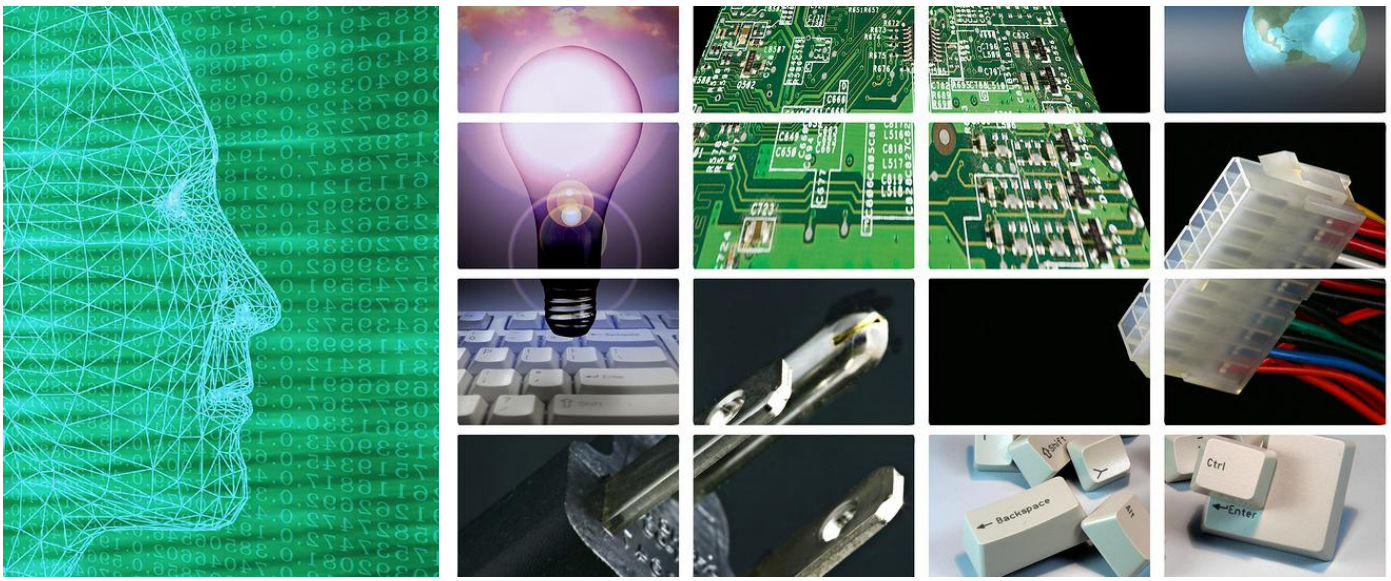
ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА
TECHNICAL UNIVERSITY OF VARNA

Година XVIII, Брой 1/2020

КОМПЮТЪРНИ НАУКИ И ТЕХНОЛОГИИ



IEEE Bulgaria Communications Chapter

FCA Faculty of Computing & Automation

COMPUTER SCIENCE AND TECHNOLOGIES

Year XVIII, Number 1/2020

Компютърни науки и технологии

Издание
на Факултета по изчислителна техника и автоматизация
Технически университет - Варна

Редактор: доц. д-р Ю. Петкова
Гл. редактор: доц. д-р Н. Николов

Печат: ТУ-Варна

За контакти:

Технически университет - Варна
ФИТА
ул. „Студентска” 1, 9010 Варна,
България
тел./факс: (052) 383 320
e-mail: ned.nikolov@tu-varna.bg
yulka.petkova@tu-varna.bg

ISSN 1312-3335

Computer Science and Technologies

Publication
of Computing and Automation Faculty
Technical University of Varna

Editor: Assoc. Prof. Y. Petkova, PhD
Chief Editor: Assoc. Prof. N. Nikolov, PhD

Printing: TU-Varna

For contacts:

Technical University of Varna
Faculty of Computing and Automation
1, Studentska Str., 9010 Varna,
Bulgaria
Tel/Fax: (+359) 52 383 320
e-mail: ned.nikolov@tu-varna.bg
yulka.petkova@tu-varna.bg

ISSN 1312-3335

СЪДЪРЖАНИЕ **CONTENTS**

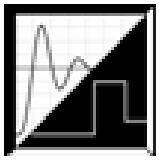
TU-Varna FCA Faculty of Computing & Automation


TU-Varna
FCA
Faculty of Computing & Automation


КТТ


TU-Varna
CSE
Computer Science & Engineering Department


СОФТУЕРНИ И ИНТЕРНЕТ
ТЕХНОЛОГИИ



A SYSTEM FOR CLUSTERING OF TIME SERIES

Ventsislav Nikolov, Aleksandar Krastev

Abstract: The clustering is an important subtask of many other actions like parallel processing, local data analysis, dimensionality reduction, etc. In this paper some aspects are considered about clustering of time series. The main accent is directed toward modification of the known clustering algorithms taking into account of the weights in both the time series values and the time series as a whole in the process of clustering as well as determining the optimal number of clusters and assessment of clustering quality statistics.
Keywords: Clustering, Time series, Clustering quality, Number of clusters, Self-organizing map

1. Introduction

Clustering of objects is an important task in many fields and it is a base for modeling and many processing techniques. It can be used for: classification, patterns recognition, dimensionality reduction, vectors quantization, data compression [2][4][5], etc. The main purpose of the clustering is creating of groups of patterns in such a way that the patterns with similar feature values to be in a same group. This could provide many advantages, some of which are the following:

- Parallel data processing. The data in a given cluster can be processed independently from the other clusters [3][7].
- Possibility for analysis only on specific parts of the whole available data [6][11].
- Better interpretability. Additional statistical analysis can be done about the clusters, like: dispersion, average linkage, etc. [21][23].
- Additional flexibility of data processing by further clustering of some clusters or merging of clusters. Thus hierarchical models can be developed [12].

Here the objects of clustering are time series and some modifications of the well-known approaches are proposed related to their characteristics: determination of the optimal number of clusters, different priorities for the time series according to their importance, different weights of the values in the series according to their position in the time axis, statistics about the clustering quality and interpretation of the cluster centers.

2. Clustering of time series

In the general case the objects that should be clustered are represented by temporal or spatial features. In our case these objects are time series and the time ordering of the observations is taken into account. The initial input data are:

- A set of time series that have to be clustered;
- The number of clusters in which the series must be grouped.

The aim is to obtain clusters of time series and every cluster consists of:

- A set of series belonging to the cluster;
- A generated synthetic series, called prototype series, considered as a representative of the cluster. This prototype series is with the same dimensionality as all other series and it is usually near to the center of the cluster.

There are many clustering techniques, for example: k-means [22], hierarchical clustering that can be divisive or agglomerative [21], self-organizing maps (SOM) [13], adaptive resonance theory (ART) [8][9], iterative self-organizing data analysis technique (ISODATA) [21], etc. Here the

clustering experiments are done by self-organizing map, which is an unsupervised trained neural network proposed by Teuvo Kohonen [13].

In addition to the clustering procedure, wich groups time series and generates prototypes series, clustering quality statistics are also generated [21][22]: inter-cluster and intra-cluster distances, R^2, adjuster R^2, etc. Some of these statistics can be used as critaria to determine the optimal number of clusters.

One important application of the clustering module developed by the authors is to reduce large sets of time series to small sets of synthetic prototype series. Thus the prototype series can be used instead of the real series for time and memory consuming operations, such as calculations of correlation matrices, with as minimal error as possible. In this way the number of time series is reduced as any calculation that should be done with a given real time series is actually performed with the cluster prototype to which the real series is classified. The number of all available series is reduced to the number of clusters and as a consequence the calculation operations that must be performed with the series decrease. This should be done because the huge data causes too many calculations, which often cannot be finished within acceptable time period.

3. Proposed solution

The procedure of clustering performed by self-organizing map is a procedure of the unsupervised neural network training. During this process the similarity of the time series behaviour is examined and this is done not only for the current values but for all other historical values as well. That is why it is possible a given series, for example with high current values to be classified into a cluster containing lower current series values. This may occur when a series with different actual values to the other series for the current time moment, but similar historical behaviour to other series in the past.

Series weights

Individual series may have different importance expressed by coefficients called weights. If the weights are all equal to ‘one’, then all series are equally treated in the clustering process. Some of the series however may be considered more important than the others. If the weight of a series is two times bigger than the weight of another series, the former is considered as two series with the same series values in the clustering alorithm. The practical signifficance of this property is that the weights can be not only integers but any arbitrary real values as well.

The effect of the usage of weights is shown in fig. 1.

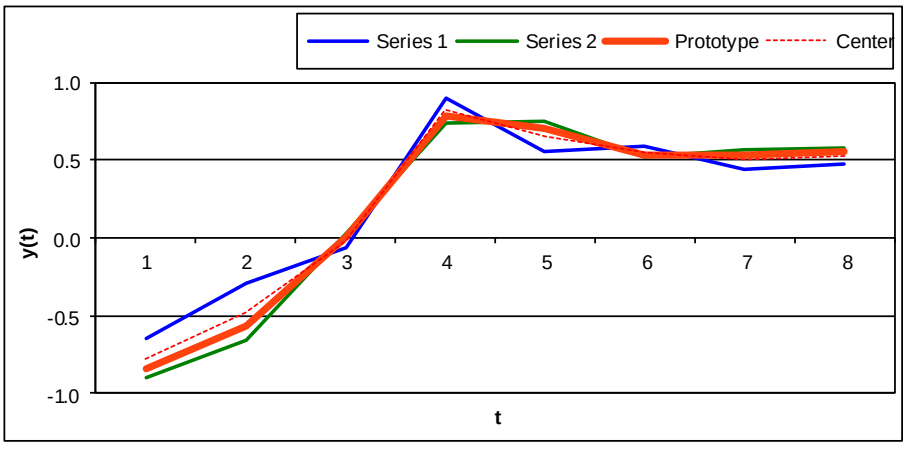


Fig. 1 The effect of a series weight to the cluster prototype

The ceneter between Series 1 and Series 2 is shown with a dashed line. If these two series are with the same weight, the center will be very close to the prototype series. However, in this example, Series 2 is with greater weight and thus it attracts the prototype shown with bold line.

Weights of series values

In the clustering algorithm the series must be compared one another by a chosen criterion. One of the most often used such criterion is the Euclidean distance (1) where compared series must be with the same length [4][10].

$$d = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2} \tag{1}$$

where x and y are the series that are compared;
i is the consequent index of x and y.

The Euclidean distance can take into account the importantce of the values by using a decay factor (2) [17].

$$d = \sqrt{\frac{1}{\sum_{i=1}^{N}\lambda^{i-1}}\sum_{i=1}^{N}\lambda^{N-i}(x_i - y_i)^2} \tag{2}$$

where λ is the decay factor which takes values between 0 and 1.

The closer to the 0 the decay factor is, the more important values are to the end of the series. When the decay factor is 1, then the formula coincides to the original Euclidean distance.

Clustering quality criteria

In order to assess how good the clustering is, a suitable criterion or criteria must be applied [14][17][20]. Some such criteria with their formulas are shown below.

- Adjusted R^2. In fig.2 the main parts of this statistics are shown.

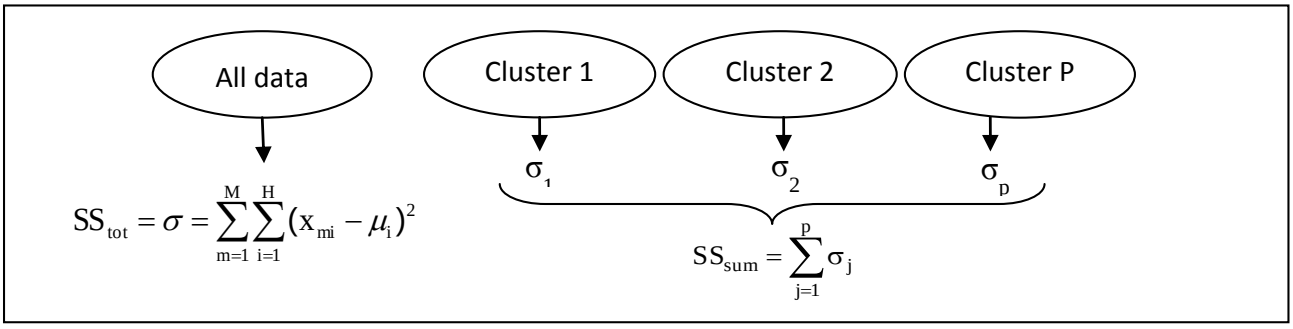


Fig. 2 Calculation of squared distances for Adjusted R^2

The following steps must be performed:

1) For all data the sum squared distance of each series to the center is calculated:

$$SS_{tot} = \sigma = \sum_{m=1}^{M}\sum_{i=1}^{H}(x_{mi} - \mu_i)^2 \tag{3}$$

where M is the number of all available series;

H is the number of values in the series;

x is current time series;

μ is center of all available series.

2) Similarly, the sum squared distance is calculated for each cluster j:

$$\sigma_j = \sum_{m=1}^{M_j} \sum_{i=1}^{H} (x_{jmi} - \mu_{ji})^2 \tag{4}$$

where M_j is the number of time series series in cluster j;

μ_j is the center of the series in cluster j.

3) The sum of the squared distances calculated in step 2 is calculated:

$$SS_{sum} = \sum_{j=1}^{P} \sigma_j \tag{5}$$

where P is the number of clusters.

4) R^2 is calculated as follows:

$$R^2 = 1 - \frac{SS_{sum}}{SS_{tot}} \tag{6}$$

5) And finally the adjusted R^2 is calculated as:

$$\text{Adjusted}R^2 = 1 - (1 - R^2)\frac{M-1}{M-P-1} = 1 - \frac{SS_{sum}}{SS_{tot}}\frac{M-1}{M-P-1} \tag{7}$$

- Euclidean distances between the cluster prototype series – fig 3. Calculated Euclidean distances can be regarded as other useful statistics about the clustering quality and they can be further processed to find the average distance, maximal distance, etc.

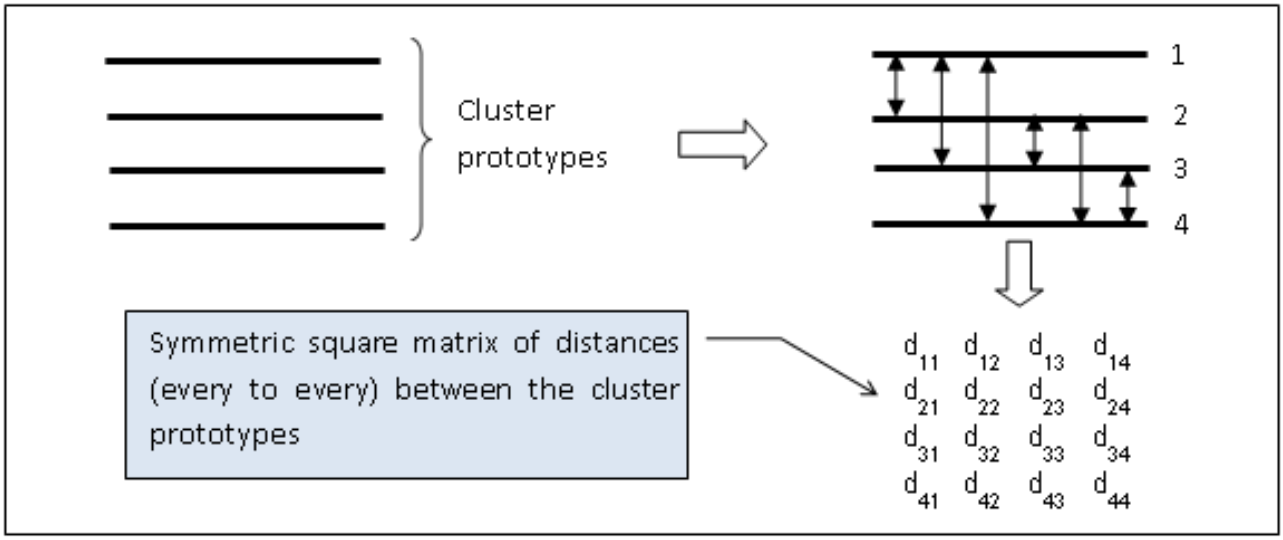


Fig. 3 Euclidean distances between cluster prototypes

- Euclidean distances can be calculated also for every cluster – fig.4. In this case the distances from every time series in the cluster to the center of the cluster are considered. The center is with the same dimensionality as the time series in the cluster and every value of the center is defined as the average value of the corresponding values of time series classified to that cluster.

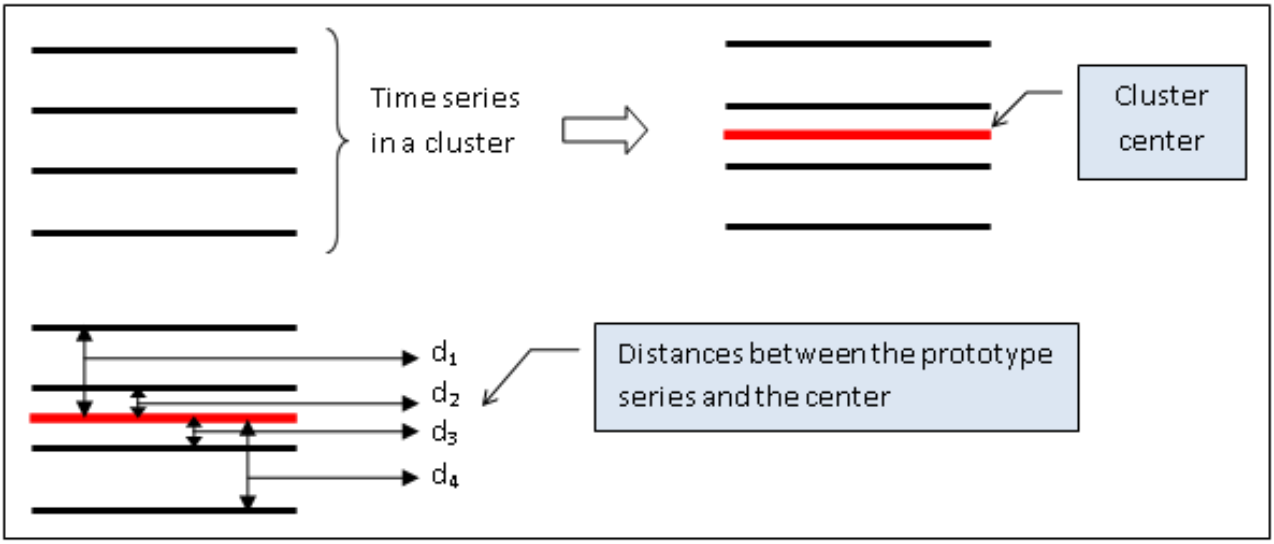


Fig. 4 Euclidean distances between the time series in a cluster and its center

- Average linkage. This statistic is based on calculation of the distances between clusters. The distance between two clusters is defined as the average distance between the series classified to these clusters. The distance between cluster X and cluster Y is calculated in the following way:

$$d_{xy} = \frac{1}{N_x N_y} \sum_{i=1}^{Nx} \sum_{j=1}^{Ny} d(x_i, y_j), \quad x_i \in X, y_j \in Y \tag{8}$$

where N_x and N_y are the number of series in cluster X and Y respectively;
$d(x_i, y_j)$ is the Euclidean distance between series i from cluster X and series j from cluster Y.

Determination of the optimal number of clusters

Finding of the optimal number of clusters is based on an appropriate criterion or criteria that must be defind beforehand. After that one possible way to proceed is to cluster available data for all possible number of clusters (1, 2,…, n) and for each of thesee clustering attempts the chosen criterion is calculated. The best quality value of the criterion determines the best number of clusters.

More often in order to find the optimal number of clusters in practical applications the criterion for optimal clustering is plot in two-dimensional space where X axis represents the number of clusters and Y axis the clustering quality. The sharp drop end point in the graphic, determines the optimal number of clusters. In fig. 5 an example is shown where the criterion is adjusted R^2.

Num Clusters	Adjusted R squared	Error	Time (sec.)
2	0.7888814	0.2111186	20
3	0.8856307	0.1143693	32
4	0.9281010	0.0718990	40
5	0.9351225	0.0648775	60
6	0.9360977	0.0639023	70
7	0.9361842	0.0638158	95
8	0.9647925	0.0352075	109
9	0.9543623	0.0456377	122
10	0.9544117	0.0455883	144
11	0.9758081	0.0241919	154
12	0.9757913	0.0242087	173
13	0.9572335	0.0427665	180
14	0.9572007	0.0427993	194
15	0.9571655	0.0428345	218
16	0.9573212	0.0426788	225
17	0.9572855	0.0427145	260
18	0.9861978	0.0138022	276
19	0.9861863	0.0138137	287
20	0.9861746	0.0138254	305
21	0.9861818	0.0138182	314
22	0.9861700	0.0138300	326
23	0.9861583	0.0138417	362
24	0.9861466	0.0138534	381
25	0.9861356	0.0138644	379
26	0.9861230	0.0138770	429

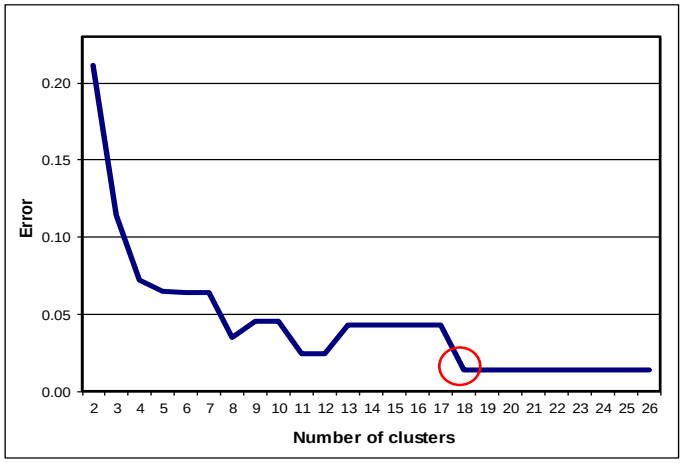


Fig. 5 Clustering of 1200 time series in consequent numbers of clusters and clustering quality calculation based on an adjusted R^2 and Euclidean distance (Error) to find the optimal number of clusters

Another more effective procedure regarding the number of calculation operations is to perform all possible clustering not in consequntive numbers of clusters but over some fixed number k [15][18] as it is shown in fig. 6.

Num Clusters	Adjusted R Squared	Error
5	0.9351225	0.0648775
10	0.9544117	0.0455883
15	0.9571655	0.0428345
20	0.9861746	0.0138254
25	0.9861356	0.0138644
30	0.9868221	0.0131779
35	0.9875898	0.0124102
40	0.9878886	0.0121114

Num Clusters	Adjusted R Squared	Error
15	0.9571655	0.0428345
16	0.9573212	0.0426788
17	0.9572855	0.0427145
18	0.9861978	0.0138022
19	0.9861863	0.0138137
20	0.9861746	0.0138254

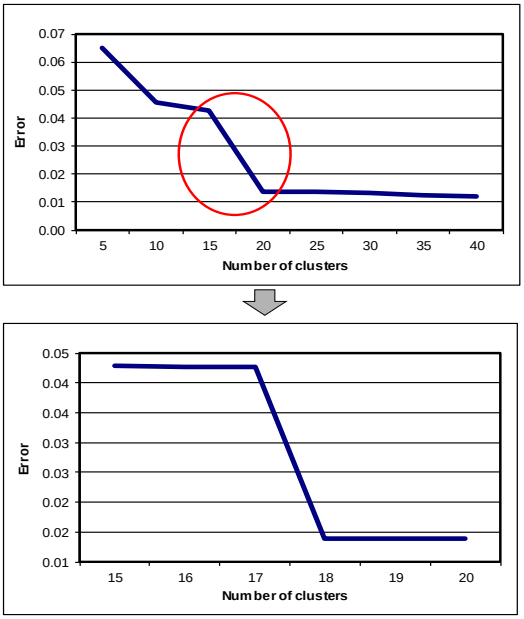


Fig.6 Two level of searching of optimal number of clusters - the first level is through 5 clusters and the second level is performed by consequtive number of clusters

In this example k = 5 and clustering is performed in 5, 10, 15, …, N number of clusters and for all of these attempts the clustering quality is plot to find the sharp drop end point i. After that additional clusterings in i+1, i+2, …, i+k-1 clusters are performed in order to find the optimal number of clusters more precisely. This is two levels searching of optimal number of clusters.

Conclusion and future work

In fig. 7 an example of 4 clusters are shown with their time series and their prototypes in bold line. It can be seen that the similar time series are grouped in the same cluster. The current clustering module is used mainly for reduction of the number of series in case of simulation calculations [1] [16]. For example, in case of correlation matrix calculations, where the correlations between all possible couples of series must be calculated, the time needed even for powerful machine is too much [19]. By clustering the number of series is reduced to the number of clusters as when a given series is needed for correlation calculation it is represented by the prototype of the cluster in which the series is classified.

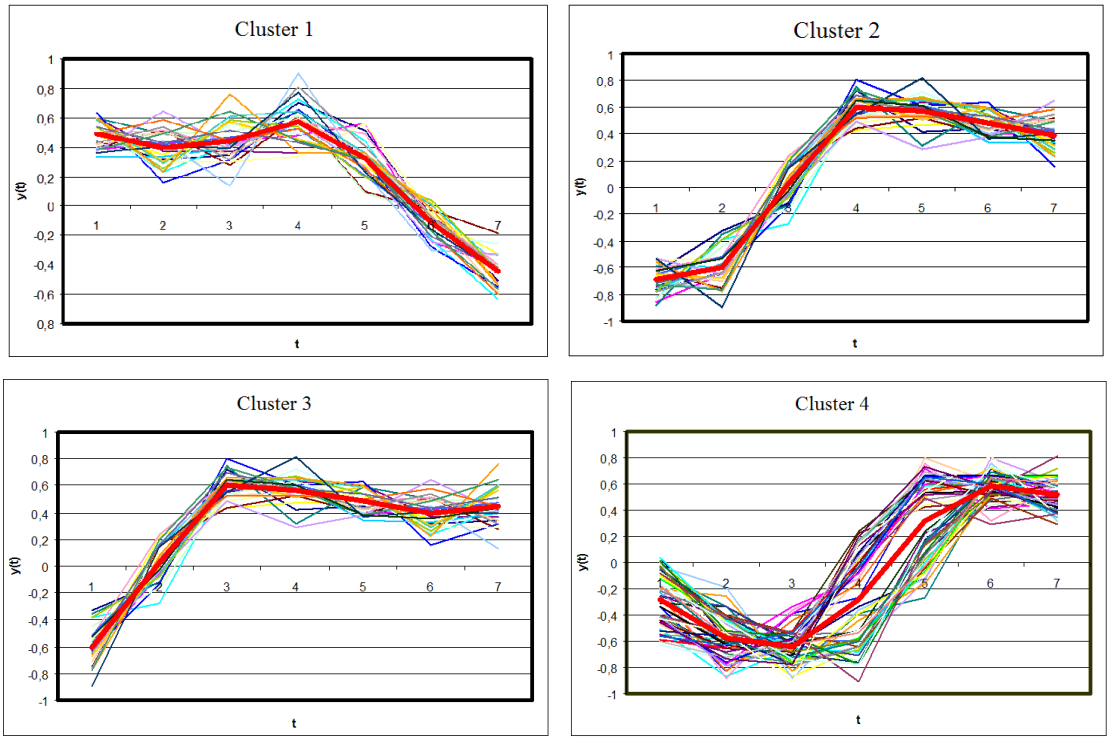


Fig. 7. Examples of clusters, their prototypes illustrated as a bold red line

It is possible alternative clustering approaches to be developed and compared to the developed one using the same or other additional criteria for clustering quality.

References

[1]. Bury, K. Statistical Distributions in engineering. Cambridge University Press, 1999.

[2]. Cameron, C. Trivedi, P. Regression Analysis of Count Data. Cambridge university press, 1998.

[3]. Carpenter, G., S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics and Image Processing, no. 37, Academic Press Professional Inc., 1987, pp. 54-115.

[4]. Chatfield, C. The analysis of time series. An introduction. Fifth edition. Chapman & Hall/CRC, 1996.

[5]. Draper, N. Smith, H. Applied Regression Analysis. Wiley Series in Probability and Statistics, 1998.

[6]. Forgy, E. Cluster analysis of multivariate data: efficiency vs. interpretability of classifications. Biometrics no. 21, 1965. pp. 768-780.
[7]. Grossberg, S. Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. Biological Cybernetics, no. 23, 1976. pp. 121-134.
[8]. Grossberg, S. Adaptive pattern recognition and universal encoding II: Feedback, expectation, olfaction, and illusions. Biological Cybernetics no. 23, 1976. pp. 187-202.
[9]. Grossberg, S. Competitive learning: From interactive activation to adaptive resonance. Cognitive Science, 11, 1987, pp. 23-63
[10]. Hamilton, J. Time Series Analysis. Princeton University Press, ISBN: 0-691-04289-6, 1994.
[11]. Hansen, P., B. Jaumard. Cluster analysis and mathematical programming. Mathematical Programming (79), 1997, pp. 191-215.
[12]. Jain, A., R Dubes. Algorithms for Clustering Data. Prentice-Hall, New Jersey, 1998.
[13]. Kohonen, T. Self-Organizing Maps. Springer, 2001.
[14]. Krishnamoorthy, K. Handbook of statistical distributions with applications. Chapman & Hall, 2006.
[15]. Maiorana, F. Performance improvements of a Kohonen self organizing classification algorithm on sparse data sets. Proceedings of the 10th WSEAS international conference on Mathematical methods, computational techniques and intelligent systems, Corfu, Greece, 2008. pp. 347-352.
[16]. Mun, J. Modeling Risk. Applying Monte Carlo Simulation, Real Options Analysis, Forecasting, and Optimization Techniques. Wiley, 2006.
[17]. O'Neill, B. Elementary Differential Geometry. Academic Press, 2006.
[18]. Palit, A. K., D. Popovic. Computational intelligence in time series forecasting. Theory and engineering applications. Springer-Verlag, 2005.
[19]. Robert, C., Casella, G. Monte Carlo statistical methods. Springer-Verlag, 1999.
[20]. Vallentin, M. Probability and Statistics Cookbook, 2011.
[21]. Xu, R. Wunsch, C. Clustering. Wiley, 2009.
[22]. Zhang, R., A. Rudnicky. A large scale clustering scheme for kernel K-means. Proceedings of the 16th International Conference on Pattern Recognition, Vol. 4, 2002, pp. 289-292.
[23]. Мандель, И. Кластерный анализ. Финансы и статистика, Москва, 1988.

