семенистая, И.Г.Анацкий, Ю.А. Бойко Разработка программного обеспечения автоматизированной системы контроля и учета энергоресурсов и воды // Инженерный вестник Дона, 2016. №4. URL: ivdon.ru/ru/magazine/archive/n4y2016/3897.

2. Картиев С.Б. Параллельный алгоритм прогнозирования коротких временных рядов / С.Б. Картиев, В.М. Курейчик, А.В. Мартынов //Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT’15». Научное издание в 4-х томах.– Физматлит М., 2015. с.27-47

3. Чистяков С.П. Случайные леса: обзор/С.П. Чистяков //Труды Карельского научного центра РАН, – 2013, – выпуск 1. – с. 117– 136.

4. J.T. Tou, R.C. Gonsalez. Pattern Recognition Principes. Addison-Wesley, 1977. – 377 p.

5. Hastie, T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. – 2nd ed. – Springer-Verlag, 2009. 746 p.

6. Курейчик В.М. Особенности построения систем поддержки принятия решений / В.М. Курейчик // Известия ЮФУ. Технические науки – 2012 № 7. – С. 92-98.

7. Machine Learning, Neural and Statistical Classification / Editors: D. Michie, D. J. Spiegelhalter, C. C. Taylor. – London: Ellis Horwood, 1994. 298 p.

8. Гладков Л.А. Генетические алгоритмы / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик / Под ред. В.М. Курейчик. М.: Физматлит, 2006. 320 с.

9. Е.В.Пучков Сравнительный анализ алгоритмов обучения искусственной нейронной сети // Инженерный вестник Дона, 2013. №4. URL: ivdon.ru/ru/magazine/archive/n4y2013/21357.

10. Knuth D. Art of Computer Programming, Volume 1: Fundamental Algorithms. 3rd edition. Addison-Vesley, 1997. 736 p.

References

1. E.S.Semenistaya, I.G. Anatsky, Yu.A. Boyko Inzhenernyj vestnik Dona (Rus). 2016. №4. URL: ivdon.ru/ru/magazine/archive/n4y2016/3897.

2. S.B. Kartiyev, V.M. Kureychik, A.V. Martynov. Proceedings of the Congress on Intelligent Systems and Information Technologies "IS & IT'15". Scientific publication in 4 volumes. Fizmatlit M., 2015. p. 27-47

3. Chistyakov S.P. Proceedings of the Karelian Research Center of the Russian Academy of Sciences, 2013, Issue 1. pp. 117-136.

4. J.T. Tou, R.C. Gonsalez. Pattern Recognition Principes. Addison-Wesley, 1977. 377 p.

5. Hastie, T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. Springer-Verlag, 2009. 746 p.

6. Kureychik V.M. Izvestiya YUFU. Tekhnicheskiye nauki. 2012. No. 7. pp. 92-98.

7. Machine Learning, Neural and Statistical Classification. Editors: D. Michie, D. J. Spiegelhalter, C. C. Taylor. London: Ellis Horwood, 1994. 298 p.

8. L.A. Gladkov, V.V. Kureychik, V.M. Kureychik Geneticheskiye algoritmy [Genetic algorithms]. M.: Fizmatlit, 2006. 320 p.

9. Ye.V.Puchkov Inzhenernyj vestnik Dona (Rus). 2013. №4. URL: ivdon.ru/ru/magazine/archive/n4y2013/21357.

10. Knuth D. Art of Computer Programming, Volume 1: Fundamental Algorithms. 3rd edition. Addison-Vesley, 1997. 736 p.