For contacts:
Assoc. Prof. Ventsislav Nikolov
Department of Computer Science and Engineering
Tchnical University of Varna
E-mail: v.nikolov@tu-varna.bg

АТАКИ НА ВЗАИМНУЮ СИНХРОНИЗАЦИЮ СЕТЕЙ В КРИПТОГРАФИИ

Александров Никита

Abstract: В данной статье представлены экспериментальные результаты оценки влияния архитектуры древовидных машин четности на вероятность взлома методом параллельного обучения. Древовидные машины четности предложены в качестве модификации алгоритма симметричного шифрования. Главное преимущество метода заключается в использовании взаимной синхронизации нейронных сетей для генерации одинакового ключа у двух конечных пользователей одновременно, без надобности передачи ключа по сети. Определена степень влияния количества нейронов в сети на устойчивость системы к внешним атакам. Определены задачи дальнейших исследований.
Keywords: нейронные сети, древовидные машины четности, время синхронизации, криптография, шифрование, криптостойкость, атака, входные нейроны, скрытые нейроны, взлом.

1. Введение

В современных реалиях для человечества остро стоит вопрос сохранности своих данных, особенно цифровых. Для защиты данных используются всевозможные системы шифрования, позволяющие путем преобразования информации в зашифрованный вид сокрыть ее от случайного или же намеренного раскрытия посторонними. В основе большинства современных систем шифрования лежит идея преобразования информации в сокрытый вид при помощи ключа [1, 2]. Такие системы шифрования условно разделяют на два типа: симметричные и асимметричные системы шифрования.

• Симметричные системы шифрования базируются на использовании единственного ключа, который используется и для шифрования информации, и для дальнейшей расшифровки информации [5, 6]. При таком подходе ключ может передаваться только по защищенным каналам связи и в случае его перехвата вся ценность шифрования теряется и информацию больше нельзя считать защищенной.

• Асимметричные системы шифрования основаны на использовании двух различных ключей: открытого ключа, который свободно передается по незащищенному каналу и закрытого ключа, который остается у пользователя и никуда не предается [7]. Ценность такого подхода состоит в том, что зашифрованное одним ключом сообщение можно расшифровать только при помощи второго ключа и наоборот. Системы ассиметричного шифрования являются очень ресурсоемкими и уступают в скорости системам симметричного шифрования. Однако подобные системы имеют значительно большую криптографическую стойкость и поэтому получили большую распространенность [4, 8].

Криптографическая стойкость всех алгоритмов неуклонно снижается с каждым днем. Это связанно с непрекращающимся техническим прогрессом, постоянным ростом вычислительной мощности техники и появлением новых, более совершенных методов обработки данных. Для постоянного обеспечения достаточной криптостойкости алгоритмы нуждаются в постоянной модификации. Одной из таких потенциальных модификаций систем симметричного шифрования, может выступить модификация с использованием явления взаимной синхронизации нейронных сетей [9].

2. Древовидные машины четности

Взаимная синхронизация нейронных сетей – это явление, которое нашло применение в криптографии. Суть явления заключается в том, что при подаче одинаковых наборов данных на две нейросети прямого распространения с особенной структурой со временем достигается полная синхронизация таких сетей. Явление синхронизации может встречаться в совершенно разнообразных физических или биологических системах. Однако в случае с нейросетями – взаимная синхронизация означает полное совпадение значений синапсов двух идентичных по своей архитектуре сетей [11, 12]. Нейронные сети прямого распространения, которые используются в криптографии для взаимной синхронизации называются древовидными машинами четности (ДМЧ) [10]. Такие нейронные сети имеют следующую структуру: один выходной нейрон, K скрытых нейронов и $K \times N$ входных нейронов. Структура данной нейронной сети представлена на рисунке 1.

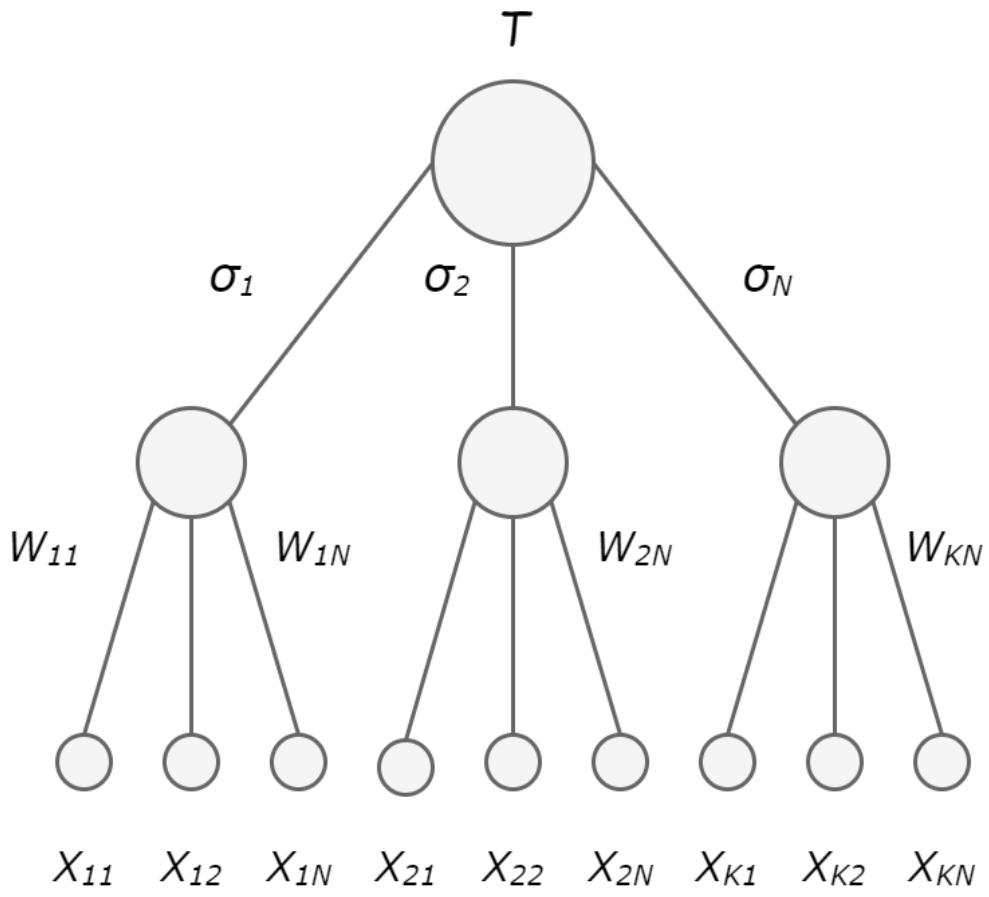


Рис. 1. Структура ДМЧ

Входные нейроны принимают двоичные значения:

$$x_{ij} \in \{-1, +1\}. \quad (1)$$

Веса между входными и скрытыми нейронами принимают значения:

$$w_{ij} \in \{-L, \dots, 0, \dots, +L\}. \quad (2)$$

Значением для каждого скрытого нейрона выступает сумма произведений входного значения и весового коэффициента:

$$\sigma_i = sgn\left(\sum_{j=1}^{N} w_{ij}x_{ij}\right), \quad (3)$$

$$sgn(x) = \begin{cases} -1\ if\ x \le 0, \\ 1\ if\ x > 0. \end{cases} \quad (4)$$

Значением выходного нейрона выступает произведение всех скрытых нейронов:

$$\tau = \prod_{i=1}^{K} \sigma_i, \quad (5)$$

Выходное значение при таких условиях также является двоичным [12].

Для модификации весовых коэффициентов сетей существует несколько правил обучения: правило Хебба, анти-правило Хебба, случайное блуждание. Однако лучшие результаты показывает именно правило Хебба [12, 15].

После достижения синхронизации между двумя ДМЧ их можно использовать для создания ключа. Ключ создается на основе значений весов этих нейронных сетей [12, 13]. Их можно использовать для одноразового блокнота или же для выбора первого числа псевдослучайной последовательности, или для любого другого способа генерации ключа. Главное преимущество данного подхода состоит в том, что ключ можно параллельно создать у двух конечных пользователей и его не нужно передавать по сети. Что значительно уменьшает вероятность взлома такого ключа.

Исходя из структуры ДМЧ становится очевидно, что для усложнения потенциального ключа необходимо увеличивать количество скрытых синапсов. Сделать это можно путем увеличения количества нейронов как скрытых, так и входных. С увеличением количества нейронов в структуре нейронной сети увеличится потенциальная криптостойкость ключа. Однако увеличения количества нейронов увеличит и время синхронизации нейронных сетей. Потенциально также должна увеличится степень защиты от взлома системы в целом, ведь даже при скрытом подключении злоумышленника к каналу связи объем передаваемых данных возрастет что уменьшит вероятность их быстрой обработки.

3. Атаки на алгоритм

Потенциальные атаки на криптографический алгоритм с использованием нейронных сетей схожи с атаками на другие алгоритмы. Необходимо учитывать, что при моделировании атаки на алгоритм необходимо делать некоторые допущения. В данном случае будет допускаться что злоумышленник может получать данные с канала связи между абонентами А и Б, однако не может их изменять. В таком случае можно рассмотреть следующие виды атак:

• Метод грубой силы. Данный метод основан на переборе всех возможных вариантов ключей, то есть все возможные вариации значений синапсов w_{ij}. Также вводится допущение что злоумышленник знает какая используется архитектура нейронной сети и сколько в ней скрытых синапсов. И даже этом случае для K скрытых нейронов, $K \times N$ входных нейронов и максимальном весе L мы получим $(2L + 1)^{\mathrm{KN}}$ вариантов [14]. Для $K = 5$, $L = 5, N = 50$ мы получи количество вариантов равное $(11)^{250}$, что в данный момент невозможно вычислить, поэтому эта атака заведомо несостоятельная.

• Метод параллельного обучения. Данный метод предполагает, что злоумышленнику полностью известна структура используемой ДМЧ и он может ее воспроизвести. В таком случае если злоумышленник может прослушивать канал связи, то существует вероятность что его ДМЧ также сможет синхронизироваться с ДМЧ абонентов. Вероятность этого

события, а также зависимость удачного взлома от архитектуры предлагается исследовать путем моделирования данной атаки.

• Генетическая атака. Данный вид является некой модификацией метода параллельного обучения. Предполагается что злоумышленник создает не одну ДМЧ для попытки параллельной синхронизации, а целую популяцию. И в процессе обучения отсеивает недостаточно пригодные ДМЧ и наращивает потенциально благоприятные для взлома [14]. Однако успех данного мероприятия в целом зависит от возможности синхронизации хотя-бы одной сторонней машины, а также от максимально допустимого размера популяции которое может одновременно обрабатывать злоумышленник. Данный алгоритм потенциально опасен лишь на ДМЧ малой размерности, ибо в противном случае злоумышленнику просто не хватит вычислительных мощностей для развертывания популяции достаточной размерности.

Из вышеперечисленных атак наиболее опасной представляется метод параллельного обучения. Для выяснения степени опасности, а также минимально необходимой структуры ДМЧ во избежание взлома был проведен следующий эксперимент. Были реализованы две взаимно синхронизирующихся ДМЧ с одинаковой архитектурой, а также третья ДМЧ, которая пыталась параллельно синхронизироваться. Синхронизировалась третья ДМЧ используя теже вектора и ответы нейронных сетей, но без возможности повлиять на процесс обновления весов пользовательских ДМЧ. И постепенно увеличивая размерность ДМЧ путем увеличения количества скрытых нейронов были реализованы множественные попытки атаки параллельным обучением.

Робота синхронизирующихся ДМЧ была организована таким образом чтобы при синхронизации конечных абонентов робота не прекращалась, а продолжалась пока количество итераций не достигнет 10000, пользовательская ДМЧ также продолжала обрабатывать вектора и пересылать результат по сети. Это было реализовано для того, чтобы узнать за сколько итераций после синхронизации могла бы синхронизироваться сеть злоумышленника, ведь количество итераций необходимых для синхронизации не постоянно и иногда может достигать больших значений.

В таком случае у нас есть 3 варианта исхода:

• ДМЧ злоумышленника синхронизировалась раньше или одновременно с ДМЧ пользователей.

• ДМЧ злоумышленника синхронизировалась позже, чем ДМЧ пользователей, но количество итераций не превысило 10000.

• ДМЧ злоумышленника не синхронизировалась за 10000 итераций.

Первый вариант является критическим и говорит нам об уязвимости системы, второй об потенциальной опасности, третий об условной безопасности системы. Для тестирования была реализована следующая архитектура нейронной сети:

• $K = 3$ скрытых нейронов,

• $K \times 3$ входных нейронов,

• $L = 1$, диапазон значений, принимаемых синапсами. Вещественное число с точностью 6 знаков после запятой.

Для каждой архитектуры проводилось 5000 тестовых попыток синхронизации, после чего количество скрытых нейронов увеличивалось и эксперимент повторялся вновь. Результаты представлены на рисунках 2-5.

В качестве дополнительной меры было принято решение провести аналогичное исследование с изменением и количества скрытых и количества входных нейронов. Для этого было проведено 25 испытаний по 5000 синхронизаций для различных вариантов архитектур ДМЧ. Таким образом была получена более наглядная статистика демонстрирующая зависимость вероятности взлома от количества скрытых синапсов. Результаты данного эксперимента представлены на рисунке 6.

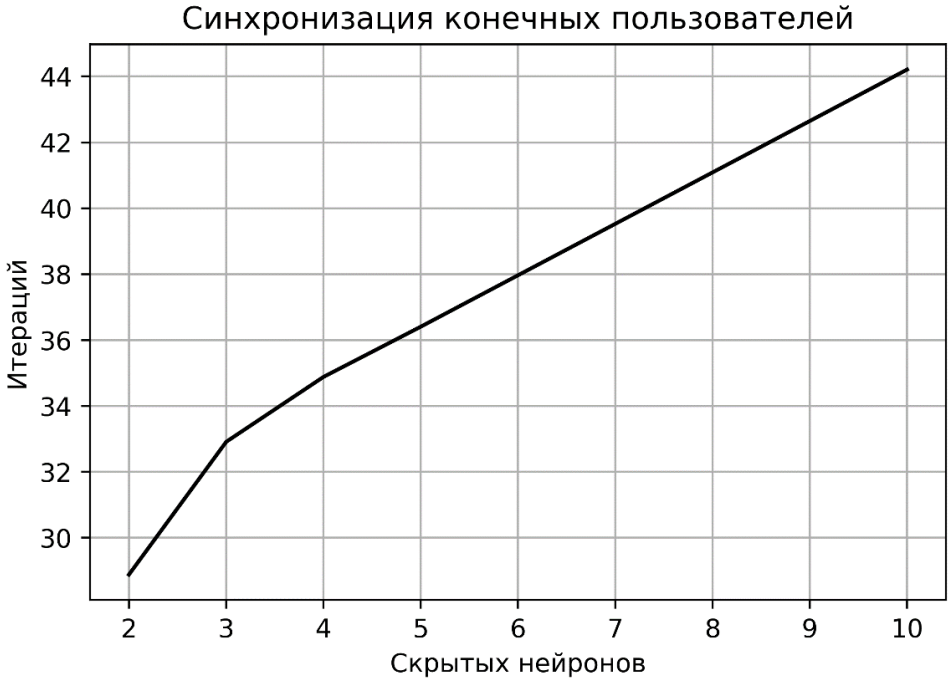


Рис. 2. Среднее количество итераций необходимых для синхронизации нейронных сетей.

Как можно видеть на графике рис. 2, при увеличении количества скрытых нейронов в сети, количество итераций необходимое для синхронизации постоянно увеличивается, однако скачок не является резким и позволяет значительно увеличить количество скрытых нейронов.

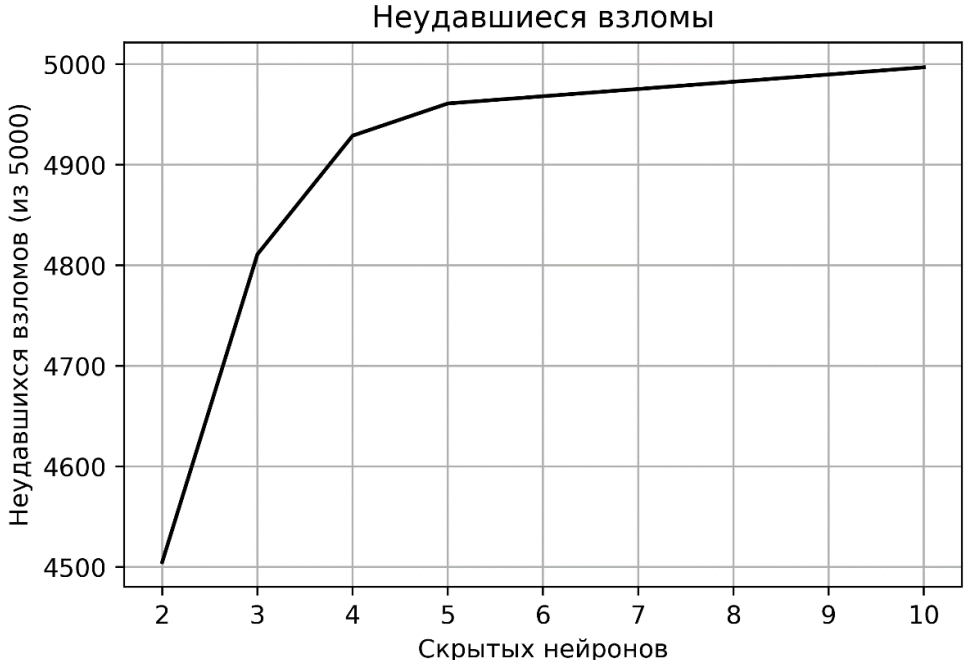


Рис. 3. Количество безуспешных попыток синхронизации нейронной сети злоумышленника

Анализируя график, представленный на рисунке 3, можно сделать вывод что при увеличении количества скрытых нейронов до 5, резко падает количество успешных или условно-успешных взломов, однако удачные синхронизации все-равно есть и пропадают только при увеличении количества скрытых нейронов до 10. Однако не следует забывать,

что на каждый скрытый нейрон в данных архитектурах приходится всего 3 входных, что является очень малым значением и не используется в реальных условиях.

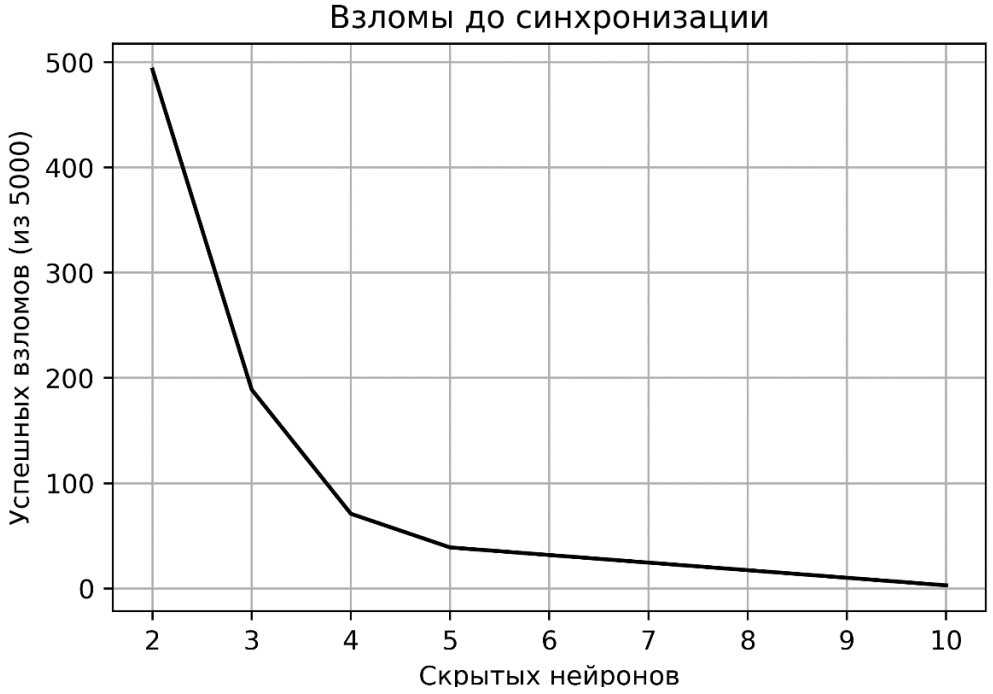


Рис. 4. Количество успешных взломов нейронной сети до синхронизации ДМЧ абонентов

Как показано на графике рис. 4, при повышении количества скрытых нейронов до 5 количество успешных взломов сокращается до 1% от общего количества попыток, при достижении 10 скрытых нейронов, успешных взломов не наблюдается.

Рис. 5. Количество успешных взломов нейронной сети после синхронизации ДМЧ абонентов

В сравнении с количеством успешных взломов рис. 4, количество условно успешных взломов рис. 5 (после синхронизации абонентов, и до 10000 итераций) значительно меньше,

и уже при 3 скрытых нейронах таких случаев не наблюдается. Связано это с тем, что при достижении взаимной синхронизации между абонентами они перестают друг на друга влиять, и без обновления весов вероятность синхронизации злоумышленника резко падает.

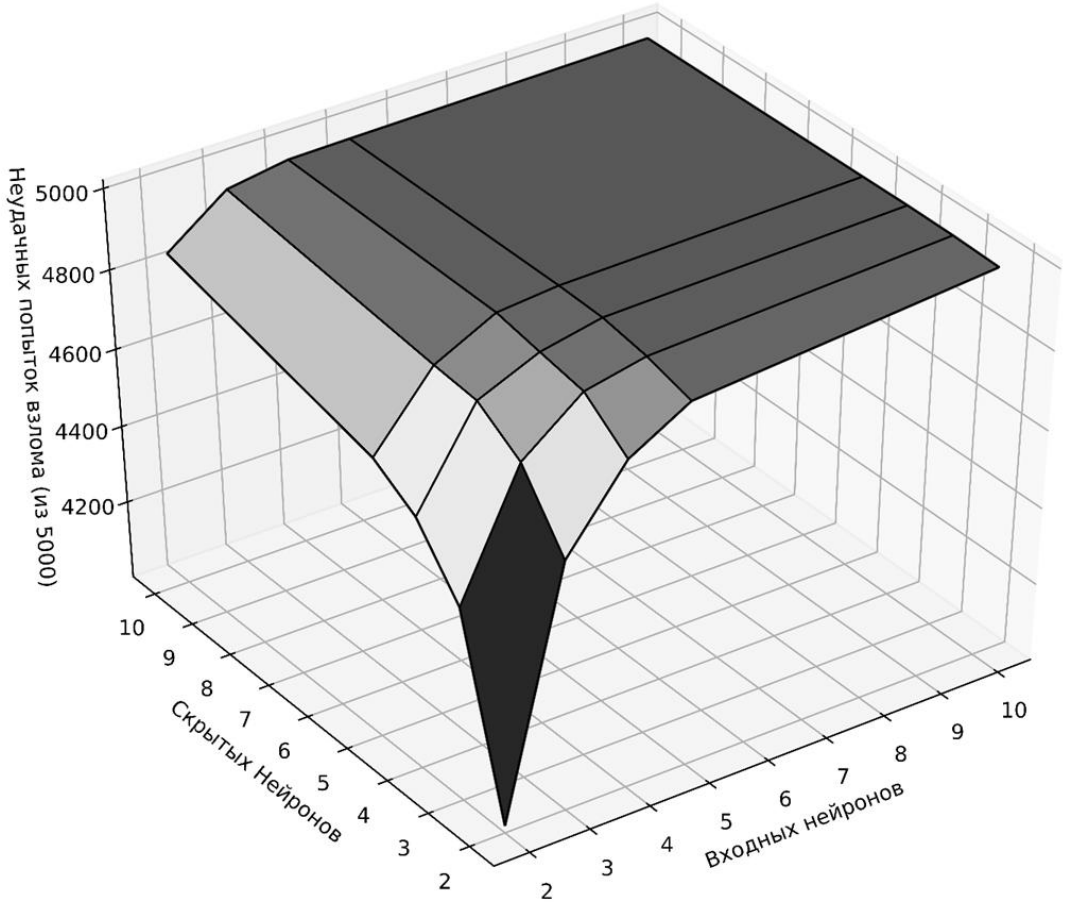


Рис. 6. Зависимость Количества неудачных взломов нейронной сети от архитектуры

График, представленный на рисунке 6, демонстрирует как возрастает стойкость алгоритма к атаке методом параллельного подключения при увеличении количества скрытых и входных нейронов. Из графика отчетливо видно, что при увеличении архитектуры до 4x4 результат уже выходит на условное плато, вероятность взлома в котором минимальна. при увеличении архитектуры до 10x10 успешные и условно успешные взломы отсутствуют.

3. Заключение

В данном исследовании экспериментально оценивалась степень влияния архитектуры нейронных сетей на вероятность взлома алгоритма одновременной синхронизацией сторонней сети. Все эксперименты проводились без учета задержки в сети и производительности устройств, на которых выполняется синхронизация. Это позволило получить достоверную информацию о степени влияния архитектуры нейронной сети на вероятность ее взлома. Результаты экспериментов позволяют сделать вывод, что даже при использовании нейронных сетей малой размерности и делая большое количество допущений, вероятность взлома сети крайне мала. И при необходимости достичь большей криптостойкости достаточно незначительно увеличить либо размерность сети или диапазон значений L. Также эксперименты продемонстрировали куда большую вероятность синхронизации сторонней нейронной сети до синхронизации нейронных сетей абонентов чем после. Это напрямую связанно с совпадением этапов обновления весов у

злоумышленника с конечными абонентами на сетях с малой размерностью и при прекращении этого процесса вероятность синхронизироваться стремится к 0. А даже при незначительном усложнении архитектуры нейронной сети вероятность взлома резко падает.

В будущем планируется экспериментально проверить возможность генетической атаки на алгоритм, а также ее потенциальную ресурсоемкость. На основе полученных данных будет предложена архитектура нейронной сети с приемлемым временем синхронизации и достаточной криптографической стойкостью. Также предполагается более детальное исследование вопроса криптографической стойкости полученного подхода с учетом других методов атаки.

References

[1]. Schneier, Bruce. Applied cryptography: protocols, algorithms, and source code in C. john wiley & sons, 2007.

[2]. Shannon, K. E. "The theory of communication in secret systems." Work on the theory of information and cybernetics, Moscow (1963): 333-402.

[3]. Mao, Wenbo. Modern cryptography: theory and practice. Pearson Education India, 2003.

[4]. Ferguson, Niels, and Bruce Schneier. Practical cryptography. Vol. 141. New York: Wiley, 2003.

[5]. Pieprzyk, Josef, Thomas Hardjono, and Jennifer Seberry. Fundamentals of computer security. Springer Science & Business Media, 2013.

[6]. Zhelnikov, V. "Cryptography from papyrus to computer." ABF, Moscow (1996).

[7]. Diffie, Whitfield, and Martin Hellman. "New directions in cryptography." IEEE transactions on Information Theory 22.6 (1976): 644-654.

[8]. Forouzan B., Cryptography and Network Security / B. Forouzan // First Edition. McGraw-Hill. – 2007. – 721 p.

[9]. Klein, Einat, et al. "Synchronization of neural networks by mutual learning and its application to cryptography." Advances in Neural Information Processing Systems. 2005.

[10]. Volkmer, Markus, and Sebastian Wallner. "Tree parity machine rekeying architectures." IEEE Transactions on Computers 54.4 (2005): 421-427.

[11]. Ahmed M. IEEE On the Improvement of Neural Cryptography Using Erroneous Transmitted Information with Error Prediction / Ahmed M. Allam and Hazem M. Abbas, Member // IEEE transactions on neural networks. – 2010. – Vol. 21, no. 12. – P. 1915–1924.

[12]. Червяков, Николай, et al. Применение искусственных нейронных сетей и системы остаточных классов в криптографии. Litres, 2018.

[13]. Kinzel W. Neural cryptography / Kinzel W., Kanter I. // 9th International Conference on Neural Information Processing. – 2002. – Vol.3. – P. 1351-1354.

[14]. Andreas Ruttor, Wolfgang Kinzel, Rivka Naeh, and Ido Kanter. Genetic attack on neural cryptography (англ.) // Physical Review E : journal. — 2006.

[15]. М.О. Александров, Є.О. Башков, "The use of interacting neural networks in cryptography", Наукові праці Донецького національного технічного університету, серія: "Інформатика, кібернетика та обчислювальна техніка", Покровськ, Україна, 2020, сс. 19-24.

For contacts:
Graduate student, Mykyta, Aleksandrov
Department of Applied Mathematics and Informatics
Donetsk National Technical University
E-mail: neckich@gmail.com

СТЕГАНОГРАФСКО ВГРАЖДАНЕ НА ИНФОРМАЦИЯ В ИЗОБРАЖЕНИЕ ЧРЕЗ ШАБЛОННА МАТРИЦА, ЗАВИСЕЩА ОТ ДЪЛЖИНАТА НА СЪОБЩЕНИЕТО

Димитър Г. Тодоров

Резюме: Поради възхода на интернет в днешно време най-важният фактор на информационните технологии и комуникацията е сигурността на информацията. Данните стават съществена част от всяка организация от няколко десетилетия. Тези данни носят някаква секретна информация, свързана с организацията. Злоупотребата с тези данни може да навреди на тази организация. Стеганографията е едно от решенията в тази насока. Тук тя се използва с различен подход и метод – чрез шаблонна матрица, зависеща от дължината на съобщението. Посредством тази матрица се отсява оригиналното съобщение, вградено преди това в изображение. При опитната постановка са постигнати добри резултати, като процентът на успешно извлечените съобщения е над 90%, а обработените изображения с вградени в тях съобщения са с приемливи размери, подходящи за използване в комуникационна среда.
Ключови думи: стеганография, шаблон, матрица, дължина на съобщение

Steganographic Embedding of Information in an Image Using a Template Matrix Depending on the Length of the Message

Dimitar G. Todorov

Abstract: Due to the rise of the Internet today, the most important factor in information technology and communication is information security. Data has been an essential part of any organization for several decades. This data carries some classified information related to the organization. Misuse of this data can harm this organization. Steganography is one of the solutions in this direction. Here it is used with a different approach and method - through a template matrix depending on the length of the message. This matrix is used to screen the original message previously embedded in an image. The experimental staging achieved good results, as the percentage of successfully retrieved messages is over 90%, and the processed images with embedded messages are of acceptable size, suitable for use in a communication environment.
Keywords: steganography, template, matrix, message length

1. Въведение

Стеганографията се занимава със съставяне на скрити съобщения, така че само подателят и получателят да знаят, че съобщението дори съществува. Тъй като никой освен подателя и получателя не знае съществуването на съобщението, това не привлича нежелано внимание [1]. Стеганографията се е използвала още в древни времена и тези древни методи се наричат физическа стеганография. Някои примери за тези методи са съобщения, скрити в тялото на съобщенията; съобщения, написани с тайни мастила; съобщения, написани на пликове в области, покрити с печати и др. Съвременните методи на стеганографията формират цифровата стеганография. Тези методи включват скриване на съобщения в шумни изображения, вграждане на съобщение в произволни данни, вграждане на снимки със съобщението във видео файлове и др.[2]

Има много видове методи за използване на стеганография. По-долу са класифицирани различните категории според файловите формати, които се използват за техники на стеганография [2],[3].

- Текстова стеганография

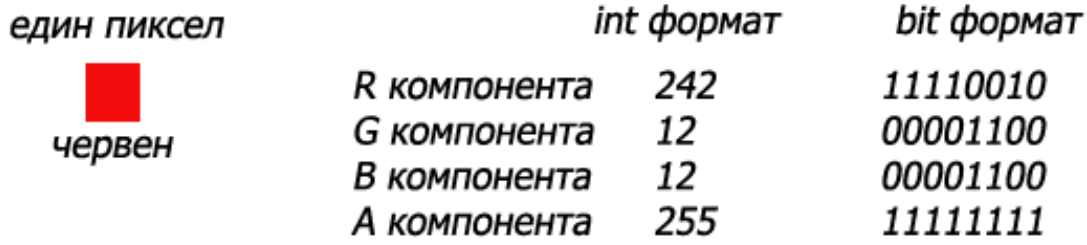
- Стеганография с изображения
- Аудио стеганография
- Видео стеганография

Тук акцентът е поставен върху стеганографията с изображения или скриването на полезната информацията чрез вграждането ѝ в изображението. При този модел изображението се явява преносител на информацията. В стила на стеганографията е важно нанесените промени на изображението, носител след вграждането на полезната информацията, да не доведат до негова осезаема промяна, т.е. оригиналното изображение и промененото – стеганографското, почти не трябва да имат визуални разлики. В противен случай системата би събудила външен интерес, следствие на доловимата разлика.

2. Изложение

2.1. Битово представяне на пиксела

Широко използван метод при вграждането е въвеждането на съобщението побитово, т.е. съобщението се преобразува от текстово в битово съобщение, а след това неговите битове се импортират в тези на изображението, заместващи някои от тях. Най-често се използва последният бит. Едно изображение с 24 битова дълбочина съдържа в себе си пиксели с 3 по 8 бита части за всеки компонентен цвят червен (R - red), зелен (G – green) и син (B –blue) [3]. За опитната постановка на това изследване са избрани за използване изображения във формат BMP. Това е некомпресиран файлов формат за изображения, разработен от Microfsoft. Освен трите компонента RGB в 32 битовата си версия (32 bpp), този формат съдържа и четвърта компонента A, която определя осветеността на пиксела. Целта на опитната постановка е да се постави точно в тази компонента съобщението, което трябва да се скрие.



Фиг.1 Състав на един пиксел в BMP файл

2.2 Процес на вграждане на съобщение в изображение

Процесът на вграждане на информацията (текста, съобщението) в изображението се състои от следните стъпки:

1. Конвертиране на съобщението от текстов формат в двоичен вид и определяне на размера му;
2. Разбиване на изображението - носител на матрица на съставните му пиксели;
3. Вграждане на конвертираното съобщение в изображението.

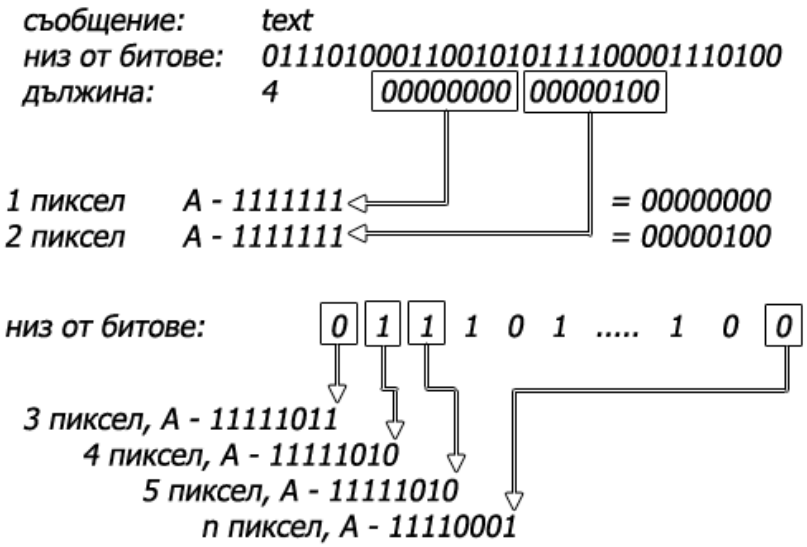
Съобщение: "text"		Дес. номер в ASCII	Двоично представяне
по символи:	t	116	01110100
	e	101	01100101
	x	120	01111000
	t	116	01110100

Фиг.2. Представяне на текста в двоичен вид

Процесът на конвертиране на текстовото съобщение е просто конвертиране на символите в низа в двоичен вид с дължина от 8 бита, при ASCII кодираща таблица – фигура 2. След това всички символи се връщат обратно в низа, но в двоичен вид. По този начин се получава резултат - низ от битове. Дължината на съобщението се взема се под внимание, тъй като тя ще бъде вградена първа и преди самото съобщение. Дължината се определя по следната опростена формула:

$$L_{msg} = \frac{L_{bit}}{8}, \tag{1}$$

където L_{msg} е дължината на съобщението, а L_{bit} е дължината на целия низ от битове генериран при конвертирането на съобщението.



Фиг.3. Процес на вграждане на дължината и съобщението

Изображението се разлага на съставна матрица от пиксели, като всеки пиксел е във вида от фигура 1, а акцентът е върху битовете от компонента А. Първите два пиксела се използват за вграждане на дължината на съобщението. Използва се цялата им дължина от битове, т.е. 2 по 8 бита – общо 16 бита. Това би позволило кодирането на 65536 дължина на съобщението. Битовете на компонентата А на първите два пиксела се заместват с едната и другата половина от битове на дължината на съобщението –фигура 3. Последният бит във всеки следващ пиксел за компонента А се заменя с поредния бит от низа от битове на текстовото съобщение, като се започва от първия бит в низа на текстовото съобщение – фигура 3. Максималната дължина на съобщение $L_{t_{MAX}}$, което може да се побере в

изображението-носител, е в зависимост от неговата разделителна способност (т.е. размерите му - I_{width} и I_{height}) :

$$L_{t_{MAX}} = \frac{(I_{width} . I_{height})-2}{8} \quad (2)$$

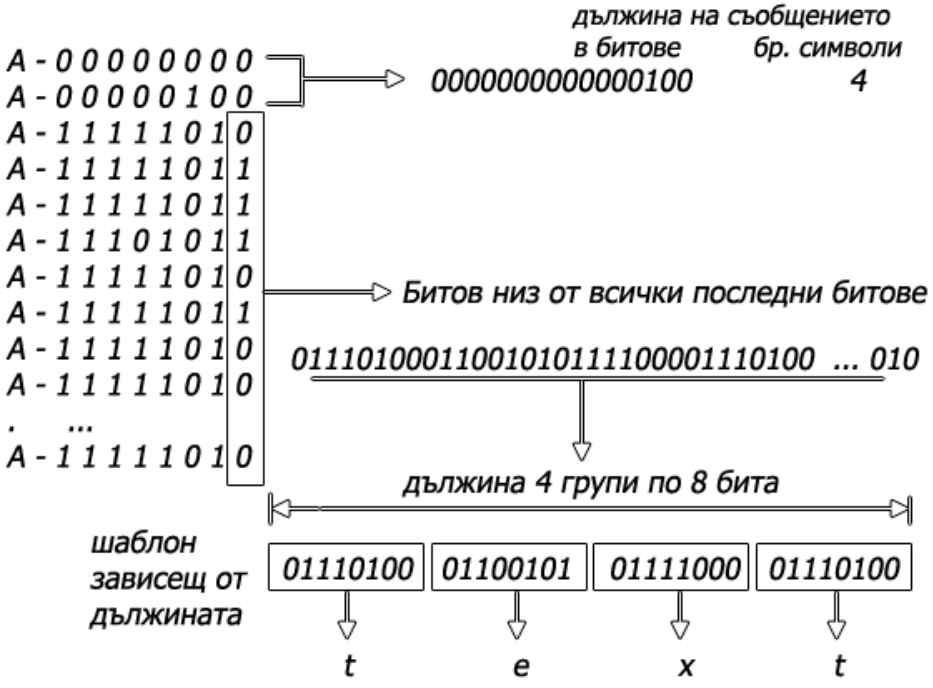
Така например, при изображение с размери 1024x768 пиксела, може да се побере съобщение с максимална дължина от 98303 символа на текста. В модела на постановката при достигане на последния бит от низа на съобщението и при наличието на още свободни за използване битове в изображението, се пристъпва към замяна на последните им битове със случайни такива. Целта е винаги изображението да търпи еднаква обща промяна, независимо от дължината на съобщението и при последващото извличане на битов низ от изображението, той да бъде винаги с еднаква дължина при използването на едно и също изображение.

2.3 Процес на извличане на съобщението от изображението

Процесът на извличане на информацията от изображението се състои от следните стъпки:

1. Разбиване на модифицираното изображение на матрица от пиксели и акцентиране върху битовете от компонента A;
2. Извличане на битовете за компонента A на първите два пиксела и декодирането на дължината на съобщението от двоичен вид;
3. Извличане на всички последни битове на компонента A на всички останали пиксели и зареждането им в единен низ от битове;
4. Извличане на истинското съобщение от общия низ от битове посредством шаблон, зависещ от декодираната дължина на съобщението.

На фигура 4 е изобразена схемата на извличане на оригиналното съобщение от изображението.



Фиг.4. Процес на извличане на съобщението

Ключова роля в модела има приложеният шаблон, без който е невъзможно извличането на коректното съобщение. Той зависи от дължината на съобщението и е с размерност от 8 бита на всяка клетка.

2.4 Експериментални резултати

Моделът е реализиран в програмната среда на C#. Реализираното приложение за опитната постановка е инсталирано и тествано върху следната компютърна конфигурация: Processor Intel Xeon E5450 3.0 GHz, 8 GB RAM, Motherboard Gigabyte EP43-DS3LR (Chipset: Intel P43; South bridge: Intel 82801JR /ICH10R/; LPCIO: ITE IT8718), Windows 7 Ultimate Service Pack 1 64-bit (x64), .NET Framework 4.7.2. Работата на приложението протича нормално, като по време на работата си заема около 15 МБ в режим на готовност и достига до около 149 МБ от цялото пространство на RAM паметта. В един единствен случай приложението не приключи своята работа: при работа с изображение с големина 24,7 МБ и разделителна способност от 6048 x 4032, поради недостиг на системна памет на системата, върху която се изпълнява, като преди да преустанови работа си, потреблението му на системна памет достигна пикова нужда от 3 ГБ.

Проведен е опит с 50 изображения, различаващи се както по големина на файла, така и по различната разделителна способност. Във всяко едно от тях е вградено едно и също съобщение с дължина от 1024 символа. В Таблица 1 са представени обобщени данни на това изследване, като в нея са изобразени представителите на различни групи изображения.

Таблица 1. Резултати от работата на системата с различни изображения

изображение		Конвертиране на съобщение ms	време за вмъкване, ms	стего-изображение		Извлечено съобщение	време за извличане, ms
размер	разд. способност			размер	разд. способност		
858 КБ	1024x768	22,1838	5,0949	116 КБ	1024x768	да	16,999
826 КБ	1024x768	25,1496	4,9747	120 КБ	1024x768	да	15,4362
548 КВ	1024x768	23,3738	5,385	90 КБ	1024x768	да	16,7695
112 КБ	800x600	38,5494	4,8829	59 КБ	800x600	да	21,3598
81 КБ	800x600	36,2246	4,8436	67 КБ	800x600	да	20,7895
1,67 МБ	1920x1200	44,2086	5,0345	321 КБ	1920x1200	да	26,8954
1,47 МБ	3840x2160	40,5388	5,5032	423 КБ	3840x2160	да	26,2369
6,04 МБ	4503x3227	1844,7407	6,419	788 КБ	4503x3227	да	1235,895
24,7 МБ	6048x4032	неуспял	неуспял		6048x4032	не	неуспял

3. Заключение

От направените опити с различни изображения може да се заключи, че системата има почти 100% извличане на вградените съобщения. Стего-изображенията или обработените изображения са с относително малки размери и са подходящи за използване при транспортирането на съобщенията в комуникационна среда. Времената, необходими за обработка, са в нормални граници, но те са и в силна зависимост от характеристиките на хардуерната хостваща система. Видно е, че времето за извличане е по-малко от времето за вграждане на съобщението, макар че двете операции са по-скоро реверсивни и се предполага, че времето им на изпълнение трябва да клони към равенство. Причината за това е използването на шаблона, зависещ от дължината на съобщението, който се използва при

извличането, като на практика всичко извън него се игнорира. Определено големите файлове не са подходящи за използване в такава система, но тези с размер до 1 Мб и разделителна способност от 1024 x 768, 800 x 600 и др. подобни, са изключително подходящи и като време, необходимо за извършване на необходимите действия, и като капацитет за побиране на чисто текстово съобщение.

Литература

[1]. Гребенников, В.В., Стеганография. История тайнописи, Самиздат (ЛитРес), 2019г., стр.66;
[2]. Грибунин, В.Г., И.Н. Оков, И.В. Туринцев, Цифровая стеганография, СОЛОН-ПРЕСС, Москва, 2009г., стр. 265;
[3]. Fridrich, J., Steganography in digital media, Cambridge University Press, Cambridge, 2010, pp. 437;

За контакти:
инж. Димитър Г. Тодоров
катедра „Компютърни науки и технологии“
Технически университет – гр.Варна
E-mail: dimgeto@abv.bg
dimgeto@gmail.com

СИСТЕМА РАСПОЗНАВАНИЯ ЛИЦ НА ОСНОВЕ CPU, GPU

Олександра Александрова

Abstract: Рассмотрены наиболее распространенные подходы к распознаванию лиц на базе нейронных сетей. Отдельные этапы процесса распознавания были реализованы на CPU и GPU для определения условий повышения производительности. В результате найден оптимальный вариант запуска этапов системы распознавания на основе нейронной сети. Выявлена необходимость запуска первого и третьего этапа системы распознавания на центральном процессоре, а второго на графическом.
Keywords: Система распознавания, CPU, GPU, количество кадров в секунду, вероятность распознавания.

1. Введение

Для реализации взаимодействия машины и лица человека необходима система распознавания лиц. На сегодняшний день разработано много методов выявления человеческой головы, отслеживание ее угла и положения [1] - [4]. Они не описывают детали формы лица, поскольку предполагают, что отслеживаемых лицо «жесткое», например, прямоугольник или эллипс. Для преодоления этого слабого места были разработаны системы распознавания лица, устойчивы к различным поз.

Существует много методов и подходов к распознаванию человеческого лица для различных задач, включая распознавание лица в видеопотоке. Также для распознавания лица используется методы по созданию трехмерного лица для повышения качества распознавания. [5]

Широкое внедрение систем видеонаблюдения привело к появлению спроса на системы распознавания и идентификации в видеопотоке. Такие системы базируются на технологиях, которые используют методы компьютерного зрения для автоматического получения информации на основе анализа последовательности изображений, полученных от видеокамер в режиме реального времени. Иными словами, эта технология использует методы распознавания и обработки изображений в результате анализа видеопотока [6]. Современные системы распознавания и идентификации с помощью видеозаписи способны обнаруживать и отслеживать в реальном времени установленные цели (например, автомобиль, группу людей) или потенциально опасные ситуации (например, дым, огонь, несанкционированный доступ) без вмешательства человека, а затем немедленно подать сигнал тревоги. Еще одним преимуществом использования таких систем является значительное уменьшение нагрузки на каналы связи и архивную базу путем фильтрации видеопотока в реальном времени. Видеокамера передает видеопоток на сервер в режиме реального времени, система распознавания и идентификации определяет соответствие информации, хранящейся в базе данных, а идентификация учитывает факторы, заранее определены в системе (например, усы или маски). [6] Важным является определить при каких условиях нейронная сеть для распознавания лиц показывает лучшие результаты, необходимо оценить влияние запуска этапов системы распознавания на центральном и графическом процессорах.

2. Эксперименты на CPU и GPU

Для оценки производительности используется система повторного распознавания Interactive Face Recognition [7]. Она представляет основную архитектуру построения конвейеров этапов, которые поддерживают макет нескольких устройств и параллельное или последовательное выполнение с помощью библиотеки OpenVINO в Python. В частности, эта система использует 3 этапа для построения конвейера, способного распознавать лица в видео, их метки (или "точки ориентира") и распознавания лиц с помощью предоставленной базы данных лица (галерея).[8]

Система считывает входной поток видеовхода кадр за кадром, будь то устройство камеры или видеофайл, и анализирует каждый кадр независимо. Для прогнозирования приложение разворачивает 3 этапа на выбранных устройствах с помощью библиотеки OpenVINO и выполняет их асинхронно. Входной кадр обрабатывается первым этапом распознавания лиц, чтобы предусмотреть ограничительные кадры для граней. Тогда ключевые точки лицо предусматриваются соответствующей вторым этапом. Заключительный шаг в обработке кадра выполняется третьим этапом распознавания лиц, которая использует найденные точки лица, ищет соответствующие лица, найденные в видеокадре, и сопоставляет с теми, что найдены в галерее. Затем результаты обработки визуализируются и отображаются на экране или записываются в выходной файл. [8]

На рисунке 1 продемонстрировано в каком порядке будут тестироваться 3 этапа системы распознавания лиц.

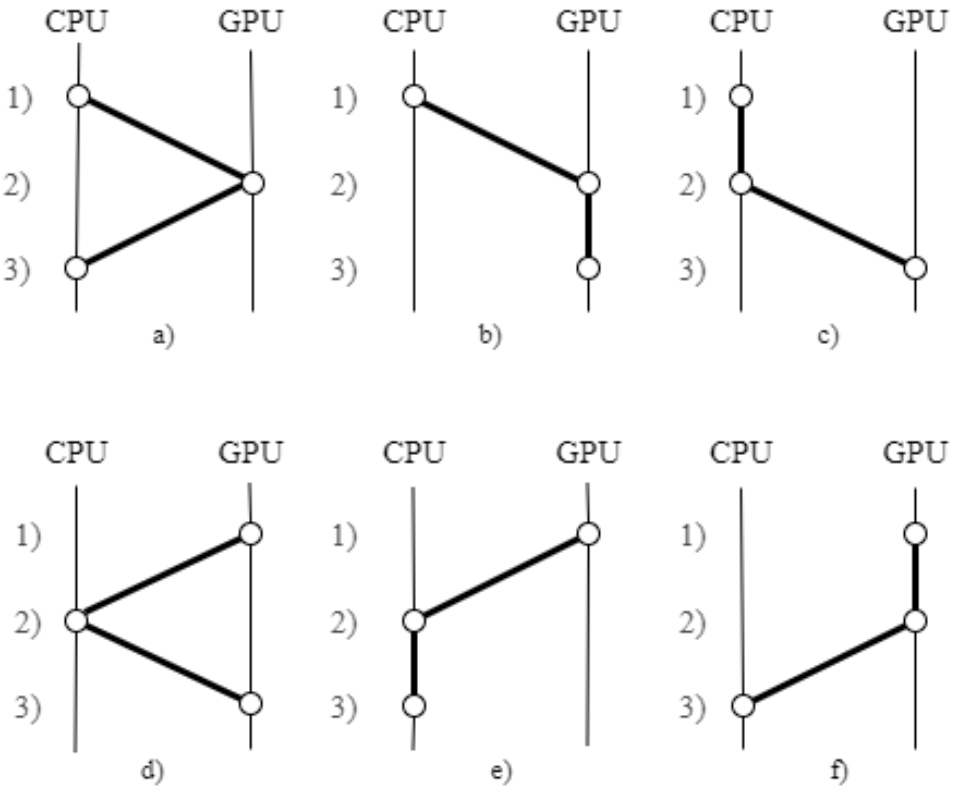


Рис. 1. Распределение этапов системы распознавания на CPU и GPU (1 – face-detection-retail-0004, 2 – landmarks-regression-retail-0009, 3 – face-reidentification-retail-0095)

Для оценки всех вариантов (рис. 1) запуска системы используется вероятность распознавания и количество кадров в секунду. Для экспериментов была использована видеопоследовательность с 342 кадров длительностью 11 секунд.

В первом эксперименте первый этап (face-detection-retail-0004), который находит лицо в кадре и обнаруживает его положение запущен на CPU, второй этап (landmarks-regression-retail-0009), который предусматривает пять ориентиров на лице (два глаза, нос и два уголка губ) запущен на GPU, третий этап системы распознавания (face-reidentification-retail-0095), который создает векторы функций близкие по косинусным расстоянием для подобных лиц и дальние для различных лиц запущен на CPU.

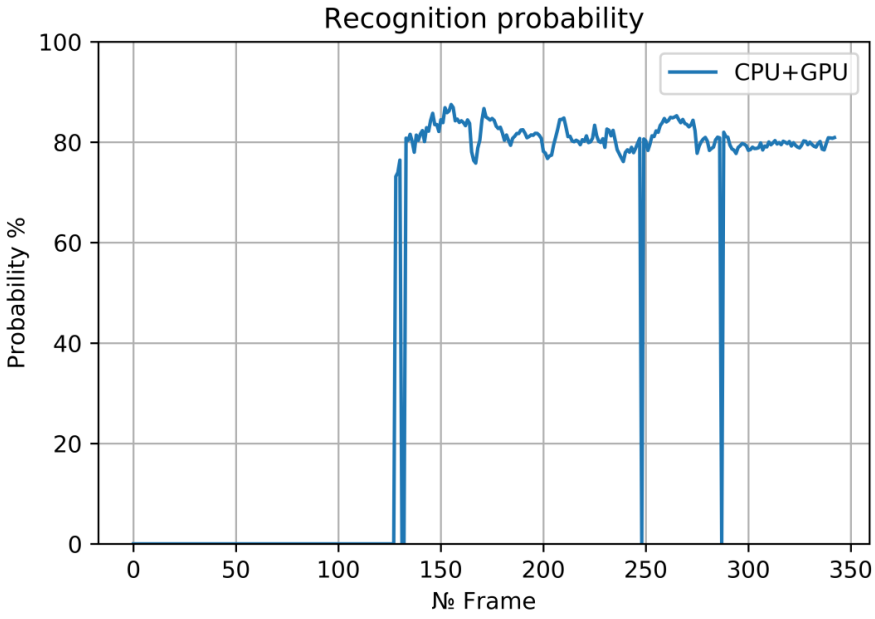


Рис. 2. Тестирование варианта A (CPU-GPU-CPU). Вероятность распознавания

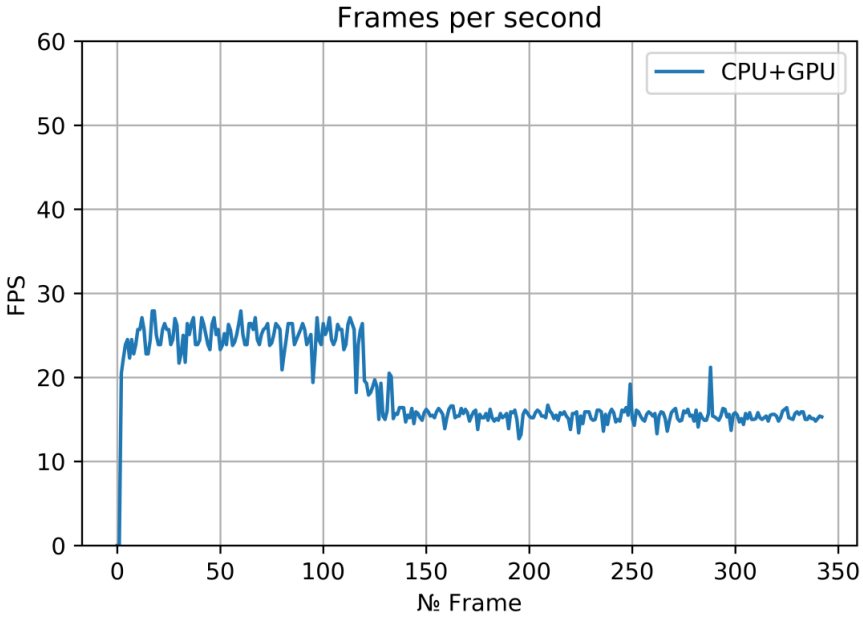


Рис. 3. Тестирование варианта A. Количество кадров в секунду

На рисунке 2 продемонстрирована вероятность распознавания, а на рисунке 3 количество кадров в секунду. На видео человек идет издалека прямо на камеру. Поскольку распознавания лица происходит после 120 кадра, этап повторной идентификации запускается в это же время.

Для второго эксперимента первый этап системы распознавания запущена на CPU, второй на GPU, третий также на GPU. Как видно на рисунке 4 вероятность распознавания по сравнению с предыдущим экспериментом имеет показатели хуже. Количество неудавшихся попыток распознать лицо человека составляет 20, что в 2 раза больше, чем в предыдущем эксперименте. Количество кадров в секунду (рис. 5) имеет больший разлет от 11 до 20 (после 120 кадра, когда непосредственно найдено лицо для распознавания).

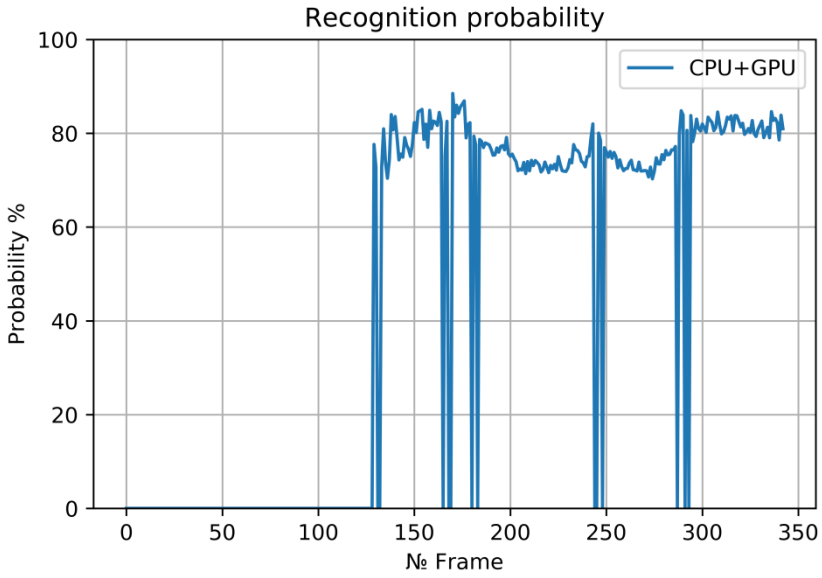


Рис. 4. Тестирование варианта B (CPU-GPU-GPU). Вероятность распознавания

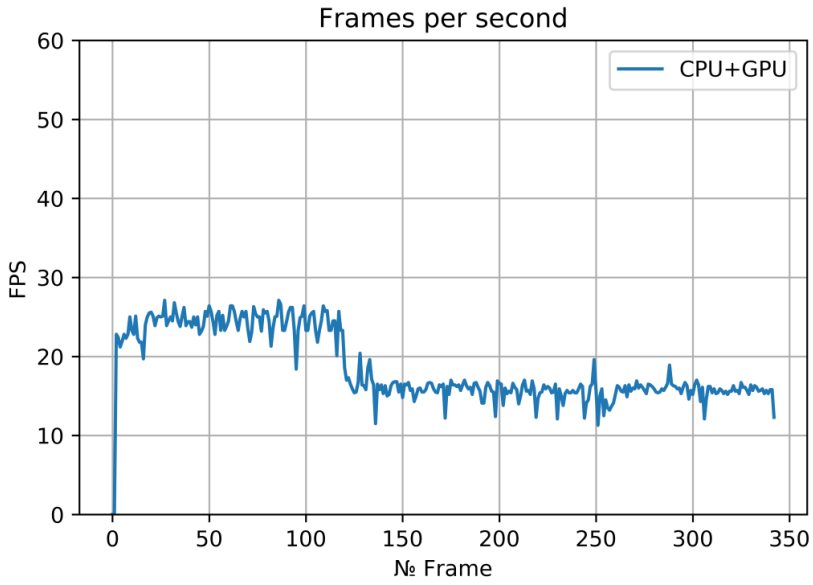


Рис. 5. Тестирование варианта B. Количество кадров в секунду

Из третьего эксперимента видно, что количество неудавшихся попыток распознать лицо достигает 56 (рис. 6). Количество кадров в секунду (рис. 7) практически без особых изменений.

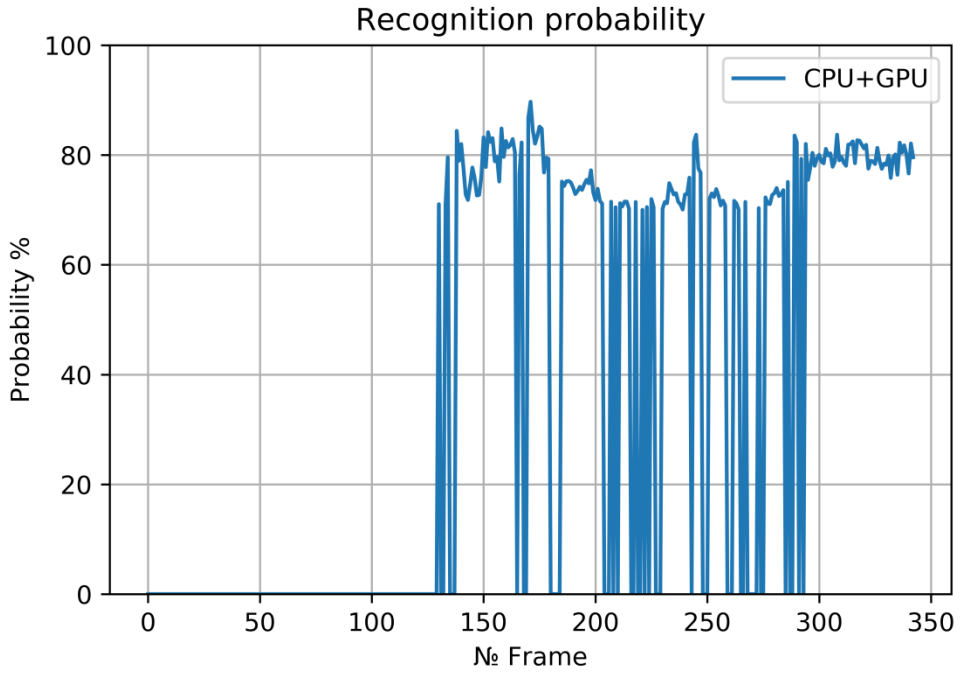


Рис. 6. Тестирование варианта C (CPU-CPU-GPU). Вероятность распознавания

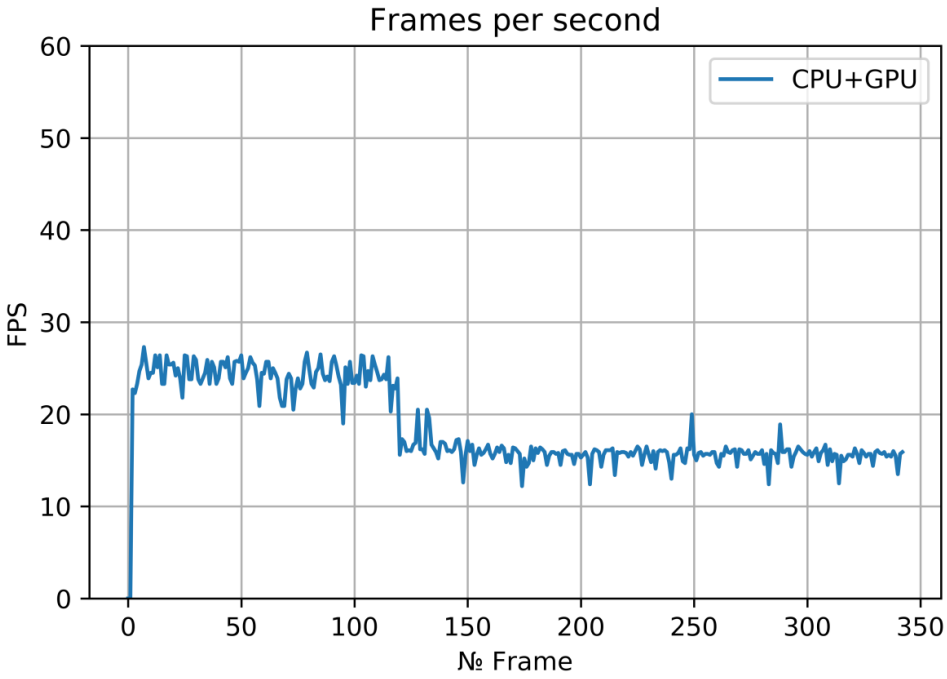


Рис. 7. Тестирование варианта C. Количество кадров в секунду

Большое количество неудавшихся попыток распознать лицо и в целом ниже процент по распознаванию при определении лица вышел в четвертом эксперименте, когда первый и третий этапы запущены на GPU, а второй на CPU. Всего 68 неудавшихся попыток (рис. 8).

На рисунке 9 видно, что количество кадров в секунду в среднем ниже, чем в предыдущем эксперименте.

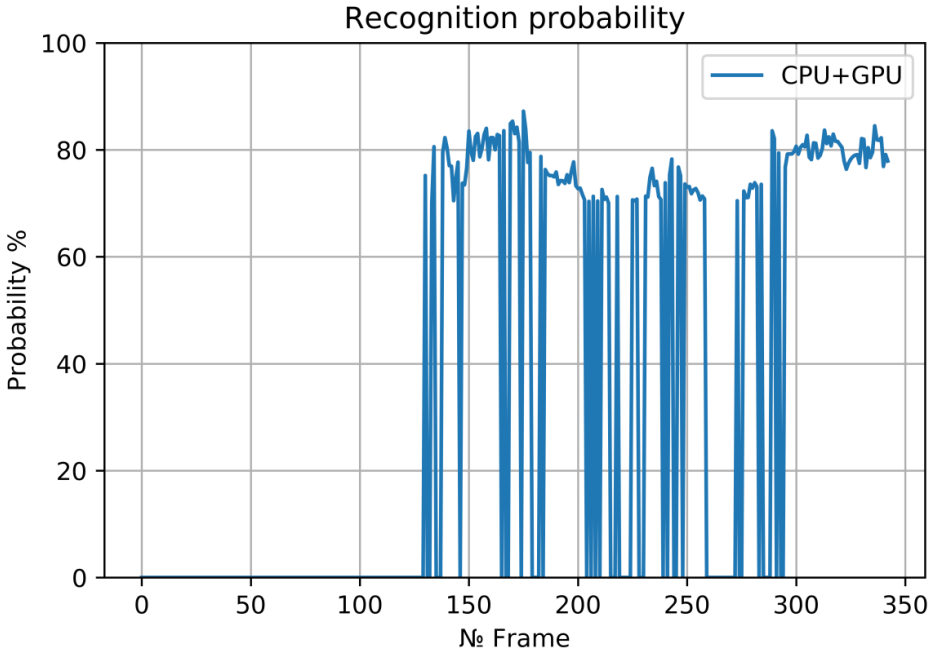


Рис. 8. Тестирование варианта D (GPU-CPU-GPU). Вероятность распознавания

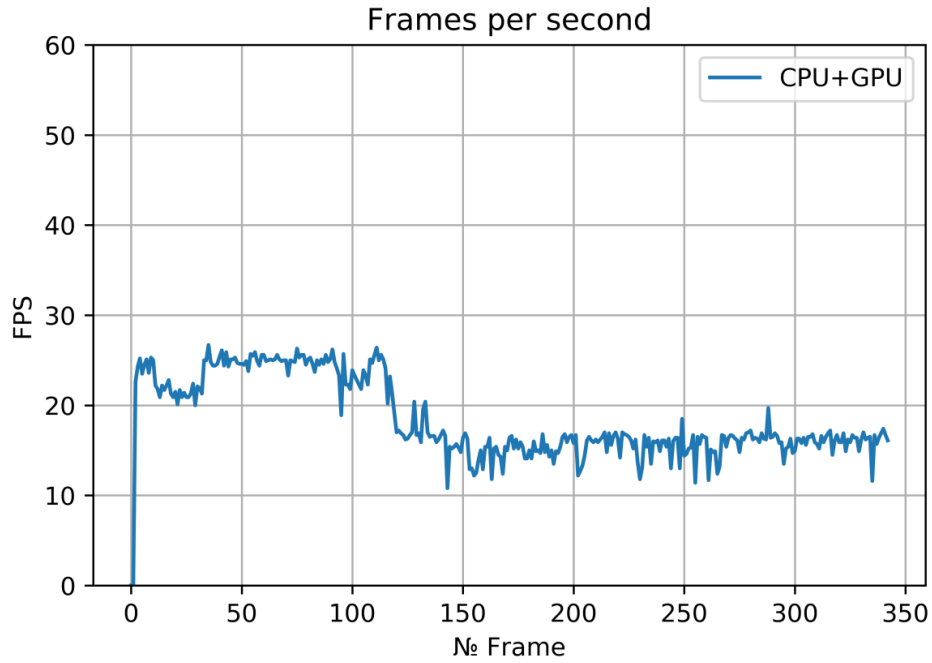


Рис. 9. Тестирование варианта D. Количество кадров в секунду

В пятом эксперименте первый этап запущен на GPU, второй и третий на CPU. Количество неудавшихся попыток всего составляет 10 (рис. 10), как и в первом эксперименте (где первый и третий этап запущены на CPU, а второй на GPU). Количество кадров в секунду (рис. 11) критически не отличается.

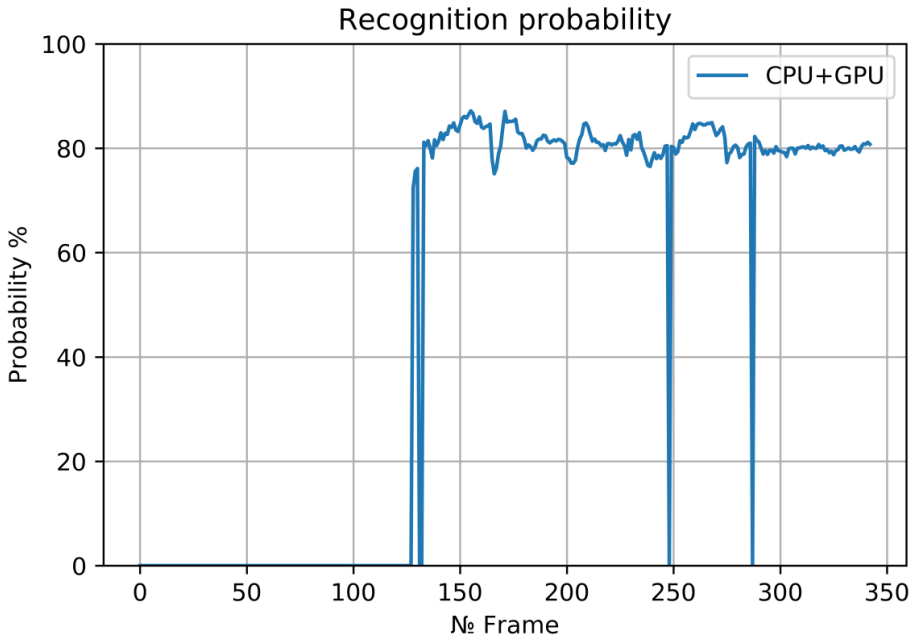


Рис. 10. Тестирование варианта E (GPU-CPU-CPU). Вероятность распознавания

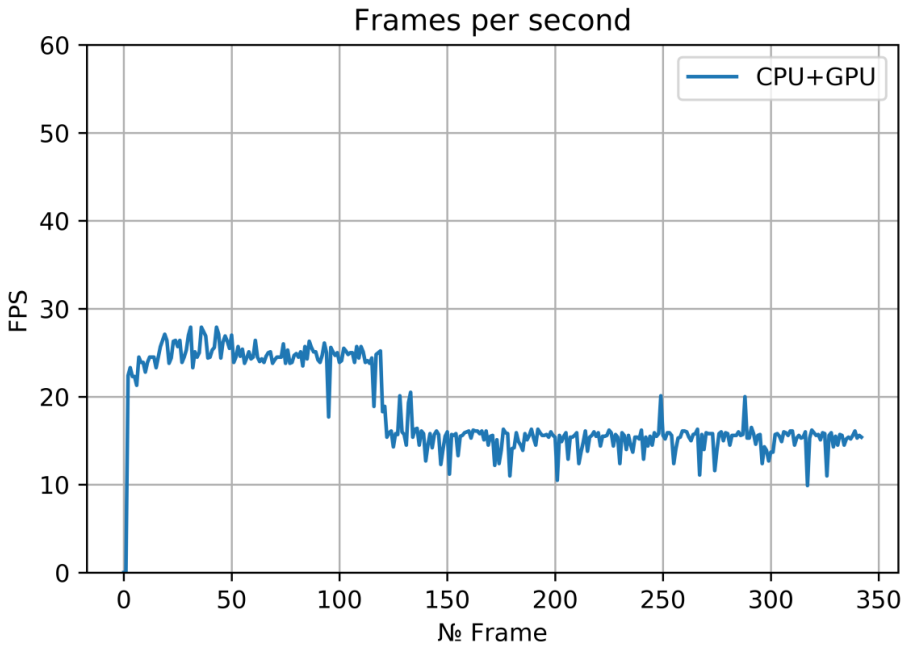


Рис. 11. Тестирование варианта E. Количество кадров в секунду

В шестом эксперименте первый и второй этапы запущены на GPU, а третий на CPU. Количество неудавшихся попыток распознать лицо составляет 10 (рис. 12), как и в первом и пятом экспериментах. Количество кадров в секунду (рис.13) идентична предыдущему эксперименту.

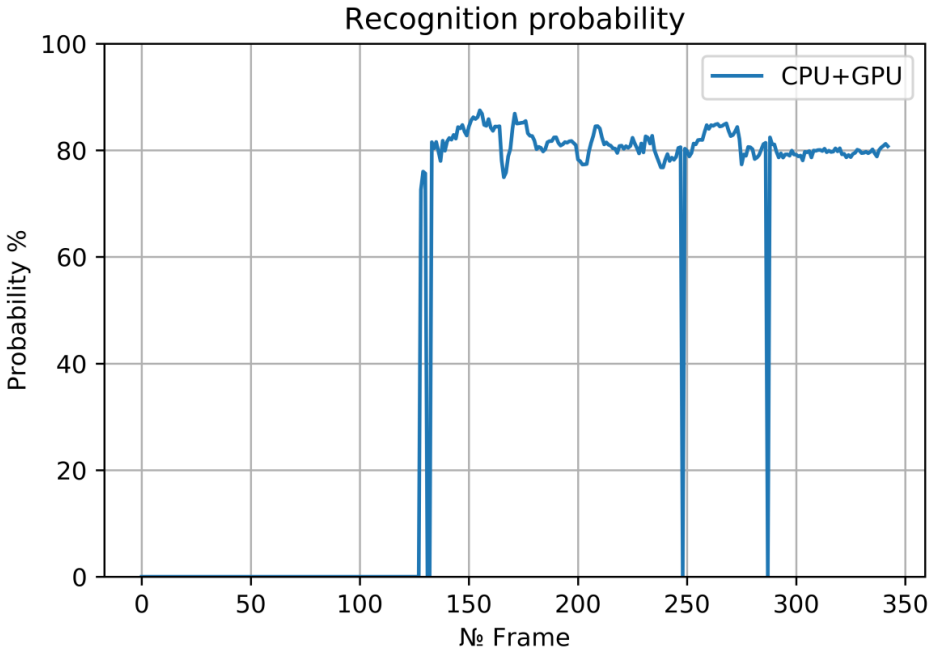


Рис. 12. Тестирование варианта F (GPU-GPU-CPU). Вероятность распознавания

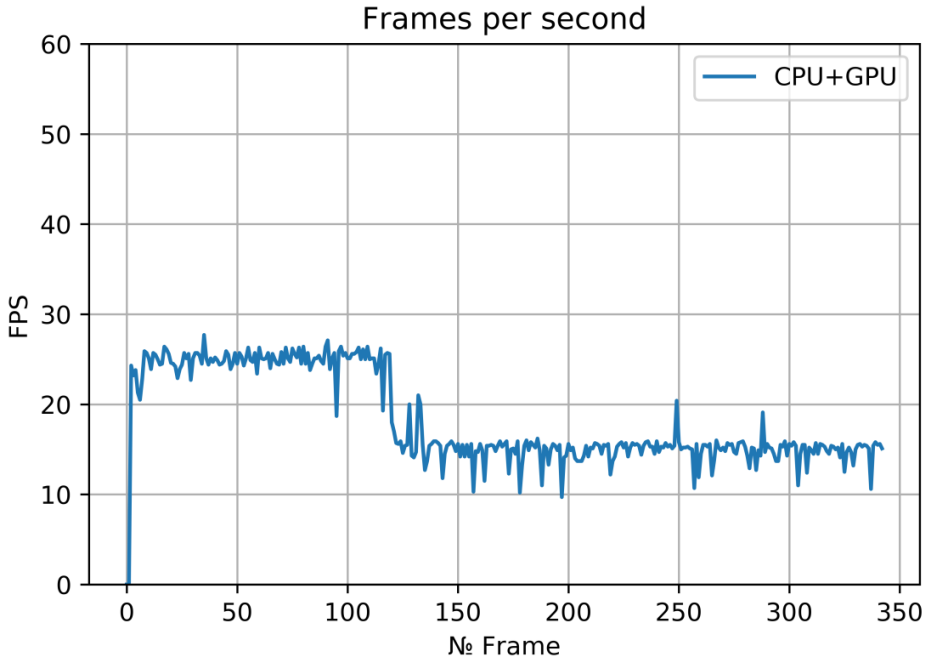


Рис. 13. Тестирование варианта F. Количество кадров в секунду

3. Выводы

Производительность работы системы распознавания лиц на основе нейронной сети была оценена на центральном и графическом процессоре. Нейронная сеть была протестирована на видео продолжительностью 11 секунд и 342 кадрах. Проанализированы количество кадров в секунду и вероятность распознавания лица. Лучший результат был получен при одновременном запуске первого этапа (face-detection-retail-0004), который находит лицо в

кадре и обнаруживает его положение на CPU, второго этапа (landmarks-regression-retail-0009), который предусматривает пять ориентиров на лице (два глаза, нос и два уголка губ) на GPU, третьего этапа (face-reidentification-retail-0095), который создает векторы функций близкие по косинусным расстоянием для подобных лиц и дальние для различных лиц на CPU. В данном случае обнаружено всего 10 неудачных попыток распознавания, стабильное количество кадров в секунду на протяжении распознавания.

Литература

[1]. Liu C. A Bayesian discriminating features method for face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, Vol. 25, 6., pp. 725-740.

[2]. Viola P. Rapid Object Detection using a Boosted Cascade of Simple Features. Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2001, Vol. 1, pp. 511-518.

[3]. Schneiderman H., Kanade T. Object Detection Using the Statistics of Parts. International Journal of Computer Vision, 2002, Vol. 56, 3, pp. 151-177.

[4]. Berg T. L., C. Berg, J. Edwards, M. Maire, R. White, et al. Names and Faces in the News. Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2004, Vol. 2, pp. 848-854.

[5]. O. Aleksandrova and Y. Bashkov, "3D face model reconstructing from its 2D images using neural networks," 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine, 2019, pp. 98-101, doi: 10.1109/ATIT49449.2019.9030503.

[6]. Le Cun Y., Bengio Y. Convolutional networks for images, speech and time series. Handbook Brain Theory Neural Networks, 1998, Vol. 7, 1, pp. 255–258.

[7]. Blanz V., Vetter T. Face Recognition Based on Fitting a 3D Morphable Model. Transactions on Pattern Analysis and Machine Intelligence, 2003, Vol. 25, 9, pp. 1063-1074.

[8]. Interactive Face Recognition Documentation [электронный ресурс] // OpenVino Documentation – Режим доступа: https://docs.openvinotoolkit.org/2020.1/_demos_python_demos_face_recognition_demo_README.html

For contacts:
Graduate student, Oleksandra, Aleksandrova
Department of Applied Mathematics and Informatics
Donetsk National Technical University
E-mail: oleksandra.aleksandrova@donntu.edu.ua

ПРОГРАМЕН СИМУЛАТОР ЗА ИЗСЛЕДВАНЕ НА СТРУКТУРИ ОТ ДАННИ

Антоанета И. Иванова, Александър А. Николов

Резюме: Идеята на настоящата статия е да представи и опише основните функционални характеристики на програмен симулатор на сложни структури от данни (в частност кубове данни). Разработеният симулатор дава възможност за изследване на компонентите на данновата структура, нейното ефективно проектиране и изграждане. Структурата и данните в нея се представят с помощта на свързан ориентиран граф, където се цели сортиране на данните с цел увеличаване на вътрешните връзки между елементите на подграфите. Данните и връзките между тях се представят чрез матрица на съседство.
Ключови думи: структури от данни, граф, силно свързан подграф, последователни алгоритми с графи, куб данни, програмен симулатор

Program Simulator for Data Structures Research

Antoaneta I. Ivanova, Aleksander A. Nikolov

Abstract: The idea of this article is to present and describe the main functional characteristics of a software simulator of data structures (in particular data cubes). The simulator provides an opportunity to study the components of the data structure, its effective design and construction. The structure and the data in it are presented as a connected oriented graph. The aim is to sort the data and increase the internal connections between the elements of the subgraphs. The data and the relationships between them are represented by an adjacency matrix.
Keywords: data structures, graph, strongly connected subgraph, sequential graph algorithm, data cube, program simulator

1. Въведение

Интензивното развитие на компютърните системи предоставя възможност за увеличаване обема и сложността на съхраняваните и обработвани данни. Проблемите, свързани с тяхното съхранение, структуриране и обработка, изискват решения, които да повишат ефективността и бързодействието при тяхната идентификация, търсене, транзитиране и обработка. Тенденцията е от обикновена файлова система или база от данни да се премине към т.нар. кубове от данни [1], [2].

Процесът на структуриране и съхраняване на информацията в подобна структура изисква познаване на основните ѝ харктеристики и свойства. Стремежът към намаляване времето за отговор при заявки към този тип структури предполага ефективното ѝ изграждане с цел намаляване на разредеността на куба от данни. Предложени са редица методи за ефективно изграждане [3], [4] на подобни структури от данни - чрез откриване на сементични връзки между данните и изключване на такива, водещи до разреденост, чрез разделянето на основния куб на подкубове, като целта е изграждане на по-малки кубове с по-висока плътност.

В основата на този подход е формалната интерпретация на данновата структура [5], като свързан ориентиран граф [6], [7] и идеята е откриване на силно свързани подграфи в основния граф, които ще обособят отделните подкубове.

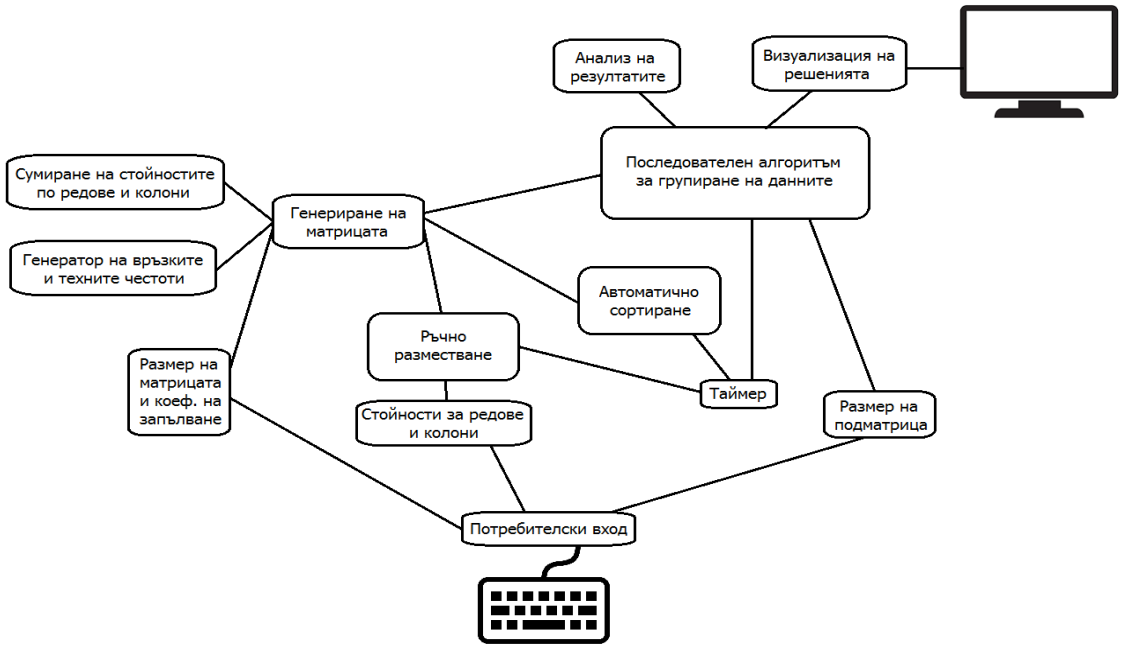
Настоящата разработка е ориентирана към проектирането и реализация на инструментално средство – програмен симулатор [8] на структури от данни, като той се

базира на основни показатели като даннови единици, свързаност, обем, честота на търсене и наложени ограничения при тяхното съхранение. Разработката на програмен симулатор на структури от данни предоставя възможност за изследване на абстрактни структури от данни, както и за тяхното ефективно проектиране, структуриране и реализация. Основната даннова структура, чрез която се представят сложните структури от данни, е свързан ориентиран граф [6] [7], а той е представен посредством матрица на съседство.

2. Проектиране и разработка на програмен симулатор на структура от данни

2.1. Структура и основни функционални характеристики

На фигура 1 е показана основната структура на разработения програмен симулатор, реализиран на C# [8].



Фиг.1. Структура на програмния симулатор

Създаденият симулатор дава възможност за изпълняване на следните действия в описаната по-долу последователност:

1. Въвежда се N - брой на върховете в графа или матрицата на съседство /данните/.
2. Въвежда се коефициент на запълване на матрицата на съседство.
3. Стартира се Генераторът на връзките, който формира матрица с размерност N x N.
4. Прави се избор между ръчно разместване или автоматично сортиране на върховете.
   - При ръчно разместване се въвеждат стойности за избраните редове и колони.
   - При автоматично сортиране приложението сортира данните в низходящ ред.
5. Въвежда се размерът на първата подматрица – m x m.
6. Стартира се алгоритъмът за оптимизация на матрицата, имащ за цел да увеличи вътрешните връзки за сметка на външните.
7. Последователно се обходжат върховете от подматрицата и тези от нейното допълнение.
8. За всяка формирана двойка върхове се проверява увеличаването или намаляването на връзките в подматрицата.
9. При увеличаване на връзките се разменят проверяваните възли (редове и колони в матрицата на съседство) на графа.

10. Данните от алгоритъма се представят в удобен вид под формата на таблица и диаграма.

2.2. Програмни компоненти на симулатора

Програмният симулатор има следните програмни компоненти (функции):

Generator ()

Предназначение: При заданено N се генерира матрица N x N, запълнена с 0 и 1. Степента на запълване зависи от коефициент в диапазона от 0.1 до 0.9.

- Създава се променлива, която ще генерира случайни числа.
- В променлива се съхранява стойността, зададена от потребителя, за коефициент на заетостта на връзките.
- Процедурата обхожда матрицата по всеки ред и стълб с помощта на вложени цикли.
- Зануляват се клетките по главния диагонал при съвпадение на номера на реда и колоната.
- За всяка итерация се генерира число на случаен принцип и то се сравнява с променливата, която съдържа стойността на коефициента. Ако случайното число е по-малко от коефициента, се създава друга променлива, която ще генерира случайно число в диапазона от едно до големината на матрицата.
- Получената стойност се предава на съответната клетка. В противен случай на разглежданата клетка се задава стойност равна на нула.
- Матрицата е запълнена при приключване работата на вложените цикли, които обхождат елементите на матрицата.

На фигура 2 е представен общият вид на генерираната матрица.

	0	1	2	3	4	5	6	7	8	9	10	sumR	sumRC
▶	0	1	2	3	4	5	6	7	8	9	10		
	1	0	1	0	1	1	0	1	0	1	0	5	9
	2	0	0	0	1	1	1	0	0	0	0	3	8
	3	1	0	0	0	0	1	0	1	1	0	4	7
	4	1	1	0	0	0	0	0	0	1	1	4	7
	5	0	1	1	0	0	0	0	0	1	1	4	7
	6	0	0	0	0	0	0	1	0	0	0	1	4
	7	0	0	1	1	0	0	0	0	0	1	3	6
	8	1	1	1	0	0	1	0	0	1	0	5	6
	9	1	1	0	0	0	0	0	0	0	0	2	8
	10	0	0	0	0	1	0	1	0	1	0	3	6
	sumC	4	5	3	3	3	3	3	1	6	3		

Фиг.2. Генерирана матрица в програмния симулатор

SumRow (), SumCol()

Предназначение: Сумиране на връзките от генериран ред/колона.

- Процедурата създава променлива, в която ще се сумират стойности от обхожданите клетки за ред/колона.
- Обхождат се всички редове/колони на матрицата и в началото на всяка итерация сумата се занулява.

- Преминава се през всички клетки на разглеждания ред/колона. Стойността на всяка клетка се прибавя към сумата на реда/колоната.
- Натрупаната сума се записва в предпоследната колона/ред на dataGridView контролата, преди да се премине към следващия ред/колона.

Generating()

Предназначение: Генериране на матрицата и допълнителните данни към нея.

- Процедурата използва стойността, зададена от потребителя, за размерност на матрицата и коефициент на заетост.
- Прави се проверка за коректно въведени данни, размерност (целочислен тип и в границите на интервала на позволените стойности) и коефициент на заетост (число в интервала от 0 до 1).
- При коректно въведени данни се създава dataGridView с получените размери и се заделя място за колоните и редовете, които ще показват сумата, получена от горе-споменатите процедури. Колоните и редовете се именуват.
- Извиква се генераторът, който запълва матрицата със стойности.
- Изчислява се и общата сума за всеки ред и колона и се записва в последната клетка на съответния ред.

ExchangeRow(int k, int r), ExchangeCol(int k, int r)

Предназначение: Размяна на два реда/колони

- Процедурите приемат за вход две целочислени стойности.
- Създава се буферна променлива, която ще помогне за размяната на двете стойности от реда/колоната.

Handmade()

Предназначение: Размяна на два реда и колони, ръчно зададени от потребителя.

- Извличат се стойностите, зададени от потребителя.
- Прави се проверка за некоректни данни (празни стойности или такива, излизащи извън границите на матрицата).
- При правилно въведени числа размяната се извършва с помощта на ExchangeRow() и ExhangeCol(), като се подават необходимите входни параметри.

SortRC()

Предназначение: Автоматично сортиране на данните чрез Bubble Sort алгоритъм в низходящ ред по общата сума от стойностите на редовете и колоните.

Algorithm()

Предназначение: Оптимизиране на данните в матрицата с цел да се покачи броят на вътрешните връзки за сметка на външните в подграфа.

Процедурата има за цел да приложи алгоритъм [6][7], който ще оптимизира матрицата, така че да има възможно най-много вътрешни връзки в близост до главния диагонал.

В тази процедура се извършват оптимизиращи размени.

TableAndChart()

Предназначение: Представяне на данните от алгоритъма в табличен и графичен вид

- Процедурата се задейства при зареждането на втория режим на работа.
- Създава се линейна диаграма.

- Именуват се осите, задават се интервали и се прави цветова легенда за данните. Обхождат се данните и се добавят точките, сочещи стойностите на връзките, към диаграмата.

3. Резултати

Чрез програмния симулатор са извършени и различни експерименти, целящи да разгледат поведението на алгоритъма в различни ситуации.

3.1. Експеримент 1

Квадратна матрица 20x20, разделена на две подматрици 10x10 по главния диагонал. Целта на изследването е да се провери дали алгоритъмът ще увеличи вътрешните връзки за сметка на външните при следните два варианта:

Вариант А - След като се генерира матрицата, се изпълнява алгоритъмът чрез многократно разместването на два реда и стълба.

Вариант Б - Матрицата се сортира в низходящ ред на заетостта на редовете и стълбовете, след което се прилага алгоритъмът. Двата варианта се изпълняват многократно за стойности на коефициента на генериране от 0.1, 0.2 ... 0.9.

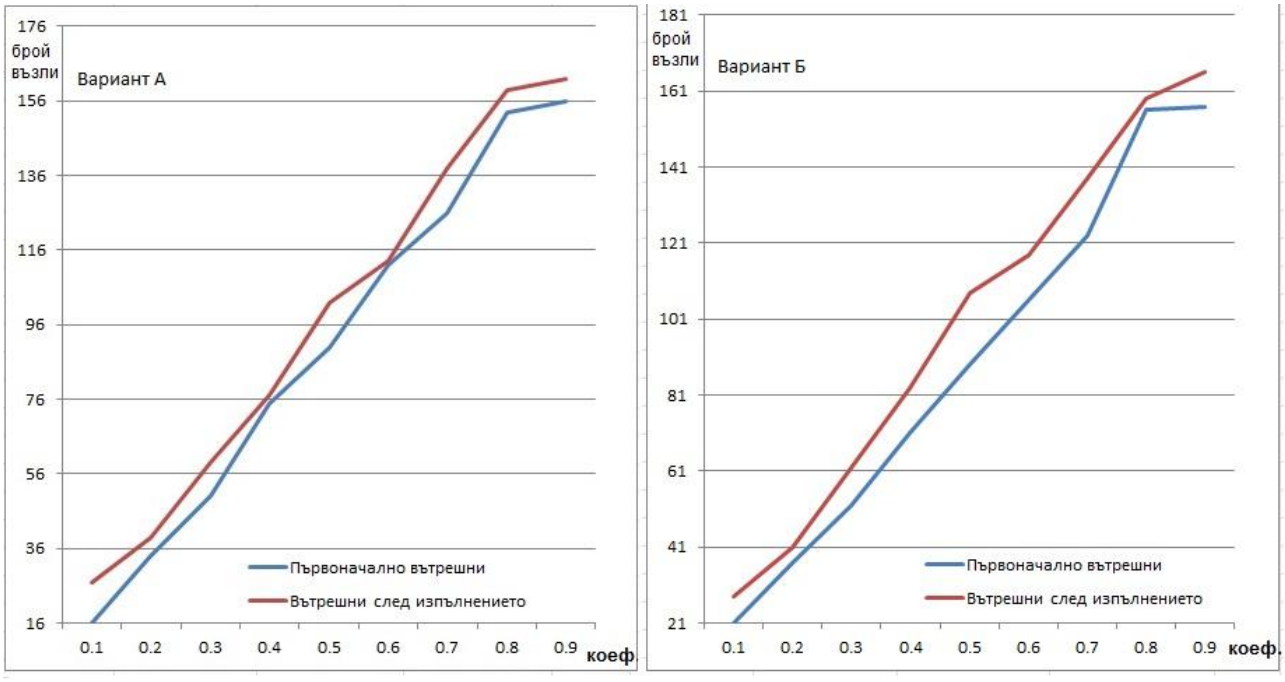
Резултатите от двата варианта от Експеримент 1 са представени в Таблица 1, Таблица 2 и фигура 3.

Таблица 1. Резултати *Вариант А*

Коеф.	**Брой връзки преди подредбата**		**Брой връзки след изпълнението**		**Време**
	вътрешни	външни	вътрешни	външни	**[s]**
0.1	16	18	27	7	00.156
0.2	34	35	39	30	00.358
0.3	50	48	59	39	00.514
0.4	75	69	77	67	00.780
0.5	90	98	102	86	00.390
0.6	112	108	113	107	00.702
0.7	126	135	138	123	00.178
0.8	153	163	159	157	00.468
0.9	156	180	162	174	00.187

Таблица 2. Резултати *Вариант Б*

Брой връзки преди подредбата		**Брой връзки след изпълнението**		**Време**
Вътрешни	Външни	вътрешни	външни	**[s]**
21	13	28	6	00.046
37	32	41	28	00.046
52	46	62	36	00.249
71	73	83	61	00.202
89	99	108	80	00.109
106	114	118	102	00.124
123	138	138	123	00.156
156	160	159	157	00.062
157	179	166	170	00.062



Фиг. 3. Резултати от Експеримент 1, *Вариант А и Вариант Б*

Извод: Резултатите от изследването показват, че и в двата случая алгоритъмът успява да увеличи вътрешните връзки за сметка на външните. При предварително сортирана матрица броят на вътрешните врзъки е дори по-висок, от случаите когато не се извърши съответното сортиране. Друго предимсвто на вариант Б е значителното увеличаване на бързодействието на алгоритъма.

3.2. Експеримент 2

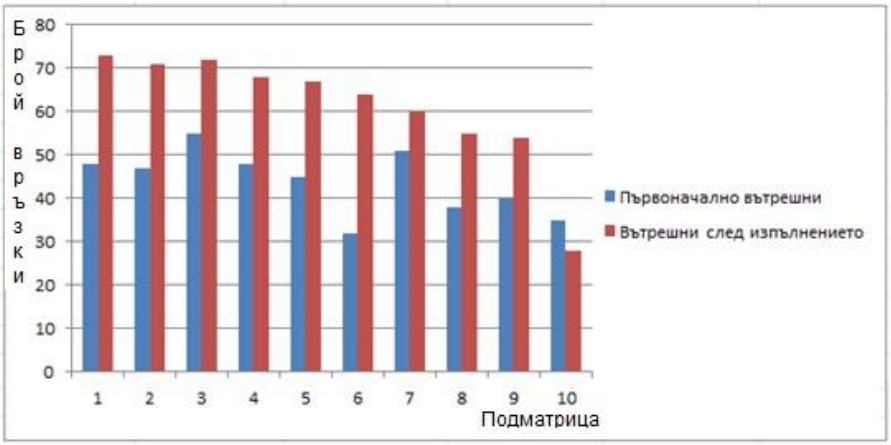
Генерира се квадратна матрица N x N. Въвежда се числото M. Матрицата се сортира. Отделя се първата подматрица с размери m x m, където m = N/M цяло число. Изпълнява се процедурата по целева размяна на редове и стълбове. Следва нова подматрица m x m и т.н. Последната матрица може да бъде с по-малки размери. За експеримента N=100, M=10 при коефициент 0.5 (средно „натоварване“ на структурата).

Целта на изследването е да се наблюдава увеличаването на вътрешните връзки в отделните подматрици.

Резултатите от Експеримент 2 са представени в Таблица 3 и фигура 4.

Таблица 3. Резултати Експеримент 2 – N=100, M=10, коефициент = 0.5

Подматрица	Брой връзки преди подредбата		Брой връзки след изпълнението		Време
	Вътрешни	Външни	Вътрешни	Външни	[s]
1	48	4906	73	4881	7.612
2	47	4859	71	4810	14.071
3	55	4804	72	4738	10.140
4	48	4765	68	4670	12.277
5	45	4711	67	4603	13.946
6	32	4679	64	4539	10.920
7	51	4628	60	4479	12.558
8	38	4590	55	4424	4.180
9	40	4550	54	4320	6.942
10	35	4515	28	4342	0.000



Фиг.4. Резултати от Експеримент 2

Извод: В следствие от експеримента, в общия случай се забелязва осезаемо покачване на вътрешните връзки за отделните подматрици. В дадения пример алгоритъмът е увеличил броя на вътрешните връзки с около 40% спрямо броя на тези, след сортирането в низходящ ред на данните.

3.3. Експеримент 3

Последователно се генерират матрици за стойности на N = 10,20,30 ... 100 за M=10 при коефициент на натоварване 0.5. Целта е да се види как се изменя бързодействието на алгоритъма при нарастване на N.

Резултатите от Експеримент 3 са представени в Таблица 4.

Таблица 4. Резултати Експеримент 3

Размер N	**Брой връзки след изпълнението**		**Време**
	Вътрешни	външни	**[s]**
10	42	0	**0.000**
20	107	95	0.234
30	157	242	0.577
40	227	563	3.276
50	286	971	5.475
60	363	1437	14.102
70	425	2015	24.100
80	506	2706	45.364
90	556	3389	65.847
100	612	4342	92.664

Извод: С увеличаването на размера на матрицата се наблюдава и покачване на необходимото на алгоритъма време за изпълнение. При значително по-големи матрици алгоритъмът разглежда много повече възможни решения и това се отразява върху технологичното му време.

4. Заключение

Представеният симулатор включва в себе си последователен алгоритъм за увеличаване на вътрешни връзки в подграфи на основен граф. Тази разработка може да бъде успешно използвана за ефективно конструиране и създаване на по-плътни подкубове от данни. Тя би имала особено приложение при оптимизиране на кубове от данни и по-скоро при процеса за намаляване на тяхната разреденост. Ако данните от всяко измерение се подредят

посредством симулатора, ще се получи определено по-плътна част от структура, което ще доведе до по-ефективна работа със структурата от данни.

Програмният симулатор ефикасно увеличава броя на вътрешните връзки по главния диагонал и представя данните на потребителя в удобен формат.

Софтуерният продукт, реализиращ програмен симулатор на структури от данни, реализиран на C# [8], предоставя възможност за изследване на абстрактни структури от данни, в това число и на кубове от данни, като се дава възможност както за тяхното изследване, така и за ефективното им проектиране, структуриране и реализация.

5. Идеи за бъдещо развитие

В настоящия проект могат да бъдат направени различни подобрения в няколко насоки. Основните направленията, в които програмният симулатор може да се подобри, са следните:

- Импортиране на данни от външни източници - части от реални бази от данни;
- Разработване на паралелен алгоритъм за сортиране на данните;
- Разработване на смесен алгоритъм (между последователния и паралелния).

Благодарности

Специални благодарности към доц. д-р инж. Митко Митев – дългогодишен преподавател в катедра „Компютърни науки и технологии“ на Технически университет - Варна, под чието ръководство и насоки беше създаден софтуерният продукт, реализиращ настоящия симулатор на структури от данни.

Литература

[1]. Lewis P., Database and Transaction Processing: An Application – Oriented Approach, Addison Wesley, New York, 2003

[2]. Molina H., J. Ullman, J. Windom, DataBase Systems: The Complete Book, Prentice-Hall, 2002, 1119p.

[3]. Naydenova I., Regular Sparsity Map, Proceedings of the 4th International Conference On Information Systems &Grid Technologies, ISBN 978-954-07-3168-1, Bulgaria, 2010, pp.51-63

[4]. Kaloyanova K., I. Naydenova.,Regular Sparsity in OLAP system, Emerging Themes in Information Systems and Organization Studies, Springer, ISBN 978-3-7908-2738-5, Verlag Berlin Heidelberg, 2011, pp. 115-126,

[5]. Ivanova A., Approaches for efficient data extraction from data cube structure, International Conference “Automatics and Informatics”, 2020

[6]. Hatt J., Online assessment of graph theory, DOI: 10.13140/RG.2.2.33897.29285, Brunel University London, 2016, 317p.

[7]. https://www.researchgate.net/publication/312041317

[8]. James R. Evans, E.Minieka, Optimization Algorithms for Networks and Graphs, 2nd Edition, New York, 1992

[9]. Наков С., Принципи на програмирането със C#, Фабер, 2018, Велико Търново

За контакти:
ас. инж. Антоанета И.Иванова
катедра „Софтуерни и Интернет Технологии”
Технически университет - Варна
E-mail: antoaneta_ii@tu-varna.bg

ПОДХОД ЗА ИЗГРАЖДАНЕ НА СТРУКТУРИ ОТ ДАННИ

Антоанета И. Иванова

Резюме: Идеята на настоящата статия е да опише подход за представяне на структури от данни с цел тяхното ефективно изграждане. Структурата и данните в нея се представят с помощта на свързан ориентиран граф, където се цели сортиране на данните и характеристиките на обектите с цел обособяване на подграфи със силно свързани върхове. Отделянето на такива подструктури е изключително полезно с цел преподреждане на характеристиките на обектите и данните за тях, като се цели получаване на подредена и по-плътна структура. Това може да реши редица проблеми и недостатъци на различни даннови структури, такива като data cube структурите.
Ключови думи: структури от данни, ориентиран граф, последователни алгоритми с графи, куб данни

Approach to Build Data Structures

Antoaneta I. Ivanova

Abstract: The idea of this article is to describe an approach to presenting data structures to build them effectively. The structure and the data in it are presented with the help of a directed graph, where the aim is to sort the data and the characteristics of the objects to separate subgraphs with strongly connected vertices. The separation of such substructures is extremely useful to rearrange the characteristics of the objects and the data about them, to obtain a sorted and denser structure. This can solve many problems and disadvantages of various data structures, such as data cube structures.
Keywords: data structures, directed subgraph, sequential graph algorithms, data cube

1. Въведение

Процесът на структуриране и съхраняване на информацията, както и оформянето на структура от данни изисква познаване на основните ѝ характеристики и свойства. Ако се говори за изграждане на куб от данни, на първо място стои определянето на основните характеристики на куба, което е свързано с дефинирането на основните метрики и факти в него.

Основен проблем, който се решава след дефинирането и изграждането на подобна структура, е намаляването на разредеността на данните в него и прилагането на редица подобрения за изграждането на структура с по-голяма плътност. Предложени са редица подходи за ефективно изграждане [1][2] на подобни структури от данни - чрез откриване на семантични връзки между данните и изключване на такива, водещи до разреденост, чрез разделянето на основния куб на части или подкубове, като целта е изграждане на по-малки кубове с по-висока плътност. Определено приложените подходи са строго индивидуални в различните ситуации и все още се търсят различни методи и алгоритми за подобни реализации.

Настоящата разработка е ориентирана към представяне на формална интерпретация на данновата структура, дефинирането ѝ като ориентиран граф [3] и прилагането на алгоритъм за подреждане на данните в ориентирания граф, така че да се обособят силно свързани подграфи и техните елементи да се използват за реализация на подкубове.

Основната даннова структура, чрез която да се представят сложните структури от данни или кубовете данни, е свързан ориентиран граф [4][5], представен посредством матрица на съседство. В нея връзките между елементите се описват с единици. Предложеният подход се стреми да размести и подреди редовете и колоните в матрицата на

съседство с цел да се получат повече единици (свързани елементи) над главния диагонал в матрицата на съседство. Това означава, че се получава подграф с повече вътрешни връзки, отколкото външни.

2. Формална интерпретация на data cube структура

2.1. Дефиниция

Формалната постановка на **Data Cube** структура **S** може да бъде предствена като множество от четири множества $\{\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{D}, \boldsymbol{T}\}$, където:

- $\boldsymbol{X} \equiv \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}\}$ (1)

  представлява множеството от наименованията на видовете данни, включени в структурата. Те се определят на етапа на проектиране и/или част от тях възникват при използването на самата структура.

  За целите на формалния подход нека зададеното множество $\boldsymbol{X} \equiv \{\boldsymbol{x_i}\}$, където $\boldsymbol{x_i} \in \boldsymbol{X}$ е определен или тип данни, имащи конкретен физически или приложен характер.

- $\boldsymbol{G} \equiv \{\boldsymbol{g_1}, \boldsymbol{g_2}, \dots, \boldsymbol{g_m}\}$ (2)

  е множеството на установените връзки на етапа на проектиране на структурата или в процеса на нейното използване.

  Например: $(\forall\{\boldsymbol{x}\} \subset \boldsymbol{X}) \rightarrow (\exists \boldsymbol{g} \subset \boldsymbol{G})$ (2.1)

- $\boldsymbol{D} \equiv \{\boldsymbol{d_1}, \boldsymbol{d_2}, \dots, \boldsymbol{d_l}\}$ (3)

  е множеството от стойности на данни, съхранявани в структурата или възникнали като такива в процеса на нейното използване.

  Следователно: $(\forall \boldsymbol{x_i} \in \boldsymbol{X}) \rightarrow \left(\exists\left\{\boldsymbol{d_1^i}, \boldsymbol{d_2^i}, \dots, \boldsymbol{d_k^i}\right\}\right)$ (3.1)

  т.е. за всяка специфицирана данна, като наименование, съществува множество от нейни конкретни стойности.

  Следователно, $(\boldsymbol{x_i} \in \boldsymbol{X}) \rightarrow \boldsymbol{x_i}\{\boldsymbol{d_1}, \boldsymbol{d_2}, \boldsymbol{d_3}, \dots\}$ (3.2)

  или това са конкретни стойности на данната $\boldsymbol{x_i} \in \boldsymbol{X}$, получени по време на изграждането на съществуващата **Data Cube** структура. Тази представа е валидна само при условие, че данните са от типа $\boldsymbol{x_i} \in \boldsymbol{X}$ не зависят или не са свързани с други типове/видове данни.

- $\boldsymbol{T} \equiv \{\boldsymbol{t_1}, \boldsymbol{t_2}, \dots, \boldsymbol{t_r}\}$ (4)

  представлява множество от регистрирани на етапа на проектиране, където

  $(\forall \boldsymbol{t_i} \in \boldsymbol{T})\left(\forall \boldsymbol{t_j} \in \boldsymbol{T}\right)(\boldsymbol{i} < j) \rightarrow \left(\boldsymbol{t_i} < \boldsymbol{t_j}\right)$ (4.1)

На базата на въведената формална постановка могат да бъдат определени следните случаи:

- Нека са зададени $\boldsymbol{x_i} \in \boldsymbol{X}, \boldsymbol{x_j} \in \boldsymbol{X}$ и $\boldsymbol{i} \neq \boldsymbol{j}$, ако е в сила твърдението

  $$\left(\forall \boldsymbol{d^i} \in \boldsymbol{D}(\exists \boldsymbol{d^j} \in \boldsymbol{D})\right) \& \left(\forall \boldsymbol{d^j} \in \boldsymbol{D}(\exists \boldsymbol{d^i} \in \boldsymbol{D})\right), \quad (5)$$

  то от това следва, че данните на $\boldsymbol{x_i} \in \boldsymbol{X}$ и $\boldsymbol{x_j} \in \boldsymbol{X}$, могат да бъдат обединени в структурно отношение, а между $\boldsymbol{x_i} \in \boldsymbol{X}$ и $\boldsymbol{x_j} \in \boldsymbol{X}$, установена свързаност от типа $\boldsymbol{g_k} \in \boldsymbol{G}$.

- Нека множеството $\boldsymbol{D} \equiv \{\boldsymbol{d_i}\}$, представлява всички стойности на данни, съхранявани в **Data Cube** структурата към определен момент от времето. Мощността на множеството $\boldsymbol{D}$ не е постоянна величина, а зависи от времето.

Следователно, ако с $\boldsymbol{T}$ се отбележи множеството от времена, по време на които са настъпили промени в $\boldsymbol{D}$, то това може да се формализира или представи по следния начин:

$$\boldsymbol{T} \equiv \{\boldsymbol{t_i}\} \equiv \{\boldsymbol{t_1}, \boldsymbol{t_2}, \dots, \boldsymbol{t_r}\}, \tag{6}$$

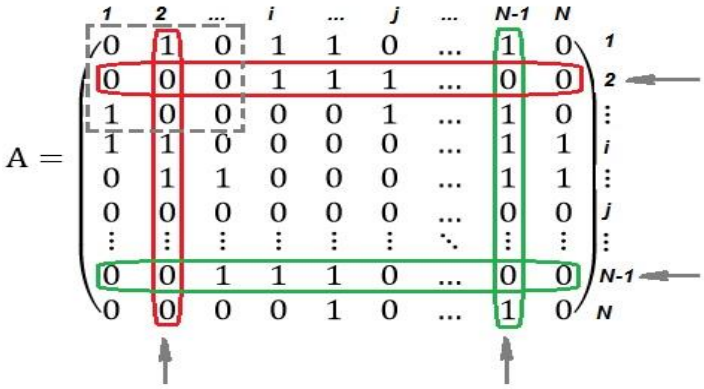
Където $(\forall \boldsymbol{t_i}, \boldsymbol{t_j} \in \boldsymbol{T})(\boldsymbol{i}, \boldsymbol{j}) \rightarrow (\boldsymbol{t_i} < \boldsymbol{t_j})$ (6.1)

и съответно съществуват $\boldsymbol{D}(\boldsymbol{t_0})$; $\boldsymbol{D}(\boldsymbol{t_1})$; ... $\boldsymbol{D}(\boldsymbol{t_i})$; ... $\boldsymbol{D}(\boldsymbol{t_j})$; ... (6.2)

2.2. Алгоритми за откриване на силно свързани подструктури

Така направената по-горе дефиниция на данновата структура дава възможност тя да се разглежда като **ориентиран граф**, който може да се представи чрез матрица на съседство **A**, която ще се подреди, с цел получаване на силно свързани подграфи.

На фигура 1 е представена матрицата на съседство **A**, генерирана в съответствие с описаната дефиниция на данновата структура, като идеята е разместване на възлите в графа, т.е. ред/колона за дадения възел в матрицата на съседство, така че да се получи подграф с по-голям брой вътрешни връзки отколкото външни.



Фиг.1. Основен вид на матрицата A

Реализацията на алгоритъма за откриване на силно свързан подграф в граф предполага преминаване през няколко **етапа**:

- определяне на размера на матрицата **N**, т.е. броя на участващите елементи в графа;
- определяне на връзките/дъгите между отделните елементи. Попълване на матрицата на съседство **A**. Тя може да се запълни:
  - с генератор на случайни числа, в зависимост от въведен коефицент на запълване, който ще определи честотата на връзките между отделните елементи в графа;
  - с данни от реална база от данни на съществуващите зависимости на отделните стойности на метриките, определени в data cube структурата;
- разделяне на матрицата **A** на подматрици с размерност **M** с цел обхождане последователно на техните елементи;
- пресмятане на брой изходящи и входящи дъги за всеки възел и сравняването на техните стойности с тези на следващите възли на другите подматрици, обходени последователни;
- при наличие на повече входящи и изходящи връзки разменяне на съответните редове и колони в матрицата на съседство. Така към първия възел в първата подматрица се добавя нов възел, който е максимално инцидентен с него. Като резултат се постига преподреждане на елементите с цел получаване на повече връзки между елементите в подматрицата за сметка на връзките с останалите възли.

В зависимост от начина на обхождане на елементите в матрицата на съседство **A** можем да определим три вида алгоритми - последователни, паралелно-последователни и паралелни.

Последователни алгоритми са тези, при които се започва с една данна, т.е. един възел и последователно към нея/него се добавят останалите данни/възли.

Паралелно-поледователни алгорими са тези, при които се започва с първоначална група на данни/възли и в последствие тази група се модифицира на базата на определени критерии.

Паралелни алгорими са тези, при които матрицата се разделя на подматрици, съгласно ограничително число. Обработката се осъществява между две подматрици.

Описаната по-горе последователност от действия за подреждане на матрицата на съседство **A** е пример за последователен алгоритъм.

3. Резултати

На база на описания подход е създаден програмен симулатор, с чиято помощ са извършени експерименти, доказващи ефективното увеличаване на вътрешните връзки в подграфите след прилагане на подхода за сметка на външните такива.

Извършени са **три групи** експерименти, които доказват това, фиксирайки и изменяйки различни входни характеристики:

- за матрица на съседство **A** за 20 елемента, разделена на две подматрици с размер 10x10 се изменя коефицентът на плътност на запълване на матрицата от 0.1 до 0.9, като се прилага алгоритъмът в два варианта – без и с предварително сортиране на броя на връзките по редове и колони.
- за матрица на съседство **A** за 100 елемента, разделена на подматрици с размерност 10x10 и фиксиран коефицент на плътност 0.5. Идеята е да се покаже промяната на брой вътрешни и външни връзки за отделните подматрици, в следствие на прилагане на алгоритъма.
- за матрици на съседство с различни размерности **N** между 10 и 100, с фиксиран размер на подматрицата 10 и коефициент на плътност 0.5. Целта е да се покаже как се изменя бързодействието на алгоритъма при нарастване на **N.**

4. Заключение

Представена е графо-аналитична интерпретация на **data cube** структура и последователен алгоритъм за увеличаване на вътрешни връзки в подграфи на основен граф. Тази разработка може да бъде успешно използвана при конструиране и създаване на **data cube** структури, намалявайки ефекта на разредеността на куба, разделяйки успешно един куб на подкубове с по-голяма плътност. Ако данните от всяко измерение на куба се подредят, следвайки описания подход, ще се получи определено по-плътна част от структурата, което ще доведе до по-ефективна обработка на данните му.

Идеята за бъдеща работа се състои в това експериментите да се осъществят върху структури, попълени на базата на реално работещи бази от данни, в частност база данни „Раков Регистър“.

Литература

[1]. Naydenova I., Regular Sparsity Map, Proceedings of the 4th International Conference On Information Systems &Grid Technologies, ISBN 978-954-07-3168-1, Bulgaria, 2010, pp.51-63

[2]. Kaloyanova K., I. Naydenova.,Regular Sparsity in OLAP system, Emerging Themes in Information Systems and Organization Studies, Springer, ISBN 978-3-7908-2738-5, Verlag Berlin Heidelberg, 2011, pp. 115-126,
[3]. Ivanova A., Approaches for efficient data extraction from data cube structure, International Conference "Automatics and Informatics", 2020
[4]. Hatt J., Online assessment of graph theory, DOI: 10.13140/RG.2.2.33897.29285, Brunel University London, 2016, 317p.
[5]. James R. Evans, E.Minieka, Optimization Algorithms for Networks and Graphs, 2nd Edition, New York, 1992

За контакти:
ас. инж. Антоанета И. Иванова
катедра „Софтуерни и Интернет Технологии"
Технически Университет - Варна
E-mail: antoaneta_ii@tu-varna.bg

ИЗСЛЕДВАНЕ НА БЕЗЖИЧНИ ТЕХНОЛОГИИ ЗА INTERNET OF THINGS (IOT) МРЕЖИ

Айдън М. Хъкъ

Резюме: Широкото разпространение и активното развитие на съвременните комуникационни технологии като 4G и 5G предоставят основа за повсеместното разрастване и разпространение на Internet of Things (IoT) технологиите. Тази статия проследява темпа на развитие на IoT технологиите в света и България, както и развитието на Bluetooth Low Energy, ZigBee и 6LoWPAN протоколите като технологии за реализация на IoT мрежи.
Ключови думи: IoT, BLE, ZigBee, 6LoWPAN, сензорни мрежи

Research of Wireless Technologies for Internet of Things (IoT) Networks

Aydan M. Haka

Abstract: The widespread and active development of modern communication technologies, like 4G and 5G, provide a basis for the ubiquitous expansion and spread of Internet of Things (IoT) technologies. This article trace the trend of development of IoT technologies in the world and Bulgaria, as well as the development of the Bluetooth Low Energy, ZigBee and 6LoWPAN protocols, as technologies for the implementation of IoT networks.
Keywords: IoT, BLE, ZigBee, 6LoWPAN, sensor networks

1. Въведение

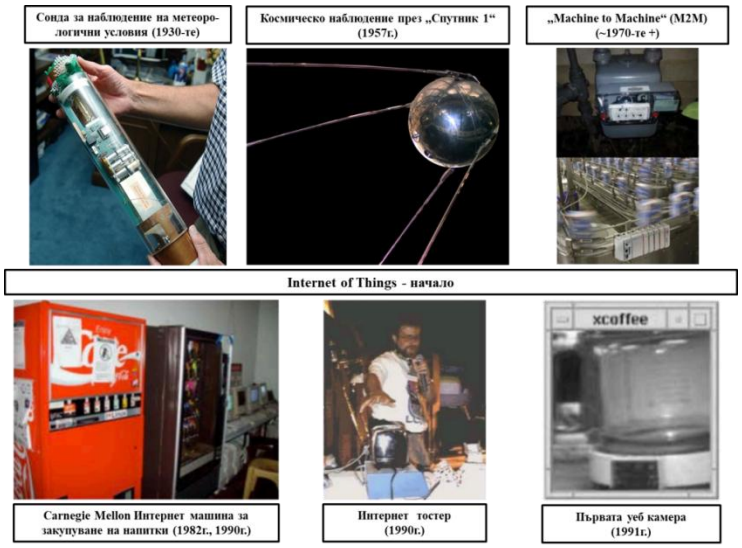
Съвременните комуникационни технологии като 4G и 5G заемат все по-голяма част от ежедневието и помагат за преодоляването на кризисни ситуации като тази с COVID-19 в световен мащаб по време на необходимата социална изолация. Благодарение на високоскоростните комуникационни инфраструктури става възможно провеждането на електронни здравни прегледи, обучения, срещи, събрания, конференции и др. Освен това за подобряване и поддържане на електронното здравеопазване, но и технологии, на чиято база се реализират и сензорни мрежи, се въвеждат все по-широко Internet of Things (IoT) технологиите, комуникацията на които разчита на високоскоростните мрежи като 4G и 5G [2, 3]. Тези технологии може да помогнат за отдалечено следене на параметри на околната среда и състоянието на пациенти в болнични заведения с цел предотвратяване разрастването на установени заразни заболявания и по-скоростно реагиране на спешни случаи.

Тази статия разглежда развитието на IoT технологиите в най-активно развиващите се пазари по света и в България. В частност се разглежда състоянието в световния пазар на едни от широко разпространените технологии за IoT – Bluetooth Low Energy (BLE), ZigBee и 6LoWPAN.

2. Развитие на Internet of Things мрежите

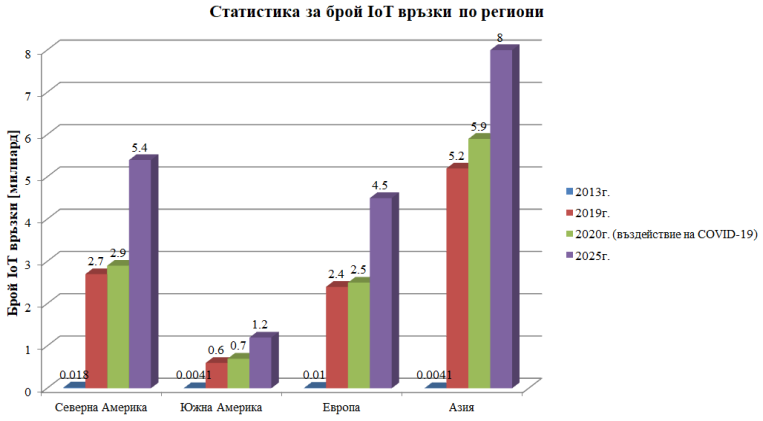
Основите на концепцията IoT се смята, че са положени през 1912 година в Чикаго с въвеждането на първата телеметрична система. За тази система се твърди, че използва телефонни линии за следене на данни от електроцентрали. Следенето на параметри продължава да се развива през годините, като през 1930-те се използват радиосонди за наблюдение на метеорологични условия. През 1957 година се основава аерокосмическата телеметрия от Съветския съюз с изстрелване на космическия апарат „Спутник“. През 1980-те започва широкото навлизане на Machine to Machine (M2M) технологиите, предназначени за

контрол и събиране на данни във фабрики и системи за домашна и бизнес сигурност (Фиг. 1). През 1990-те години M2M се насочват към безжичните технологии и се поставя началото на частните радио мрежи за наблюдение, а през 1995 година Siemens представя първия клетъчен модул за M2M. Втората вълна от навлизане и разработка на клетъчни решения за M2M става необходима, когато Федералната комисия за съобщения налага изключване на аналоговите мрежи в полза на цифровите [4].



Фиг. 1. Развитие на Internet of Things

В настоящето IoT мрежите разчитат на 4G и 5G инфраструктурите за бързо и надеждно предаване на малко количество данни, свързани с отдалечено следене параметри на околната среда, състояние на пациенти и др. През годините IoT технологиите стават все по-известни и се разпространяват все повече. Според [5, 6] за активно развиващите се региони по света IoT/M2M свързаността от милиони през 2013г. се разраства главоломно и достига до милиарди през 2019г. и 2020г (Фиг. 2). Разликата в броя на реализираната свързаност от 2019г. до 2020г. е значителна, като ефект върху разрастващия се брой указва глобалната пандемия от COVID-19. Такъв тип вирусни епидемии изискват отдалечено следене на параметри на околната среда или състояние на пациенти в болнични заведения, а съвременните IoT мрежи от сензорни възли може да отговорят на изискванията за това. Прогнозите [7] за 2025г. предвиждат увеличаване на IoT свързаността спрямо 2020г. с 2.5 милиарда за Северна Америка, 0.5 милиарда за Южна Америка, 2 милиарда за Европа и 2.1 милиарда за Азия.



Фиг. 2. Статистика на IoT свързаността от 2013г. до 2020г. и прогноза за 2025г.

Според [8] през 2019г. 4G свързаността е 4 милиарда, което е над 50% от общия брой свързаност, 5G свързаността се очаква да достигне 1 милиард, а IoT свързаността - 12 милиарда. До 2025г. се очаква 4G свързаността да достигне 5 милиарда, което е малко под 60% от общия брой свързаност, 5G - 1.8 милиарда, а IoT свързаността - 25 милиарда.

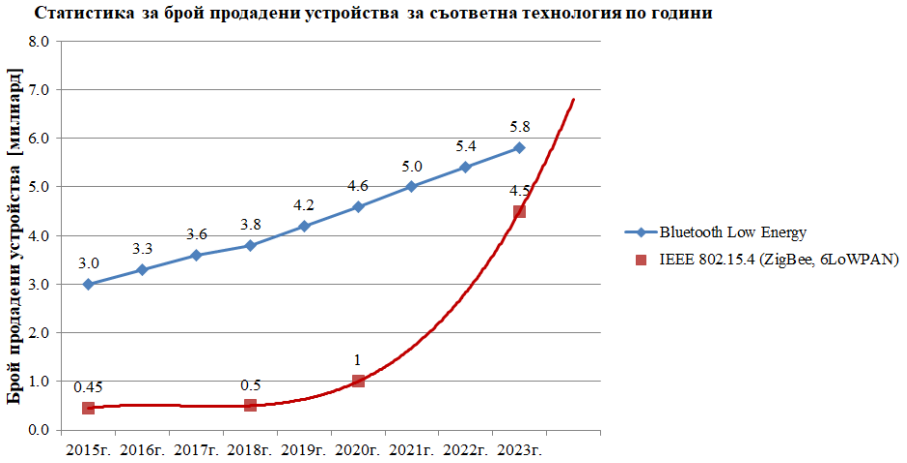
Съществуват множество IoT протоколи, които позволяват изграждане на сензорна мрежа за отдалечено следене, като Bluetooth Low Energy (BLE), ZigBee, Z-Wave, 6LoWPAN, Thread, WiFi-ah (HaLow) [9] и др.

Един от широко използваните протоколи е BLE. Сформираната през 1998г. търговска асоциация с нестопанска цел Bluetooth Special Interest Group (SIG) [10], която ръководи развитието на Bluetooth технологията, обслужва водещи в бранша компании членки по цял свят. Асоциацията се занимава основно със създаване на нови и подобрени Bluetooth спецификации, осигуряване на оперативна съвместимост на Bluetooth устройствата и увеличаване информираността, разбирането и приемането на технологията. Разрастването и разпространението на Bluetooth технологията, като една от основните за IoT, се потвърждава от увеличаващия се брой работни групи – 14, активни проекти за разработка на спецификации – 80, нови членове на работни групи и комитети през 2019г. – 2 089, както и броя на закупените Bluetooth устройства, който годишно нараства от 3 милиарда през 2015г. до 4.2 милиарда през 2019г. със среден темп на растеж от 0.33 милиарда на година. През 2020г. продажбите на Bluetooth устройства достигат до 4.6 милиарда, а до 2024г. се очаква да достигнат до 6.2 милиарда. Прогнозите [11] показват, че след 2020г. класическите Bluetooth устройства в пазара постепенно ще намаляват и все повече ще навлизат класически устройства, съвместими с BLE стандарта и изцяло базирани на BLE, като продажбите на тези устройства до 2024г. се очаква да достигнат 7.5 милиарда. Едно от иновативните решения, които предоставя BLE, е Low Energy Audio (LE Audio), което според Bluetooth SIG ще подобри производителността на Bluetooth звука, ще добави поддръжка на слухови апарати и въведе споделяне на звук.

Други значително разпространени IoT протоколи са ZigBee, стандартизиран и поддържан от ZigBee Alliance [12], и 6LoWPAN [13]. Протоколите се базират на IEEE 802.15.4 радио среда за комуникация и позволяват изграждане на пълно свързана мрежа. Протоколът ZigBee позволява използване на универсален език за комуникация между умни устройства, а 6LoWPAN предава малки пакети в мрежи с нискоенергийни устройства, като позволява предаване на пакети от сензорна мрежа през IPv6 мрежи. През първите години на разпространение на ZigBee чиповете продажбите достигат до хиляди на месец, а през 2018г. - до милиони. Статистиките за 2018г. [14] показват, че продажбите на IEEE 802.15.4 чипове в световен мащаб за „умни“ домове и сгради, измервания, „умни“ градове и индустриална автоматизация достигат половин милиард, а до 2023г. се очаква да са продадени около 4.5 милиарда устройства, от които 3.8 милиарда се очаква да използват ZigBee. Според прогнозите от 2019г. [15] продажбите на IEEE 802.15.4 чиповете до 2020г. ще достигнат 1 милиард, като продажбите на нови чипове до 2024г. се очаква да достигнат 1 милиард (Фиг. 3).

Основно BLE технологията се използва за: свързване на носими устройства като часовници, устройства за локализиране, здраве и фитнес, сензори по тялото; достъп до заобикалящи ни устройства като гривни за контрол на достъпа, домашна и офис автоматизация; M2M комуникация с нисък коефициент на работа като сензори и устройства за контрол в домове, офиси и фабрики; комуникация в рамките на затворена система; свързване на всичко, което има данни за интернет. Технологията ZigBee намира основно приложение, когато е необходима сигурна мрежа с дълъг живот на батерията на устройствата и с ниски изисквания за скорост при предаване на данни, като безжични ключове, електрически измервателни уреди с вграден дисплей, системи за контрол на трафика, потребителско и производствено оборудване с безжично свързани компоненти

нуждаещи се от нискоскоростна безжична връзка. Технологията 6LoWPAN основно приложение намира при контрол състоянието на оборудване, следене състоянието на околната среда, реализиране на сигурност и защита, приложение в домашна среда, автоматизация на сгради.



Фиг. 3. Статистика за брой продадени устройства за технология от 2015г. и прогноза до 2023г.

В Таблица 1 е представено сравнение на техническите спецификации на разглежданите технологии за IoT [16, 17, 18].

Таблица 1. Сравнение на технически спецификации на BLE, ZigBee и 6LoWPAN

Спецификации	**Bluetooth Low Energy**	**ZigBee**	**6LoWPAN**
Стандарт	IEEE 802.15.1	IEEE 802.15.4	IEEE 802.15.4
Предавателна мощност	15mW	15mW	15mW
Моментна скорост за предаване на данни	1Mbps	250Kbps	250Kbps
Производителност	67Mbps/W	17Mbps/W	17Mbps/W
Обхват	150m (пространство без препятствия)	300m (пространство без препятствия) 75-100m (на закрито)	75m
Топология	ad-hoc, Piconet (Звезда), Scatternet	Звезда, Дървовидна, Пълно свързване	P2P-Multicast, Звезда, Пълно свързване
Модулация	GFSK @ 2.4GHz	BPSK, O-QPSK	BPSK, O-QPSK
Защита	128bit AES CCM	128bit AES	AES-CCM-64
Комуникационен режим	Broadcast, Connection, Event Data Models Reads, Writes	Unicast, Groupcast, Broadcast	Unicast, Broadcast
Цена на устройства	Главно устройство 1брой ≈ от 77.27лв Сензорен възел 1брой ≈ от 3.40лв	Координатор 1брой ≈ от 185.46лв Сензорен възел 1брой ≈ от 8.50лв	Координатор 1брой ≈ от 185.46лв Сензорен възел 1брой ≈ от 80.66лв

Съвременните комуникационни технологии като 4G и 5G, както и тези за IoT се развиват активно през последните години във всички страни. В по-развитите страни това

става с по-високи темпове, а в зависимост от темпа на развитие и икономическото положение на страните се използват различни технологии и комуникационни протоколи.

3. Развитие на Internet of Things мрежите в България

В България по-широко разпространената високоскоростна комуникационна технология е 4G [1], като постепенно навлизат 5G [19, 20, 21] и IoT технологиите. С решение на Министерски съвет на Република България на 21 юли 2020г. е приет Национален стратегически документ „Цифрова трансформация на България за периода 2020-2030г.“ [22], целите на който се изразяват в: разгръщане на сигурна цифрова инфраструктура; осигуряване на достъп до адекватни технологични знания и цифрови умения; укрепване на капацитета за научни изследвания и иновации; отключване потенциала на данните; цифровизация в полза на кръгова и нисковъглеродна икономика; повишаване ефективността на държавното управление и качеството на публичните услуги. За постигане на поставените цели се предвижда спазване на следните принципи: ориентиран към потребителите подход и достъп на всички до цифрови услуги; етичен и социален отговорен достъп, използване, споделяне и управление на данните; технологиите като фактор от ключово значение; киберсигурност от етапа на проектиране; сътрудничество. Този документ обръща внимание на важността както от 4G и 5G, така и от IoT. Според документа IoT технологиите в България заедно с други иновативни такива се нуждаят от подкрепа в сферата на научни изследвания и иновации, цифрова икономика, енергетика, околна среда и климат.

Според националната програма „Цифрова България 2025“ [23], приета на 5 декември 2019г. с решение на Министерски съвет на Република България, ерата на цифровата трансформация настъпва скоростно, с потенциал да промени бизнеса и обществото. Увеличава се производителността, реализират се нови идеи, технологии, управленски и бизнес модели, създават се нови бизнес канали за достъп до пазара с относително ниски разходи, като потенциал за осъществяване на това имат IoT, изкуствен интелект и блокчейн технологиите. Ключови технологии за осъществяването на т.нар. четвърта индустриална революция и подобряването на сфери като здравеопазването, транспорта, енергетиката или публичните услуги са IoT, симулации, добавена/виртуална реалност (VR/AR), автономни роботи, облачни технологии (Cloud computing), триизмерно/адитивно отпечатване (3D printing), хоризонтална и вертикална системна интеграция, големи данни (Big Data), изкуствен интелект и когнитивни системи, машинно самообучение, интелигентни мобилни приложения, блокчейн технологии, цифрови платформи и др. Средство за свързване и предаване на данни между всички устройства се очаква да бъдат навлизащите 5G мрежи. За осъществяване на цифровата трансформация в България програмата отбелязва като приоритетна област създаването на подходящи условия за развитие на цифровите мрежи и услуги от операторите и предоставянето на по-добър достъп до тях. Според отчета по програмата [24] от 2019г. 75.1% от домакинствата в България имат достъп до интернет в домовете си. За достъп до интернет домакинствата използват мобилна връзка чрез мрежата на мобилните телефонни оператори (64.0%) и фиксирана кабелна връзка - 57.8%. Предприятията използващи високоскоростен, надежден и непрекъснат достъп до глобалната мрежа са 93.7%, като за достъп до интернет се използва основно DSL или друг вид фиксирана технология (80.5%), като при 64.8% от тях скоростта на най-бързата фиксирана връзка е по-висока от 30 Mbps.

В България телекомуникационните оператори, които основно приемат и внедряват IoT решения са A1 и Vivacom.

Стартирана през 2018г. IoT мрежата на A1 се базира на стандарта Narrowband-IoT (NB-IoT), чиито основни предимства са: по-ниски разходи за устройства, по-дълъг живот на батериите, по-добро покритие на закрито, оптимизиран пренос на данни, двупосочна

комуникация, ниско времево закъснение, висока сигурност на връзката, качество на предлаганите услуги, двупосочна комуникация и лицензиран спектър [25]. Операторът A1 предлага в България NB-IoT решения за контрол на въздуха, управление на отпадъци, интелигентно осветление и умно паркиране, първоначално тествани в Монтана [26].

Платформата представена през 2018г. Viva Smart на телеком оператора Vivacom обединява всичките ѝ IoT проекти. Чрез нея телекомът предлага „умни“ услуги в три направления: умен град (VIVA Smart City), облачни решения и колокация на оборудване (Smart Data Hub) и дигитално образование. Системата работи успешно в редица градове по цял свят като Дубай, Измир, Дъблин и др., и има различни модули за областите: мобилност; околна среда; енергия; безопасност; управление [27]. Платформата позволява всяка община, независимо от размера, броя жители и проблемите ѝ, да избере кои модули за „умни“ решения да използва като има възможност за надграждането им. Операторът Vivacom предоставя в Община Бургас IoT решения за комуникационна свързаност на системите за отдаване на велосипеди под наем, интелигентно видео наблюдение на градската среда и интегрирания градски транспорт, а през 2018г. стартира пилотно решение за „умно“ паркиране в града, базирано на LoRaWAN технологията. Освен това стартира и пилотен проект за „умно“ сметосъбиране в Община Пловдив [28].

Телеком операторът Telenor има опит в ранните IoT решения, известни като M2M комуникация в Норвегия и Швеция, а през 2018г. стартира NB-IoT технология за IoT [29].

В България 6LoWPAN протоколът [30] се използва в пилотна мрежа за IoT, стартирана през 2016г. в София. Технологията BLE основно се използва при спорт и развлечения, а през 2019г. в София се провежда среща за популяризиране на BLE стандарта като IoT решение [31].

4. Заключение

Съвременните комуникационни технологии в световен мащаб се насочват към високоскоростните технологии, известни като 5G, наред с това навлизат и се използват все повече IoT технологиите. Широкото разпространение на IoT технологиите в света през последните години е предпоставка за очакване все по-голямото им разрастване и разширяване. България следи посоката и тенденцията на комуникационното развитие в цял свят, но със значително по-бавни темпове. В последните години операторите продължават с тестването, разширяването и мигрирането към 5G технологиите, като се поддържат и подобряват съществуващите 4G. Наред с това се обръща внимание и на важността и ползите от IoT, и се внедряват и тестват все повече решения, отговарящи на изискванията и съществуващите проблеми в страната.

Литература

[1]. Хъкъ, Айдън. Изследване на 4G клетъчни безжични мрежи. //Компютърни науки и технологии, ТУ-Варна, 2017, Брой 1, стр. 46 – 50, ISSN 1312 – 3335

[2]. Ericsson Mobility Report, June 2020. https://www.ericsson.com/en/mobility-report/reports/june-2020, Last visit on 14.08.2020

[3]. 5G Americas White Paper, 5G THE FUTURE OF IOT, July 2019, https://www.5gamericas.org/white-papers/, Last visit on 14.08.2020

[4]. Zennaro M. Intro to Internet of Things ITU ASP COE TRAINING ON “Developing the ICT ecosystem to harness IoTs”, 13-15 December 2016, Bangkok, Thailand. https://www.itu.int/en/ITU-D/Regional-Presence/AsiaPacific/SiteAssets/Pages/Events/2016/Dec-2016-IoT/IoTtraining/IoT%20 Intro-Zennaro.pdf, Last visit on 14.08.2020

[5]. GSMA Connected Living Understanding the Internet of Things (IoT), July 2014, https://www.gsma.com/iot/wp-content/uploads/2014/08/cl_iot_wp_07_14.pdf, Last visit on 14.08.2020
[6]. GSMA 5G in IoT and the Impact of COVID-19, 17 Jun 2020, https://www.gsma.com/iot/resources/5g-in-iot-and-the-impact-of-covid-19/, Last visit on 14.08.2020
[7]. GSMA IoT Connections, https://www.gsma.com/mobileeconomy/, Last visit on 14.08.2020
[8]. GSMA The Mobile Economy 2020, https://www.gsma.com/mobileeconomy/wp-content/uploads/2020/03/GSMA_MobileEconomy2020_Global.pdf, Last visit on 14.08.2020
[9]. Link Labs Wireless IoT Network Protocols, https://www.link-labs.com/blog/complete-list-iot-network-protocols, Last visit on 14.08.2020
[10]. Bluetooth, https://www.bluetooth.com/, Last visit on 14.08.2020
[11]. Bluetooth Market Update 2020, https://www.bluetooth.com/?utm_campaign=markets-story&utm_source=internal&utm_medium=paper&utm_content=mktupdaterpt-btcom-pdflink%0D, Last visit on 14.08.2020
[12]. zigbee alliance, https://zigbeealliance.org/, Last visit on 14.08.2020
[13]. electronicsnotes, What is 6LoWPAN - the basics, Last visit on 14.08.2020
[14]. Zigbee Leads the Wireless Mesh Sensor Network Market, https://zigbeealliance.org/news_and_articles/zigbee-leads-the-wireless-mesh-sensor-network-market/#:~:text=In%20fact%2C%20wireless%20mesh%20sensor,Half%20a%20billion%20802.15., Last visit on 14.08.2020
[15]. RESEARCHANDMARKETS, 802.15.4 IoT Markets to 2025 - World Analysis by Product Design, Technology, Frequency, Topology, Geography, Chipset/Module Revenues & Average Sale Price, https://www.globenewswire.com/news-release/2019/11/12/1945159/0/en/802-15-4-IoT-Markets-to-2025-World-Analysis-by-Product-Design-Technology-Frequency-Topology-Geography -Chipset-Module-Revenues-Average-Sale-Price.html, Last visit on 14.08.2020
[16]. Decuir J., IEEE, Bluetooth 4.9: Low Energy, https://californiaconsultants.org/wp-content/uploads/2014/05/CNSV-1205-Decuir.pdf, Last visit on 14.08.2020
[17]. Zigbee Technical Presentation, 2019, https://zigbeealliance.org/developer_resources/zigbee-technical-presentation/, Last visit on 14.08.2020
[18]. Kushalnagar N., Montenegro G., 6LoWPAN: Overview, Assumptions, Problem Statement and Goals, https://tools.ietf.org/html/draft-kushalnagar-lowpan-goals-assumptions-00, Last visit on 14.08.2020
[19]. 5G мрежа за забавление от ново поколение, https://www.a1.bg/5g, Last visit on 14.08.2020
[20]. 5G тестове за по-добро бъдеще, https://www.telenor.bg/5g-tests, Last visit on 14.08.2020
[21]. VIVACOM демонстрира 5G в реалната си мрежа, https://vivacom.bg/bg/residential/za-nas/novini/finansi?read=vivacom-demonstrira-5g-v-realnata-si-mreja, Last visit on 14.08.2020
[22]. Цифрова трансформация. Национален стратегически документ „Цифрова трансформация на България за периода 2020-2030г.“ https://www.mtitc.government.bg/bg/ category/283, Last visit on 14.08.2020
[23]. НАЦИОНАЛНА ПРОГРАМА ЦИФРОВА БЪЛГАРИЯ 2025, https://www.mtitc.government.bg/bg/category/85/otchet-na-nacionalna-programa-cifrova-bulgariya-2025-i-putna-karta-kum-neya-kum-dekemvri-2019-g, Last visit on 14.08.2020
[24]. Отчет по изпълнението на Национална програма „Цифрова България 2025“ към декември 2019 г., https://www.mtitc.government.bg/bg/category/85/otchet-na-nacionalna-programa-cifrova-bulgariya-2025-i-putna-karta-kum-neya-kum-dekemvri-2019-g, Last visit on 14.08.2020
[25]. A1 Блог, Narrowband IoT – бъдещето на умните градове, https://blog.a1.bg/2018/04/20/narrowband-iot-future-smart-cities/, Last visit on 14.08.2020

[26]. A1 Блог, Narrowband IoT – по-ниски разходи, по-добър пренос на данни, https://blog.a1.bg/2019/03/27/narrowband-iot-better-than-the-rest/, Last visit on 14.08.2020
[27]. VIVA SMART – НОВАТА ПЛАТФОРМА НА VIVACOM ЗА ЦЯЛОСТНИ IoT УСЛУГИ И РЕШЕНИЯ, https://www.vivacom.bg/bg/residential/za-nas/novini/finansi?read=viva-smart-novata-platforma-na-vivacom-za-c-ja-lostni-io-t-uslugi-i-re-sh-eni-ja, Last visit on 14.08.2020
[28]. VIVACOM, Интегриран годишен доклад 2018, https://www.vivacom.bg/web/catalog/sustainable/files/assets/common/downloads/publication.pdf, Last visit on 14.08.2020
[29]. Telenor, Trusted IoT solutions from one of the world's major mobile operators, https://www.telenor.com/innovation/internet-of-things/, Last visit on 14.08.2020
[30]. COMPUTERWORLD, Български стартъп изгражда в София пилотна мрежа за Интернет на нещата, https://computerworld.bg/entrepreneurship/2016/09/14/3461161_bulgarski_startup_izgrajda_v_sofiia_pilotna_mreja_za/, Last visit on 14.08.2020
[31]. Технологията Bluetooth Low Energy и нейното приложение в Java, 2019, https://www.facebook.com/events/781086382243954/?active_tab=about, Last visit on 14.08.2020

За контакти:
ас. д-р инж. Айдън Мехмед Хъкъ
катедра „Компютърни науки и технологии"
Технически университет – Варна
E-mail: aydin.mehmed@tu-varna.bg

СРАВНИТЕЛЕН АНАЛИЗ НА LI-FI БЕЗЖИЧНИ МРЕЖОВИ СИМУЛАТОРИ ЗА ОБРАЗОВАТЕЛНИ ЦЕЛИ

Диян Ж. Динев

Резюме: Нарастващият в световен мащаб брой мобилни устройства изисква използването на по-високи скорости на пренос на данни. Li-Fi технологията за Internet of Things осигурява високоскоростен, двупосочен и сигурен безжичен достъп. Това изисква проучване на тази технология по отношение на качеството на услугата. Това може да стане чрез използване на симулационен софтуер, който ще сведе до минимум разходите и времето за изграждане на такава мрежа. Тази статия представя сравнителен анализ между разработен от автора симулатор и някои от най-известните симулатори за изследване на качеството на услугите в Li-Fi Internet of Things мрежи. Предложена е система от критерии за извършване на сравнителен анализ на симулатори за Li-Fi мрежа.
Ключови думи: Анализ, IoT, Li-Fi, Симулатори на мрежи

Comparative Analysis of Li-Fi Wireless Network Simulators for Education Purposes

Diyan Zh. Dinev

Abstract: Worldwide growing number of mobile devices requires the use of higher speed data transfer rates. The Internet of Thinks Li-Fi technology provides high speed, bidirectional and secure wireless access. That requires the study of that technologiy in terms of quality of service. This can be done by using simulation software that will minimize the cost and time of building such a network. This paper presents information about comparative analysis between author's proposed simulator and some of the most well-known simulators to explore the quality of service on the Li-Fi Internet of Things networks. A system of criteria for performing a comparative analysis of simulators for the Li-Fi network is proposed.
Keywords: Comparison, IoT, Li-Fi, Network Simulators

1. Въведение

Комуникациите при Internet of Things (IoT) се базират основно на безжични мрежи, където се търсят решения за повишаване на ефективността в условията на ниски скорости, устойчивост на откази, адаптивност, възможност за самоорганизация. Съвременните безжични мрежи обаче, понякога не могат да осигурят нужната скорост на комуникация, особено в случаите, когато в мрежата са свързани много устройства. В подобни случаи фиксираната широчина на честотната лента на радио мрежите (Radio Frequency – RF) може да направи скоростите на предаване по-ниски и да намали възможността за свързване към защитена мрежа при достигането на голям брой свързани в мрежата устройства.

С активното развитие на клетъчните мрежи от четвърто и пето поколение [1] се развиват и клетъчните IoT мрежи [2]. Постепенното навлизане на тези мрежи в заобикалящия ни свят през последните години [1] и внедряването им за различни приложения в ежедневието диктува необходимостта от изследване на такъв тип мрежи.

Изграждането на цялостна реална инфраструктура може да спомогне за изследването на качеството на услугите при мрежи за Интернет на обектите. Това изисква сериозни инвестиции и е икономически неоправдано в повечето случаи. Преди обаче да се инвестира във физическото оборудване, е по-разумно да се изгради и изследва такава мрежа, като се използва симулационна среда.

Основно предимство на симулаторите е възможността за виртуално изграждане на мрежа за изследваната технология, благодарение на което може да се моделира нейната цялостна работа. Главен недостатък на тези симулационни продукти е, че не винаги предоставят достоверни резултати, които биха се получили при реално изградена мрежа. Те могат да предоставят близки до реалните резултати и обща визия, за да се направят изводи за функционирането на даден алгоритъм за подобряване качеството на обслужване.

Този подход за изследване на Li-Fi мрежа за IoT може да се въведе и в образованието. За целите на обучението основните изисквания към една симулационна среда са:

- моделиране на различни механизми за приоритизация на трафика при Li-Fi;
- моделиране на различни видове трафик;
- възможност за изграждане на различни мрежови архитектури за Li-Fi;
- възможност за визуално представяне на изследваната мрежа;
- анализ на получените резултати;
- лиценз за ползване;
- лесна инсталация;
- време за изучаване;
- нагледно представяне на въведените данни;
- извеждане на статистически резултати за направените тестове;
- използване на процесора;
- използване на паметта.

Тази статия представя информация за сравнителен анализ между предложения от автора симулатор и някои от най-известните симулатори за изследване на качеството на услугата в Li-Fi Internet of Things мрежите. Предложена е система от критерии за извършване на сравнителен анализ на симулатори за Li-Fi мрежа.

2. Критерии за анализ на симулационни среди за Li-Fi мрежи

През последните години в литературата са публикувани редица проучвания и доклади за оценка на различни среди за симулация на безжични мрежи. Например в [3] авторите сравняват пропускателна способност и закъснението при предаване на данни, моделирани в OMNET ++, NS-2 и OPNET. В [4] се представя казус за анализ и сравнение на J-Sim [6], OMNeT ++, NS-2 и NS-3 [5] инструменти чрез внедряване на прост алгоритъм за мрежово управление. Инструментите се оценяват по отношение на използваемостта, усилията за инсталиране и изпълнение на симулация, наличието на използваеми модели и други аспекти. Според авторите всеки от инструментите има своите силни и слаби страни в сравнение с другите кандидати и няма ясно изразен победител сред сравняваните симулатори.

В [7] се сравнява ефективността на популярните мрежови симулатори, включително NS-2, NS-3, OMNet ++ и др. Критериите за оценка включват производителността на изпълнението на симулацията и отпечатъка на паметта за различни размери на мрежата.

Изследването, представено в [8], описва два набора от критерии за сравнение на симулатори за безжични мрежи:

- първи набор: език за програмиране, документация, използвана операционна система, графичен потребителски интерфейс, година на последна версия;
- втори набор: моделиране на консумацията на енергия, моделиране на мобилност, скалируемост, разширение на функционалността.

В разгледаните проучвания използваните критерии не са достатъчни за оценяване на симулационните среди по отношение на обучението. За целта може да се обединят част от представените в [4, 5, 6, 7, 8] критерии с тези, касаещи обучението. Това се прави, за да се прецизира комплексната оценка на сравняваните симулатори. За извършване на комплексния сравнителен анализ се предлага следната комбинация от критерии:

- моделиране на различни механизми за приоритизация на трафика при Li-Fi;
- моделиране на различни видове трафик;
- възможност за изграждане на различни мрежови архитектури за Li-Fi;
- симулиране на Li-Fi мобилност;
- анализ на получените резултати;
- възможност за визуално представяне на изследваната мрежа;
- нагледно представяне на въведените данни;
- извеждане на статистически резултати за направените тестове;
- поддържане на графичен потребителски интерфейс;
- наличие на ръководство за потребители и разработчици;
- поддържана операционна система;
- скалируемост;
- език за програмиране;
- лесна инсталация;
- използване на паметта (Commit);
- лиценз за ползване.

3. Комплексна оценка за качеството на симулационни среди за Li-Fi безжични мрежи

Поради очевидната разнотипност на показателите за оценка на симулатори, извеждане на конкретна функция за качеството на различните алгоритми е невъзможна. В такива случаи се използват усреднени комплексни показатели. За да се изчисли комплексният показател за качество, се избира една от следните математически зависимости – квадратична, геометрична, аритметична или хармонична.

Методът с усреднени комплексни оценки [9] може да се използва за реализиране на сравнителен анализ между симулатори за изследване на качеството на услугите при Li-Fi мрежи. Аритметичният комплексен показател се изчислява по формулата:

$$R_a = \frac{\sum_{i=1}^{n} b_i \, d_i}{\sum_{i=1}^{n} b_i} \tag{1}$$

Геометричният комплексен показател се изчислява по следната формула:

$$R_г = \left(\prod_{i=1}^{n} d_i^{b_i} \right)^{\frac{1}{\sum_{i=1}^{n} b_i}}, \tag{2}$$

където нормираната количествена оценка на і-тия показател е:

$$0 \le d_i \le 1 \tag{3}$$

n - брой на показателите, а b_i е i-я тегловен коефициент, като:

$$\sum_{i=1}^{n} b_i = 1 \tag{4}$$

За целите на изследването тегловните коефициенти b_iсе вземат равни.

Съществуват редица симулационни среди, позволяващи симулиране на Li-Fi технология. В настоящото изследване са разгледани едни от най-известните: OptSim, Veins

VLC, NS-2, NS-3, MATLAB. Разглежданите среди са сравнени с предложения симулационен софтуер и е представена средна аритметична и средна геометрична оценка на всеки от тях. На база получените оценки може да се определи най-подходящият симулационен софтуер за целите на обучението.

OptSim [13] – комерсиална симулационна среда, осигуряваща моделиране и симулация на мрежи и системи на ниво разпространение на сигнала. С лесен за използване графичен потребителски интерфейс и голямо количество софтуерни инструменти за измерване на сигналите, е удобна както за ползване в индустрията, така и за работа в научните среди.

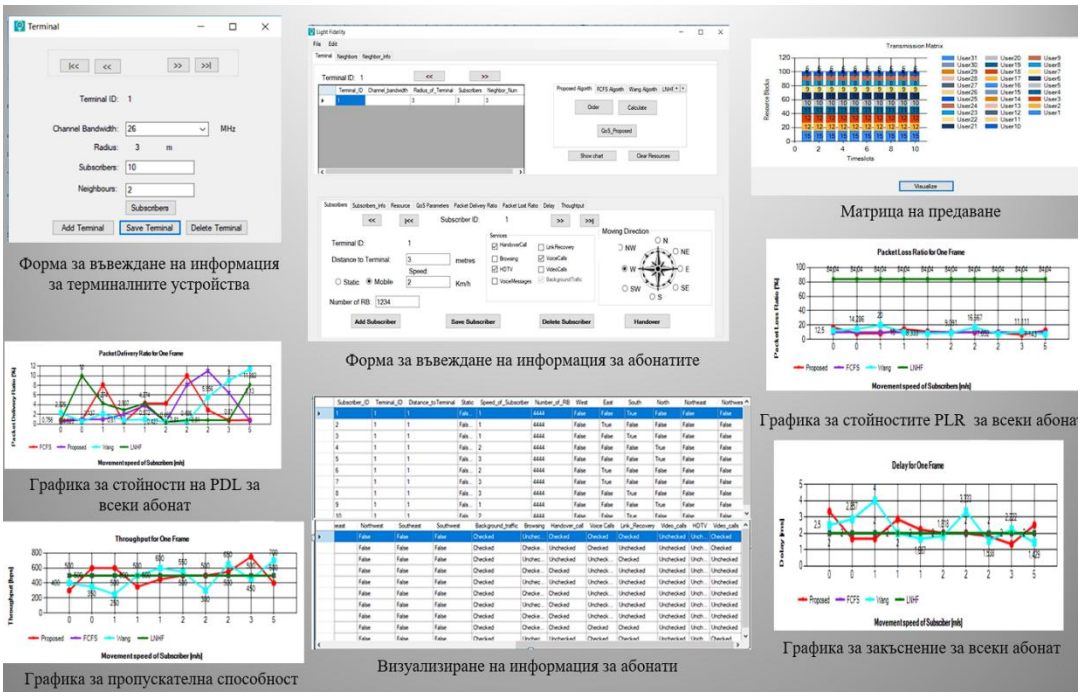
VEINS VLC [14] – симулатор с отворен код за достъп, даващ възможност за проучването на VLC мрежите в различни аспекти. Той включва множество приложения на VLC мрежите, хетерогенни мрежи, комбиниращ VLC с IEEE 802.11 и др.

NS-2 [11] – безплатен мрежов симулатор, на C++, осигуряващ модулна поддръжка на различни безжични мрежи. Позволява симулиране на TCP трафик и маршрутизиране. Има богата и добре организирана документация за работа със симулатора. Не е удобен за реализацията на големи мрежи.

NS-3 [12] – симулатор с отворен код, предназначен за научни и образователни цели. Продуктът е лицензиран за изследвания и разработки. Предоставят се допълнителни модули за изграждане на VLC симулационни модели на мрежи.

MATLAB [10] – изчислителна среда и патентован скриптов език, разработен от MathWorks. Продуктът съчетава в себе си работна среда, настроена за итеративен анализ и проектиране на процеси, прилагане на алгоритми, графики на функции и данни, взаимодействие с други програми и др. MATLAB има разработен продукт за Li-Fi - „Li-Fi Toolbox“. Той предоставя функции, съвместими със стандарта и приложения за проектиране, симулация и проверка на Li-Fi комуникационни системи. Основен негов недостатък е високата цена за използване на пакета “ Li-Fi Toolbox“.

На фигура 1 са показани основните функционалности на разработения симулационен продукт, описан в [15, 16].



Фиг. 1 Предложен симулационен продукт

В Таблица 1 са представени основните предимства и недостатъци на разгледаните симулатори за Li-Fi безжични мрежи.

Таблица 1. Основни предимства и недостатъци на Li-Fi симулатори

Симулатор	Предимства	Недостатъци
OptSim	предоставя лесно проследяване и визуализиране на получените резултати; голям набор от средства за изследване на сигналите.	платен лиценз за ползване; недостатъчна надеждност заради грешки.
VEINS VLC	отворен код; модулна система; сложните сценарии може да бъдат тествани просто; множество поддържани протоколи и платформи.	липса на правдоподобност; трудно се моделира реална система.
NS-2	безплатен, не изисква скъпо оборудване; сложните сценарии може да бъдат тествани просто; лесен и бърз начин за отстраняване на грешки; множество поддържани протоколи и платформи;	трудно се моделира реална система; недостатъчна надеждност заради грешки; лоша визуализация на мрежата.
NS-3	безплатен; модулна система; има възможност за модулни библиотеки; индивидуалните модули съдържат директорийна структура; позволява на възлите да използват външно маршрутизиране.	липса на правдоподобност; модули, чиито компоненти са базирани на NS-2 не работят; лоша визуализация на мрежата; необходима активна поддръжка.
MATLAB	интерпретира кода по време на писането; предоставя лесно извършване на изчисления и визуализиране на резултатите; предоставя функции и приложения, съвместими със стандарта, за дизайн, симулиране и проверка на Li-Fi системи.	може да бъде значително бавен; изисква добра хардуерна платформа за ускоряване на работата; няма персонализиран модул за Li-Fi; платен лиценз за ползване.

Съществуващите симулатори на Li-Fi мрежи притежават редица недостатъци, свързани с работата и функционалността им, като:

- висока цена за ползване на продукта;
- сложен графичен потребителски интерфейс;
- невъзможност за симулиране на мрежа с множество възли;
- сложна настройка на продукта;
- довеждат до натоварване на компютърната система;
- изискват квалифицирани потребители;
- трудно или невъзможно модифициране на предоставения код;
- не предоставят нагледно представяне разпределението на ресурсите по потребители.

Проведени са експерименти с въвеждане на параметри на една и съща Li-Fi мрежа с едни и същи входни данни с всеки от симулаторите. Резултатите за всеки от предложените критерии са представени в Таблица 2.

Таблица 2. Критерии за оценяване на предложения и разглежданите симулатори за Li-Fi безжични мрежи

Критерии за сравнение на LTE симулатор	Изследвани симулатори					
	OptSim	**VEINS VLC**	**NS-2**	**NS-3**	**MATLAB**	**Динев**
Моделиране на различни алгоритми за приоритизация на трафика при Li-Fi	Да частично	Да частично	Да частично	Да напълно	Да напълно	Да напълно
Моделиране на различни методи за	Да частично	Да частично	Да напълно	Да напълно	Да напълно	Да напълно

разпределяне на ресурсите						
Симулиране на Li-Fi мобилност	Да частично	Да частично	Не	Да частично	Не	Да напълно
Генериране формата на Li-Fi вълни	Не	Да напълно	Не	Да напълно	Да напълно	Не
Моделиране на различни видове трафик	Да напълно	Да частично	Да напълно	Да напълно	Да напълно	Да напълно
Поддържане на графичен потребителски интерфейс	Да напълно	Да напълно	Да напълно	Да напълно	Да напълно	Да напълно
Наличие на ръководство за потребители и разработчици	Да частично	Да напълно	Да напълно	Да напълно	Да напълно	Да напълно
Анализ на получените резултати	Да напълно	Да напълно	Да напълно	Да напълно	Да напълно	Да напълно
Възможност за визуално представяне на изследваната мрежа	Да напълно	Да напълно	Да напълно	Да напълно	Да напълно	Да частично
Лесна инсталация	Да частично	Да напълно	Да частично	Да частично	Да напълно	Да напълно
Скалируемост	Да напълно	Да напълно	Да частично	Да частично	Да напълно	Да частично
Език на програмиране	C и C++	C++	C++,	C++, Python	Matlab script, Python	Visual Basic
Използване на паметта	649 510KB	450 874KB	560 840KB	570 932KB	940 832KB	230 732KB
Лиценз за ползване	Платен търговски	Платен търговски	Безплатен	Безплатен	Платен стандартен и академичен + Допълнителен платен пакет за Li-Fi	Безплатен

В Таблица 3 са представени комплексните аритметични и комплексните геометрични оценки на изследваните симулатори, съгласно формули (1) и (2), базирайки се на извършените експерименти, резултатите, от които, са представени в Таблица 2.

Таблица 3. Комплексни оценки за сравнение на симулационни продукти за Li-Fi мрежи

Комплексен показател	**Изследвани симулатори**					
	OptSim	**VEINS VLC**	**NS-2**	**NS-3**	**MATLAB**	**Динев**
$\mathbf{R_a}$	0.364	0.326	0.360	0.369	0.399	0.342
$\mathbf{R_z}$	0.236	0.232	0.251	0.289	0.328	0.233

4. Заключение

Според представените резултати от изследването поради широкия набор от разглеждани критерии най-подходящи за изследване на Li-Fi мрежите се явяват MATLAB, OptSim и NS-3. При изследване на критериите поотделно, обаче, се вижда, че предложеният симулатор от автора предоставя по-добри стойности за показателите: “Лесна инсталация“,

„Използвана памет“ и „Лиценз за ползване“, които са от изключителна важност по отношение на използването на симулатор при обучение.

В критериите за сравняване не е посочен критерии „Визуализация на матрицата на предаване“, т.к. за разлика от предложения от автора симулатор, нито един от останалите симулатори не предоставя тази възможност. Той предоставя съизмерими, с останалите симулатори, резултати спрямо много други критерии. Това доказва, че предложеният от автора симулатор е много подходящ за целите на обучението.

Литература

[1]. Ericsson Mobility Report, June 2020. https://www.ericsson.com/en/mobility-report/reports/june-2020, Last visit on 10.08.2020

[2]. 5G Americas, IoT Deployments, https:5gamericas.org/resources/deployments-iot/, Last visit on 11.12.2020

[3]. TIMM-GIEL, A., MURRAY, K., BECKER, M., LYNCH, C., GORG, C., AND PESCH, D. 2008. Comparative simulations of WSN. In Proceedings of ICT Mobile and Wireless Communications Summit. Stockholm,Sweden.

[4]. LESSMANN, J., HEIMFARTH, T., AND JANACIK, P. 2008a. ShoX: An easy to use simulation platform for wireless networks. In Proceedings of the 10th EUROS/UKSim International Conference on ComputerModelling and Simulation. IEEE, Cambridge, UK, 410–415.

[5]. LESSMANN, J., JANACIK, P., LACHEV, L., AND ORFANUS, D. 2008b. Comparative study of wireless networksimulators. In Proceedings of the 7th International Conference on Networking. Cancun, Mexico, 517–523.

[6]. SOBEIH, A., VISWANATHAN, M., MARINOV, D., AND HOU, J. C. 2007. J-sim: An integrated environment for simulation and model checking of network protocols. In Proceedings of the 21st IEEE International Parallel and Distributed Processing Symposium. Long Beach, California USA, 1–6.

[7]. WEINGARTNER, E., VOM LEHN, H., AND WEHRLE, K. 2009. A performance comparison of recent network simulators. In Proceedings of the 5th IEEE International Conference on Communications. Dresden, Germany, 1–5.

[8]. Saidallah, M., Fergougui, A. E., Elalaoui, A. E. A Survey and Comparative Study of Open-Source Wireless Sensor Network Simulator. //International Journal of Advanced Research in Computer Science (IJARCS), Vol. 7, No. 3, 2017, E-ISSN: 0976-5697

[9]. Алексиева, В., Вълчанов, Хр., Атанасов, Е. Електронна търговия, ръководство за лабораторни упражнения. //Университетско издателство при ТУ-Варна, 2019, pp: 7-9, ISBN: 978 954-20-0798-2

[10]. MATLAB Simulator. https://uk.mathworks.com/products/matlab.html, Last visit on 10.11.2020

[11]. NS2 Simulator. http://ns2simulator.com/, Last visit on 10.11.2020

[12]. NS3 Simulator. http://ns3-code.com/ns3-download/, Last visit on 10.11.2020

[13]. OPSim Smilutor, https://www.synopsys.com/photonic-solutions/rsoft-system-design-tools/system-network-optsim.html, Last visit on 10.11.2020

[14]. Veins VLC Simulator, https://www.ccs-labs.org/software/veins-vlc/, Last visit 10.11.2020

[15]. Dinev, D., Aleksieva, V., Valchanov, H. Comparative Analysis of Prototypes Based on Li-Fi Technology. // Proceedings, 2019 16-th Conference on Electrical Machines, Drives and Power Systems (ELMA), 6-8 June 2019, Varna, Bulgaria, pp: 531-534, ISBN: 978-1-7281-1412-5

[16]. D. Dinev, Comparative Analysis of Traffic Prioritisation Algorithms in Li-Fi Networks, 2020 International Conference on Biomedical Innovations and Applications (BIA), Varna, Bulgaria, 2020, pp. 121-124, doi: 10.1109/BIA50171.2020.9244517.

За контакти:
ас. инж. Диян Желев Динев
катедра „Софтуерни и интернет технологии”
Технически университет – Варна
E-mail: diyandinev@tu-varna.bg

АЛГОРИТМИ ЗА НАМИРАНЕ НА НАЙ-КРАТЪК ПЪТ В МАРШРУТ НА ЛЕТАТЕЛЕН АПАРАТ, МОДЕЛИРАН ЧРЕЗ ГРАФ

Илиян Ж. Бойчев

Резюме: В настоящата статия са представени два алгоритъма за намиране на най-кратък път в граф - алгоритъм на Дийкстра [1] и алгоритъм с минимален брой възли [1]. Графът представя летателен маршрут на безпилотен летателен апарат. Представен е подобрен алгоритъм, който доразвива и усъвършенства алгоритъма за намиране на най-кратък път с минимален брой възли. Предложеният алгоритъм намалява времето за търсене на път в степен, зависеща от близостта между началния и крайния възел в графа.
Ключови думи: алгоритъм на Дийкстра, алгоритъм с минимален брой върхове, граф, безпилотен летателен апарат

Algorithms for Finding the Shortest Path in an Aircraft Route Modeled by Graph

Iliyan Zh. Boychev

Abstract: This article presents two algorithms for finding the shortest path in a graph (Dijkstra algorithm and algorithm with minimum number of nodes), described during the flight route of unmanned aerial vehicles. An improved algorithm is presented. It improves the algorithm for finding the shortest path with a minimum number of nodes. Suggested algorithm reduces the path search time to a degree, which depends on the proximity between the initial and final node in the graph.
Keywords: Dijkstra algorithm, algorithm with minimum number of nodes, graph, unmanned arial vehicle

1. Увод

Развитието на компютърното инженерство води до все по-голямо автоматизиране на производствени процеси и намаляване на участието на човек в тях, като то се свежда до наблюдение и поддръжка на системите. Алгоритмите за управление на тези системи се развиват до степен, в която системите се самоуправляват или вземат решения на базата на изкуствен интелект.

В последните години използването на летателни апарати се разширява. Използвани първоначално за военни цели, към този момент те намират масово употреба за цивилни цели от всякакъв тип (за фотографии, заснемане на филми и събития, логистични и транспортни услуги и др.). Наред с използването им за любителски цели, те заемат основно място за следене на подвижни и отдалечени обекти, което включва приложения като: обследване на трудно достъпни места, следене на движещи се обекти, откриване на природни бедствия и т.н. Това ги прави незаменим помощник, тъй като ограничават достъпа на хора до опасни за здравето и живота места, а едновременно с това се намалява времето за изпълнение на задачите. Възможността за прикрепяне на камера ги прави безценен помощник в приложения, изискващи наблюдение в реално време.

В представените изследвания летателните апарати се използват в контекста на приложение за осигуряване наблюдение на сгради в зона на сигурност с голяма площ. Експерименталните постановки са насочени към изследване на възможностите за обхождане и наблюдение на множество сгради, разположени в района на Технически университет - Варна.

Алгоримите, представени в статията, са изследвани за различни топологии на свързване на обекти, представени в [2], а моделирането на граф - в [3].

2. Алгоритми за намиране на най-кратък път

2.1. Алгоритъм на Дийкстра

Този алгоритъм е най-ефективният за намиране на път от даден връх до всички останали върхове в дадения граф. Алгоритъмът работи правилно, само ако теглата на дъгите в графа са положителни числа. Това го прави подходящ за използване при определяне на най-краткият път на летателен маршрут, тъй като теглата в графовете, съставени при различни топологии на свързване на сгради, са разстояния, т.е. те задължително са положителни числа и са по-големи от 0.

Като недостатък на алгоритъма на Дийкстра може да се посочи, че той намира най-краткият път като абсолютна стойност, но не съхранява върховете, през които се преминава по този път. В същото време, за определяне на летателния маршрут е необходимо да се съхранява информация за върхове от графа, през които се преминава, за да може да се извлекат координатите на тези върхове (точки) и да се зададе автономен полет на летателния апарат. За реализация на навигацията по време на самия полет е от важно значение да се знае кои са точките, през които преминава траекторията на маршрута, а не пряко дължината на този път. Следователно, след като се намери кой е минималният път, трябва да се съхранят и всички точки по него.

За съхраняване на върховете по време на търсене на минималните пътища се използва подобрен алгоритъм на Дийкстра, при който се въвежда масив за съхранение на предшествениците. След приключване на алгоритъма, от този масив може да се намери минималният път между началния и краен връх.

Алгоритъм на Дийкстра:

1. Задава се начален връх Vs, от който започва търсенето на най-кратък път. Инициализира се масив с предшествениците parents [N]

2. Инициализира се масив d[N] с разстоянията между Vs и всички останали върхове. Първоначално стойностите на елементите се инициализират по следните правила:

- *Ако връх Vi е съседен на Vs, то d[i] = matrixDistances[s][i]*
- *Ако връх Vi не е съседен на Vs, до d[i] = MAX_DISTANCE*

3. Инициализира се множество V`, което съдържа всички върхове от графа, без началния връх Vs: V`= V\{Vs}

4. Проверява се дали има поне един връх Vi, за който разстоянието d[i] < MAX_DISTANCE. Ако има такъв връх, се изпълняват следните операции:

4.1. Намира се такъв връх Vj, за който разстоянието d[j] да е минимално

4.2. Връх Vj се премахва от множеството V: V` = V`\{Vj}

4.3. За всеки връх Vi от V`се намира минималното разстояние:

$$d[i] = min(d[i], d[j] + matrixDistances[j][i])$$

4.4. Записва се текущият връх Vj като връх в parents[]:

$$parents[i] = Vj$$

Основен недостатък на алгоритъма на Дийкстра е намирането на всичките минимални пътища от началния връх, а не само между началния и крайния. Това води до определено забавяне по време на търсенето. За да се намери търсеният път между началния и крайния връх, се налага извличане на пътя от масива parents. За графове, които не се променят за дълъг период от време, намирането на всички минимални пътища от началния връх може да повиши ефективността на търсене. Масивът parents съхранява върховете в ред, при който, ако трябва да се търси път между същия начален връх Vs до друг краен връх, може директно да се получат върховете, обхождайки масива, а не да се изпълнява отново целият алгоритъм.

2.2. Алгоритъм на най-кратък път с минимален брой възли

Летателният маршрут на летателния апарат се състои от точки, които са разположени в тримерното пространство. В общия случай те са разположени така, че образуват криви. Това се обуславя от два фактора: формата от обекта, който ще се изследва, и разположените в пространството препятствия, които трябва да бъдат заобиколени.

За да се изпълни движението по необходимата траектория, се налага летателният апарат да изпълнява маневри при преминаване от една точка към друга. Колкото повече точки има в избрания маршрут, толкова повече маневри ще изпълнява летателният апарат. Изпълняването на маневрите е свързано пряко с консумацията на енергия, тъй като трябва да се регулира скоростта на движение, което е свързано с регулиране оборотите на двигателите, задвижващи перките. Допълнително е възможно и да има завъртане на летателния апарат около оста си за позициониране спрямо обектите, които ще се изследват, а това води също до консумация на енергия. Поради тези съображения се приема, че по-малкият брой точки по време на полета намалява броя на маневрите. Това води до намиране на маршрут в графа, съдържащ минимален брой възли (точки). Малкият брой точки, съставящи пътя, е свързан и с по-малкия обем на изпращаната информация към летателния апарат.

Алгоритъмът за намиране на минимален брой възли се основава на алгоритъм за обхождане в ширина. При обхождането отново трябва да се запазват предшествениците, през които се преминава, за да се възстанови след това пътят между началния и краен връх.

Алгоритъм:

1. *Задава се начален връх Vs и се инициализират следните масиви:*

   - q[] - масив с върхове за обхождане, като първоначално Vs се добавя в списъка q[0] = Vs

   - UsedVertexes[] - масив за съхраняване на обходени върхове

   - parents[] – масив за съхраняване на предшествениците

2. *Докато в масива q[] има върхове, се изпълняват следните операции:*

   2.1. Извлича се връх от началото на опашката Vp

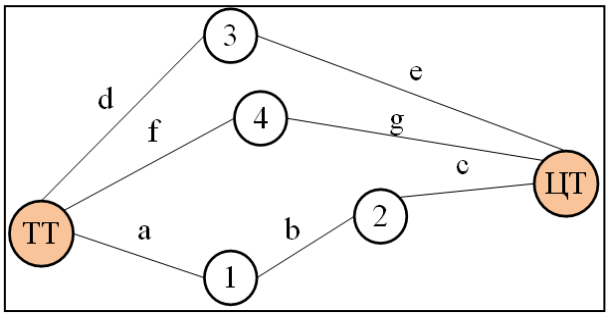
   2.2. За всеки Vi връх, който е наследник на Vp, се изпълняват следните операции:

   - *Ако връх Vi не е бил посещаван, се маркира като обходен в usedVertexes[]*
   - *Добавя се връх Vi в масива q[] за последващо обхождане*
   - *Връх Vp се добавя в масива на предшествениците parents[]*

При този алгоритъм върховете се обхождат по нива, докато се достигне последното ниво от върхове. След това се намира пътят между началния и краен връх, използвайки

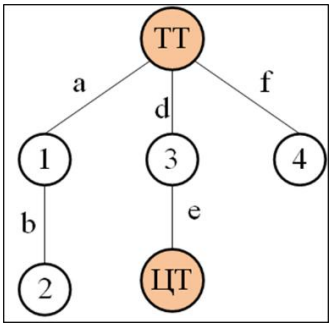
масива parents. Недостатък на алгоритъма е, че целият граф трябва да бъде обходен. Колкото повече нива на върхове има графът, толкова по-бавно се изпълнява търсенето. Друг недостатък на търсенето в този алгоритъм е, че, ако има няколко пътя между началния и краен връх, ще се намери само единият от възможните пътища. Той ще бъде този, започващ с по-малък номер от наследниците на предшественик.

На фигура 1 е представен примерен граф за илюстриране на минималните пътища между текуща точка (ТТ) и крайната (целева) точка (ЦТ), до която трябва да достигне.



Фиг. 1. Граф с примерни пътища между начална и крайна точка

На фигура 2 е представено обхождането в ширина на графа от фигура 1 при прилагане на алгоритъма по минимален брой възли, като връзките между върхове 2 и ЦТ и между 4 и ЦТ липсват, тъй като върховете вече са обходени и са маркирани като такива.



Фиг. 2. Обхождане на граф в ширина

При изпълнение на алгоритъма за намиране на най-кратък път по брой върхове между ТТ и ЦТ ще се намери най-кратък път с върхове ТТ – 3 – ЦТ. Пътят съдържа общо 3 върха, но според фигура 1 между върхове ТТ и ЦТ има още един път, минаващ през връх 4, който също ще има дължина от 3 върха. Дължината (d) на двата пътя съответно е (формули 1 и 2):

$$d1(ТТ, ЦТ) = d(ТТ, 3) + d(3, ЦТ) = d + e \quad (1)$$

$$d2(ТТ, ЦТ) = d(ТТ, 4) + d(4, ЦТ) = f + g \quad (2)$$

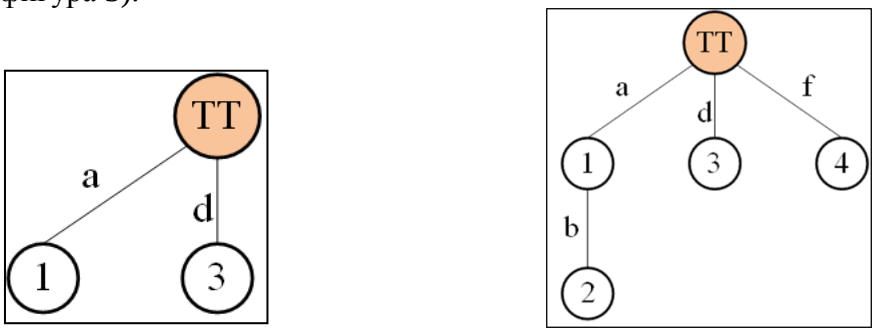
Двата пътя имат различна дължина, според представените пътища на фигура 1, като пътят с минимална дължина е $\min(d1, d2) = d2$, т. к. $(f + g) < (d + e)$.

От това следва, че по-краткият път е през връх 4, но алгоритъмът избира пътя през връх 3, тъй като той е с по-малък номер в графа.

2.3. Подобрен алгоритъм за намиране на най-кратък път по брой върхове

Алгоритъмът за намиране на най-кратък път отнема приблизително едно и също време за намиране на един връх, който се намира на ниво близо до началния връх и такъв, който се намира на ниво, което е отдалечено от него. Това се дължи на факта, че графът се обхожда винаги целият в ширина, след което се възстановява пътят между началния и краен връх.

При търсене на път между два върха се поражда идеята обхождането да продължава до момента, в който се достигне до крайния връх. След достигането му, вече има път до него от началния и обхождането може да се прекрати. Останалите върхове от графа не е необходимо да се обхождат (фигура 3).



А) Връх ТТ-връх 3 Б) Връх ТТ-връх 2

Фиг. 3. Обхождане на граф чрез предложен алгоритъм

Предложеният подобрен алгоритъм надгражда и доразвива алгоритъма за намиране на най-кратък път по брой върхове, като се внася допълнително условие за прекратяване на обхождането. Това условие намалява времето за търсене на крайния връх в зависимост от това на кое ниво на съседство се намира в графа.

Подобрен алгоритъм за намиране на най-кратък път по брой върхове:

1. *Задава се начален връх Vs и се инициализират следните масиви*
   - *q[] - масив с върхове за обхождане, като първоначално Vs се добавя в списъка q[0] = Vs*
   - *usedVertexes[] - масив за съхраняване на обходени върхове*
   - *parents[] – масив за съхраняване на предшествениците*
2. *Докато в масива q[] има върхове, се изпълняват следните операции:*

2.1. *Извлича се връх от началото на опашката Vp*

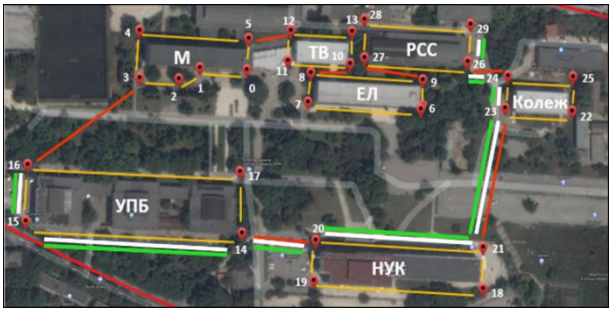
2.2. *За всеки Vi връх, който е наследник на Vp, се изпълняват следните операции:*
   - *Ако връх Vi не е бил посещаван, се маркира като обходен в usedVertexes[]*
   - *Добавя се връх Vi в масива q[] за последващо обхождане*
   - *Връх Vp се добавя в масива на предшествениците parents[]*
   - *Ако връх Vp съвпада с крайният връх Ve, обхождането се прекратява*

Предложеният алгоритъм търси крайния връх по нарастващ номер на съседите. Когато се достигне нивото на крайния връх, той се открива след проверка на всички възли, които имат по-малък номер от него в това ниво. Очаква се предимството на алгоритъма да е в намаляване на времето до намиране на крайния връх.

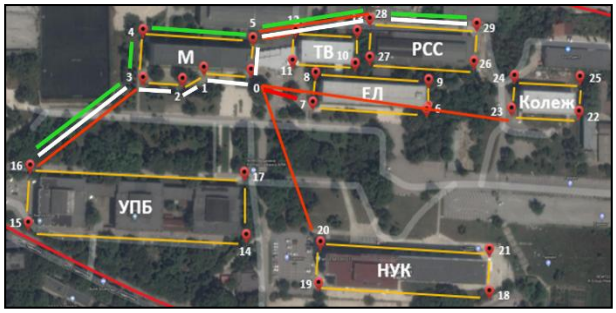
3. Сравнителен анализ на алгоритмите

3.1. Експериментален тест 1

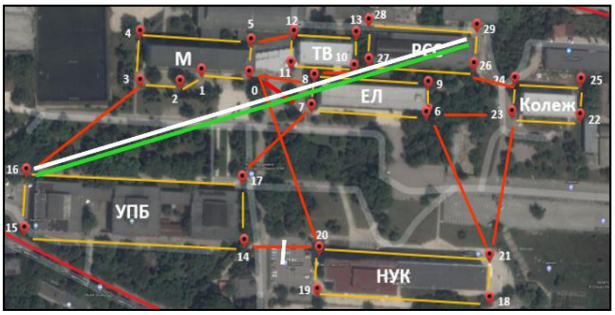
За първия експериментален тест са избрани начална точка 16 от сграда НУК и крайна точка 29 от сграда РСС. На фигура 4 са представени получените най-кратки пътища (маршрути) върху географската карта на района на ТУ-Варна за трите алгоритъма. Със зелена линия е показан маршрутът по алгоритъм на Дийкстра, а с бяла линия - другите два алгоритъма.



А) Топология „Кръг“



Б)Топология „Звезда“



В) Топология „Пълно свързване“

Фиг. 4. Маршрути между точки 16 и 29

В таблици 1, 2, 3 са представени маршрутите за трите топологии, техните общи дължини и времената за намирането им.

Таблица 1. Сравнение на маршрути между точки 16 и 29, получени от алгоритъм на Дийкстра

	Алгоритъм на Дийкстра			
Топология	**Възли (точки) в маршрут**	**Брой възли (точки)**	**Дължина на маршрута [м]**	**Време [us]**
Кръг	16 15 14 20 21 23 24 26 29	9	611.773	300
Звезда	16 3 2 1 0 5 28 29	8	417.248	216
Пълно свързване	16 29	2	380.479	293

Таблица 2. Сравнение на маршрути между точки 16 и 6, получени от алгоритъм с минимален брой възли

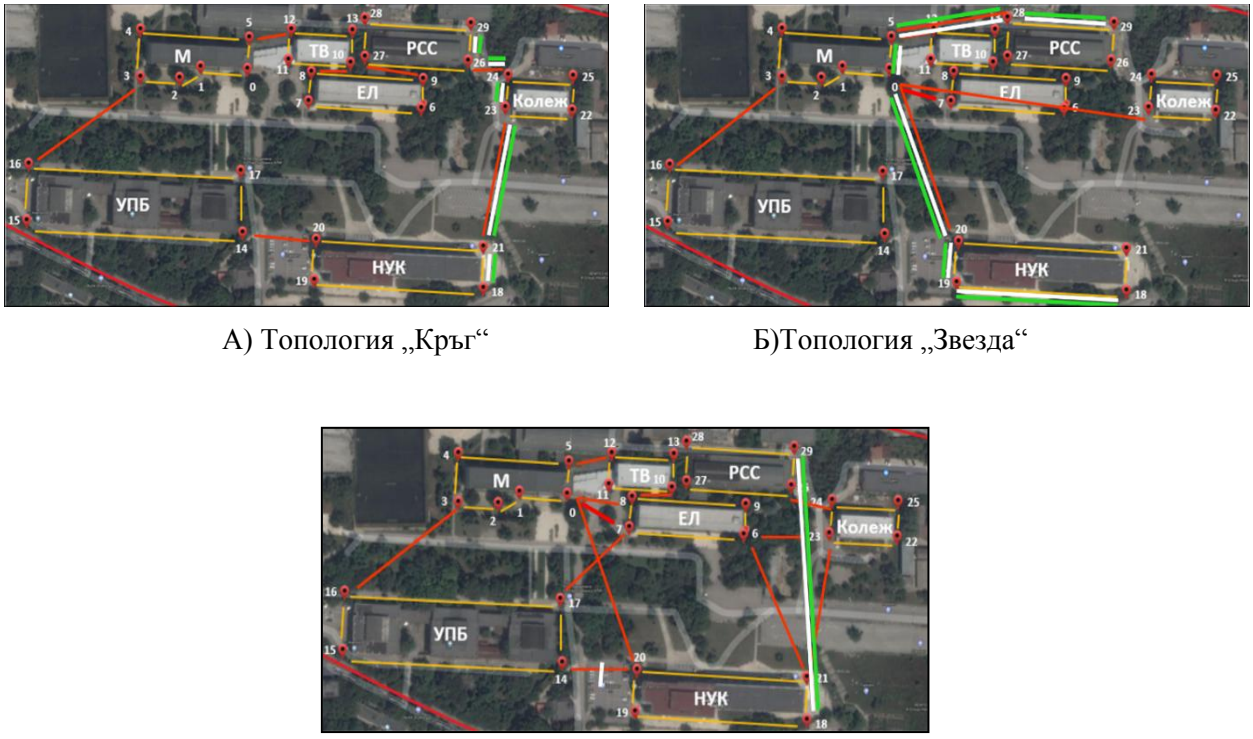
	Алгоритъм с минимален брой възли			
Топология	**Възли (точки) в маршрут**	**Брой възли (точки)**	**Дължина на маршрута [м]**	**Време [us]**
Кръг	16 15 14 20 21 23 24 26 29	9	611.773	170
Звезда	16 3 4 5 28 29	6	425.044	188
Пълно свързване	16 29	2	380.478	204

Таблица 3. Сравнение на маршрути между точки 16 и 29, получени от предложен алгоритъм

	Подобрен алгоритъм			
Топология	**Възли (точки) в маршрут**	**Брой възли (точки)**	**Дължина на маршрута [м]**	**Време [us]**
Кръг	16 15 14 20 21 23 24 26 29	9	611.773	142
Звезда	16 3 4 5 28 29	6	425.044	81
Пълно свързване	16 29	2	380.478	19

3.2. Експериментален тест 2

За втория експериментален тест са избрани начална точка 18 от сграда НУК и крайна точка 29 от сграда РСС. На фигура 5 са представени получените най-кратки пътища (маршрути) върху географската карта на района на ТУ-Варна за трите алгоритъма. Със зелена линия е показан маршрутът по алгоритъм на Дийкстра, а с бяла линия - другите два алгоритъма.



А) Топология „Кръг“ Б)Топология „Звезда“

В) Топология „Пълно свързване“

Фиг. 5. Маршрути между точки 18 и 29

В таблици 4, 5 и 6 са представени маршрутите за трите топологии, техните общи дължини и времената за намирането им.

Таблица 4. Сравнение на маршрути между точки 18 и 29, получени от алгоритъм на Дийкстра

	Алгоритъм на Дийкстра			
Топология	**Възли (точки) в маршрут**	**Брой възли (точки)**	**Дължина на маршрута [м]**	**Време [us]**
Кръг	18 21 23 24 26 29	6	228.899	216
Звезда	18 19 20 0 5 28 29	7	520.284	218
Пълно свързване	18 29	2	203.675	280

Таблица 5. Сравнение на маршрути между точки 18 и 29, получени от алгоритъм с минимален брой възли

	Алгоритъм с минимален брой възли			
Топология	**Възли (точки) в маршрут**	**Брой възли (точки)**	**Дължина на маршрута [м]**	**Време [us]**
Кръг	18 21 23 24 26 29	6	228.899	164
Звезда	18 19 20 0 5 28 29	7	520.284	168
Пълно свързване	18 29	2	203.675	204

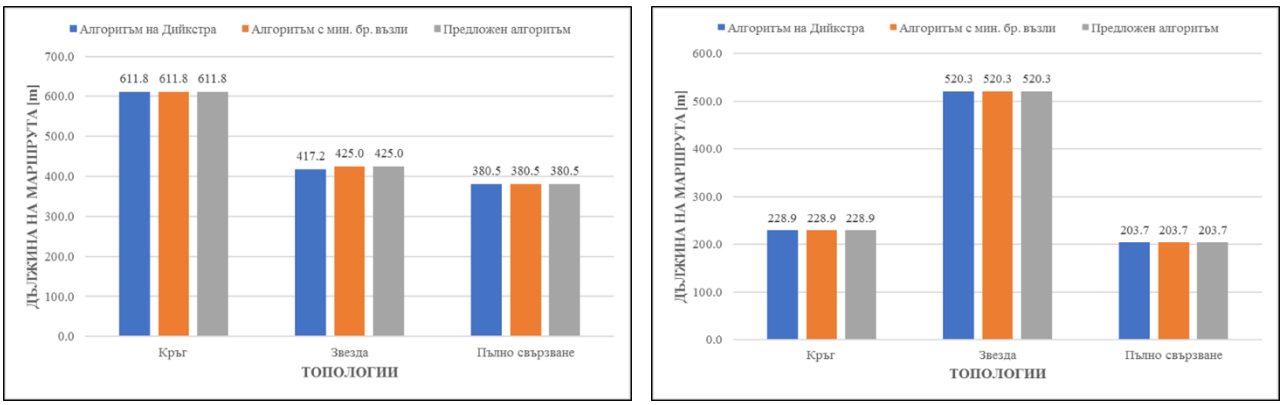
Таблица 6. Сравнение на маршрути между точки 18 и 29, получени от предложения алгоритъм

	Подобрен алгоритъм			
Топология	**Възли (точки) в маршрут**	**Брой възли (точки)**	**Дължина на маршрута [м]**	**Време [us]**
Кръг	18 21 23 24 26 29	6	228.899	69
Звезда	18 19 20 0 5 28 29	7	520.284	81
Пълно свързване	18 29	2	203.675	20

4. Експериментални резултати от сравнителния анализ

4.1. Резултати от сравнението по дължина на най-кратките пътища

На фигура 6 са представени сравнителни графики на дължините на намерените най-кратки пътища от трите алгоритъма.



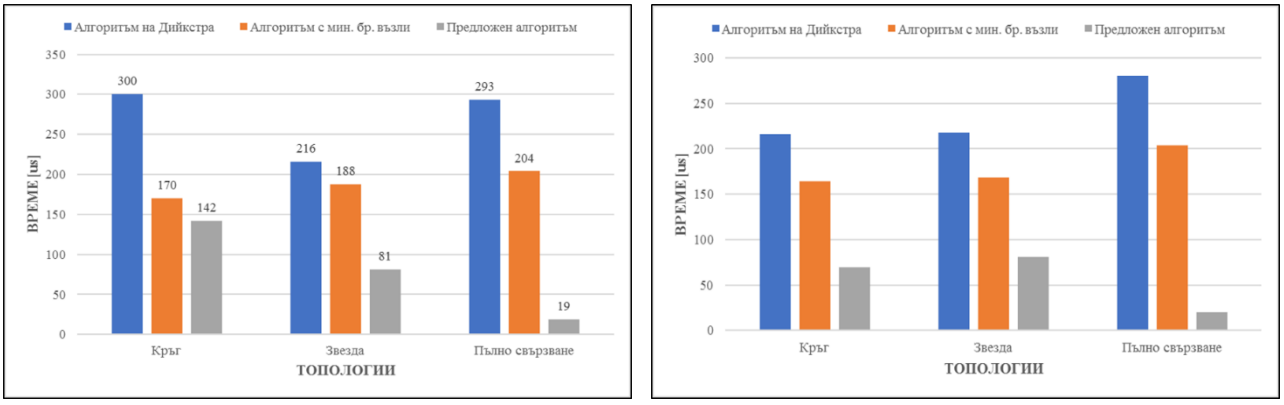
А) Между възли 16 и 29 Б) Между възли 18 и 29

Фиг. 6. Сравнителни графики на дължини на пътищата

Предложеният алгоритъм предоставя дължини на пътища, съвпадащи с тези на алгоритъма на минимален брой възли. В някои случаи предоставя пътища с по-малко на брой възли спрямо алгоритъма на Дейстра, въпреки малко по-голямата им дължина.

4.2. Резултати от сравнението по дължина на най-кратките пътища

На фигура 7 са представени сравнителни графики на времената на изпълнение на трите алгоритъма за експерименталните резултати от точка 3.



А) Между възли 16 и 29 Б) Между възли 18 и 29

Фиг. 7. Сравнителни графики на алгоритмите спрямо времето за изпълнение

Предложеният алгоритъм има време на изпълнение, зависещо от това къде се намира крайният връх като отдалеченост от началния по ниво. Той дава по-добри резултати по време на изпълнение спрямо останалите два алгоритъма.

5. Заключение

В представената статия се предлага подобрен алгоритъм за намиране на най-кратък път в предварително съставен маршрут на летене на безпилотен летателен апарат. Алгоритъмът се базира на алгоритъм на най-кратък път по минимален брой възли. Предложеният алгоритъм подобрява значително времето за намиране на търсения път, независисимо от текущата топология. Подобрението се основава на прекратяване на търсенето при откриване на крайния възел по време на обхождането на графа.

Откриването на маршрут с минимален брой възли осигурява възможност за намаляването на времето за обхождане на обектите в текущата топология, тъй като се намалява броят на маневрите в пространството (завъртане на летателния апарат, увеличаване/намаляване на скоростта при преход между отделните възли), като в същото време се увеличават участъците от пътя с линиейно придвижване.

Литература

[1] Nakov P.,The Programming=++Algorithms, 2013, Book

[2] Бойчев И., Изследване на алгоритми за обхождане на обекти, разположени в зона за сигурност, СБОРНИК ДОКЛАДИ Пета научна конференция с международно участие "Компютърни науки и технологии" ТУ-Варна, гр. Варна, том 1, ISSN 1312-3335, 2018 г., стр. 151-155

[3] Boychev I. Research algorithms to optimize the drone route used for security IEEE XXVII International Scientific Conference Electronics – ET, 2018, Sozopol, Bulgaria, 978-1-5386-6692-0

За контакти:
инж. Илиян Ж. Бойчев
катедра „Компютърни науки и технологии”
Технически университет - Варна
E-mail: i.boychev@tu-varna.bg

COMMUNICATION MODULE FOR CONTROL AND COLLECTION OF MICROCLIMATE DATA IN GLASSHOUSE

Yanislav V. Prokopiev, Todorka N. Georgieva, Plamena J. Edreva

Abstract: This article focuses on the development of a Micro greenhouse microclimate data collection module. A complete solution is presented to automate the data collection process. Data collected by the system is transmitted through a communication module to a server or displayed on a mobile device. This provides real-time updated information about the microclimate.
Keywords: Data, microclimate, module, communication

Introduction

Microclimate in the site and monitoring parameters. One of the main factors influencing the growth of plants is the microclimate. Monitoring and control of the basic parameters characterizing the production environment are foreseen [1]:

- Air temperature;
- Soil temperature;
- Air Humidity;
- Soil humidity;
- Light (illumination);
- The composition and the movement of air

The monitoring and control of the microclimate in the greenhouse are realized through a flexible system consisting of hardware part, sensors, controller, actuators and software part. Sensors for temperature, humidity, CO2, light and, respectively, external sensors for wind speed and direction, solar radiation, rain, outdoor temperature and humidity are used. In the program of the controller are set requirements for humidity and temperature of air, humidity and temperature of soil, solar radiation and other parameters of microclimate. One day can be divided into several time parts, each with the appropriate parameters [2]. The control is performed according to the settings of the parameters for the whole day.

Exposition

Communication module for control and data collection

The block diagram of the communication module, consisting of controller, sensors, power steering Relay-Contact module, and communication module with integrated antenna [3] is presented in figure 1.

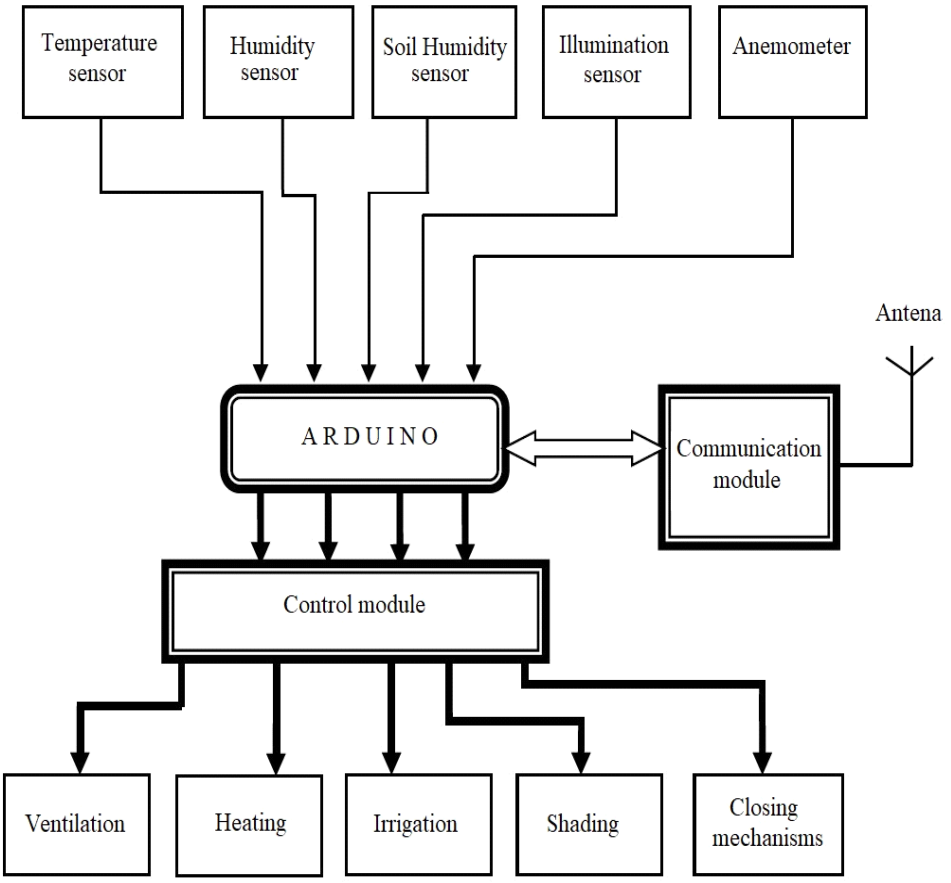


Fig. 1. Communication module for control and data collection

The use of a power control module is dictated by the fact that the microcontroller module has very low levels of output control signals (both in voltage and maximum current). The relay contactor control unit, the schematic diagram of which is shown in Fig. 2, allows the connection of powerful actuators to the microcontroller. These actuators can be supplied with different voltages

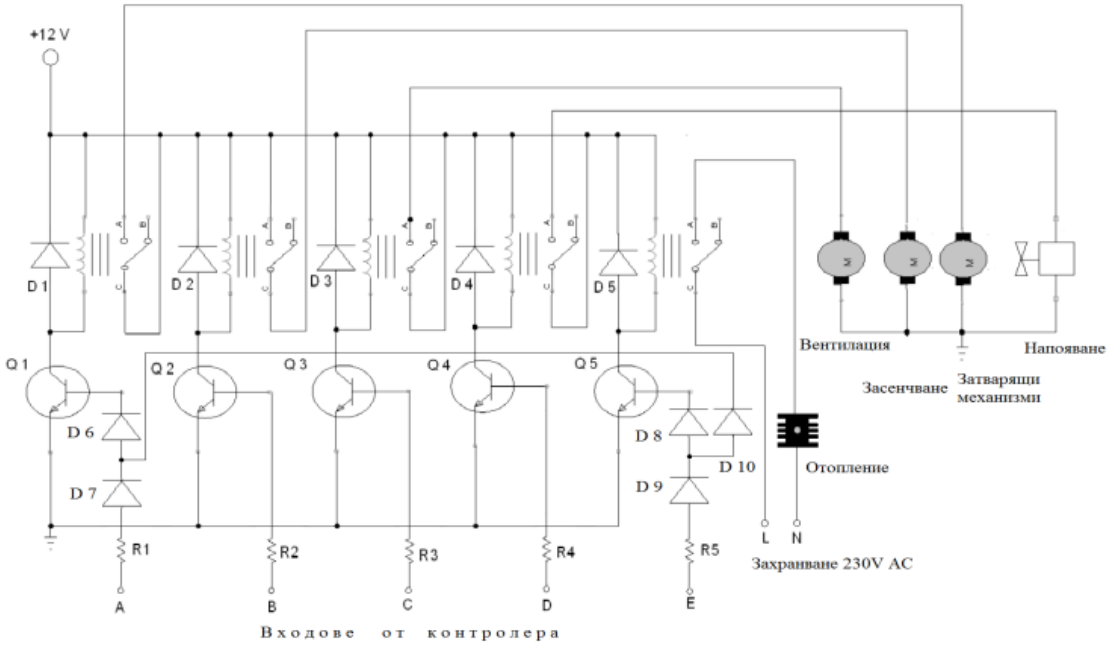


Fig. 2. Schematic diagram of the power control module

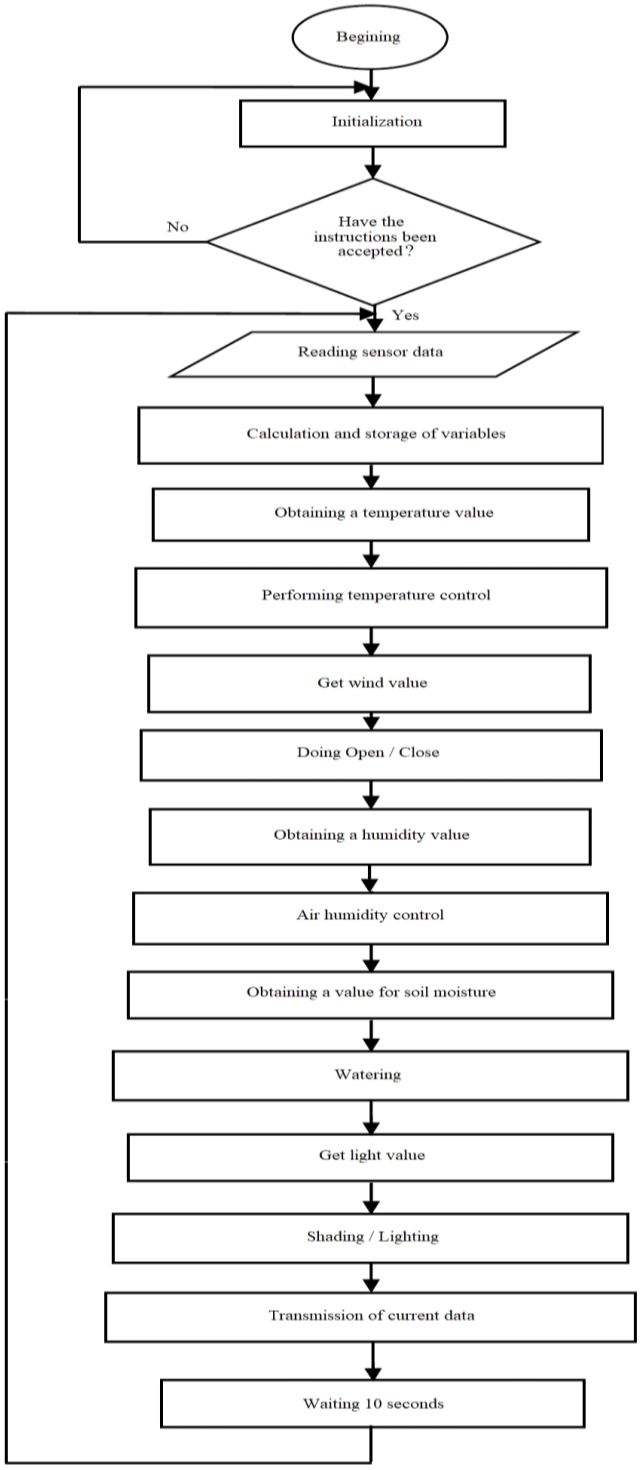


Fig. 3. Data collection algorithm

The solution offered is with high reliability and practical realization. A programming algorithm is proposed for the switching controller to function. Block diagram of the loop is shown in Fig. 3.

The program of this cycle also relies on the exchange of data with the communication module and their subsequent transfer to the Internet [4]. Data can be collected in a cloud-based server, for example, and sent for visualization on a user mobile device. Before this, however, they must be collected, processed and the system generates some reaction depending on the current values of the environmental parameters.

The cycle for capturing the current temperature and the system response, according to this parameter is presented in Fig. 4. Depending on the instantaneous values recorded and the threshold levels set out in the programme, it may include or exclude heating systems or ventilation in order to maintain the optimum microclimate in the steam room.

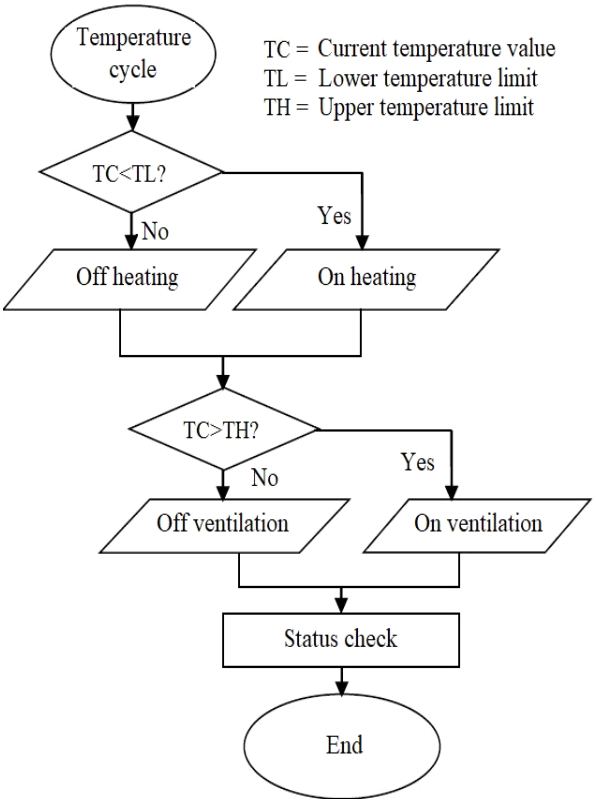


Fig. 4. The current temperature cycle and the system response depending on this parameter

An algorithm for controlling air humidity is proposed on Fig. 5.

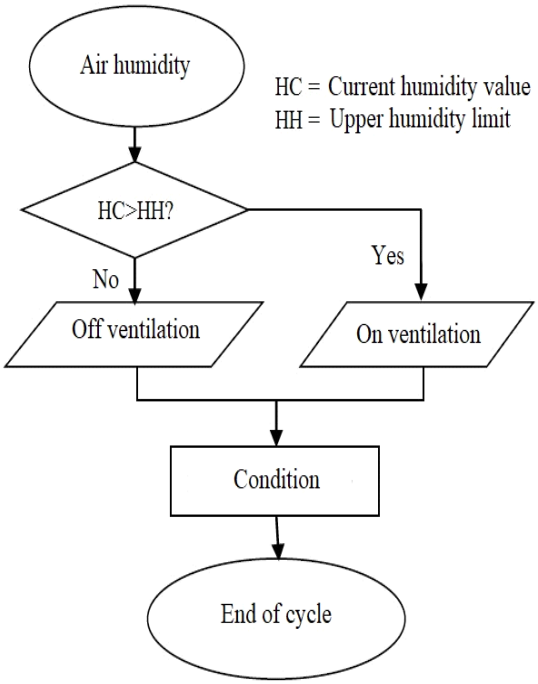


Fig. 5 Air Humidity Control algorithm

In this cycle, a system reaction is foreseen by triggering an actuator (ventilation) [5]. Avoiding conflicts in the controller can be ensured by using appropriate programming code.

The algorithm shown in Fig.6 illustrates the cycle for monitoring and managing soil moisture.

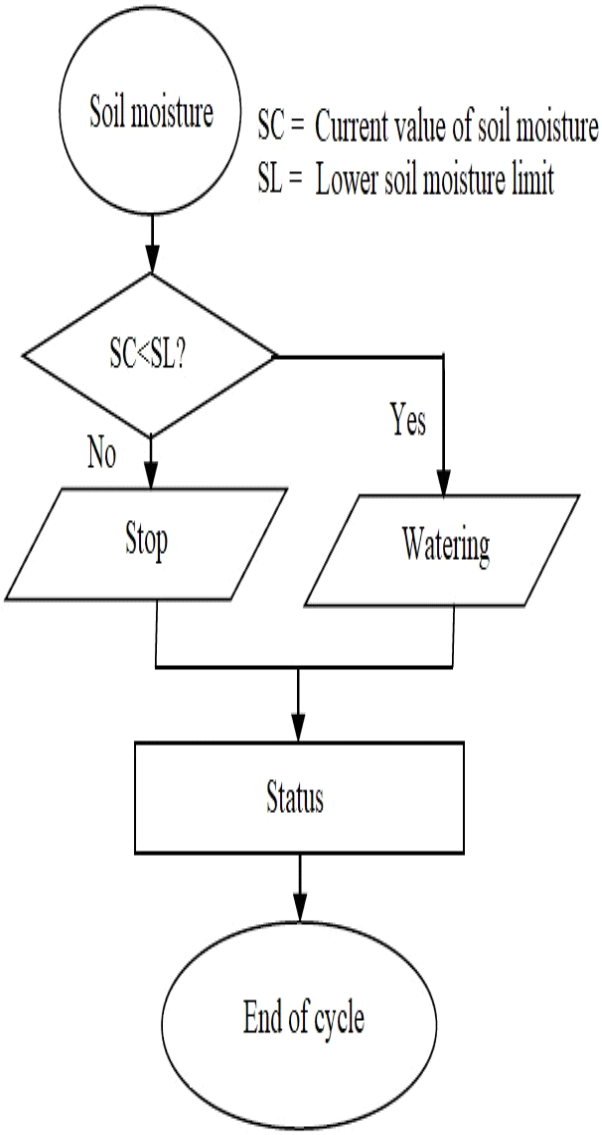


Fig. 6 Soil moisture monitoring and control

The operation of the irrigation system and the increase in soil moisture will inevitably lead to an increase in the amount of water evaporated from the leaves of the plants (especially at higher temperatures) and from there to an increase in humidity [6]. This is just one example of the complex impact of activating or stopping each of the actuators.

A diagram of the wind speed monitoring cycle is presented on fig. 7. Depending on its value, the system is monitored by closing or opening the ceiling windows, taking into account the temperature in the room. This facilitates the operation of ventilation, prevents the achievement of extreme levels of humidity and the development of pathogens on plants, while at the same time observing the energy efficiency of the system.

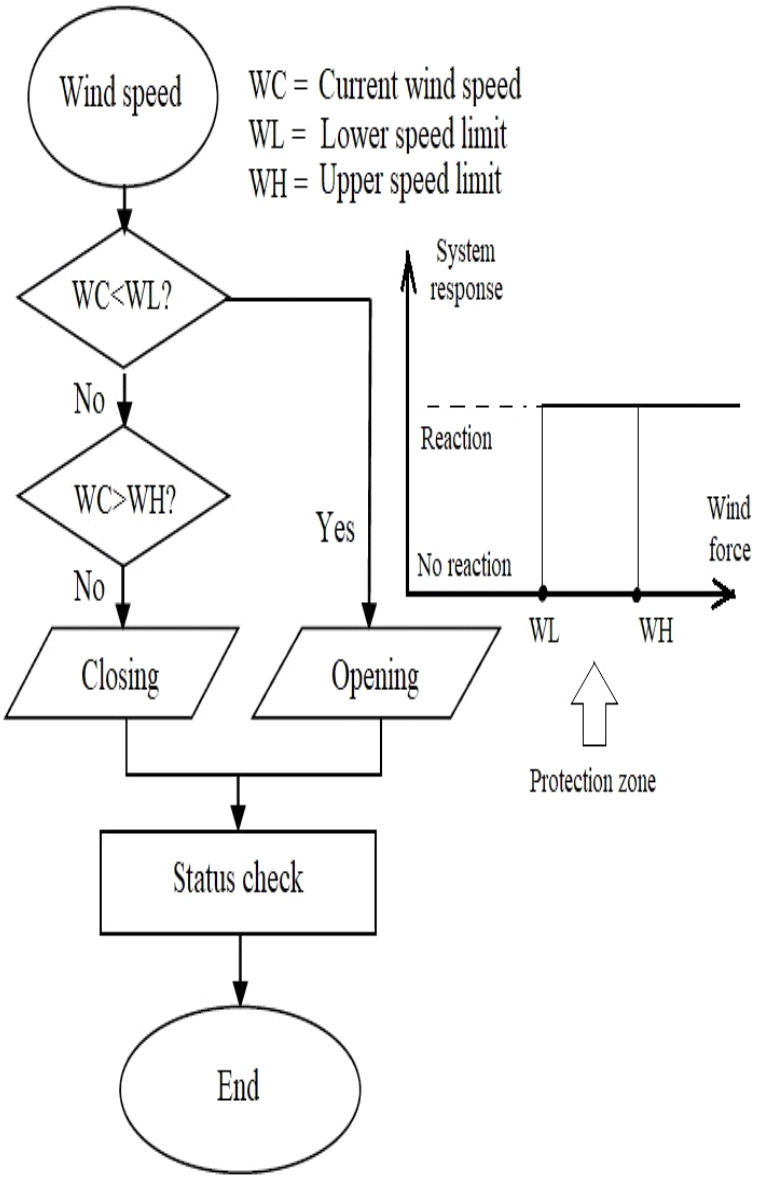


Fig. 7. The wind-monitoring cycle diagram

With the setting in the programming cycle of a lower lower limit and a higher upper limit of the permissible wind speed, the hysteresis is ensured when the ceiling closing mechanisms are activated. Thus, their premature depreciation is prevented and further limits the power consumption of the system [7] [8].

Additional hardware protection against opening of the skylights during operation of the heating system is provided by the diode logic in the base circuits of the transistors Q1 and Q5, controlling the relays of the closing mechanisms and the heating system, respectively.

On Fig. 8 can also be seen the sub-cycle for checking the intensity of illumination and providing shade in very strong sunshine. In this cycle, in addition to the maximum allowable level, there is also an intermediate level, and the goal is again to provide a zone without system response.

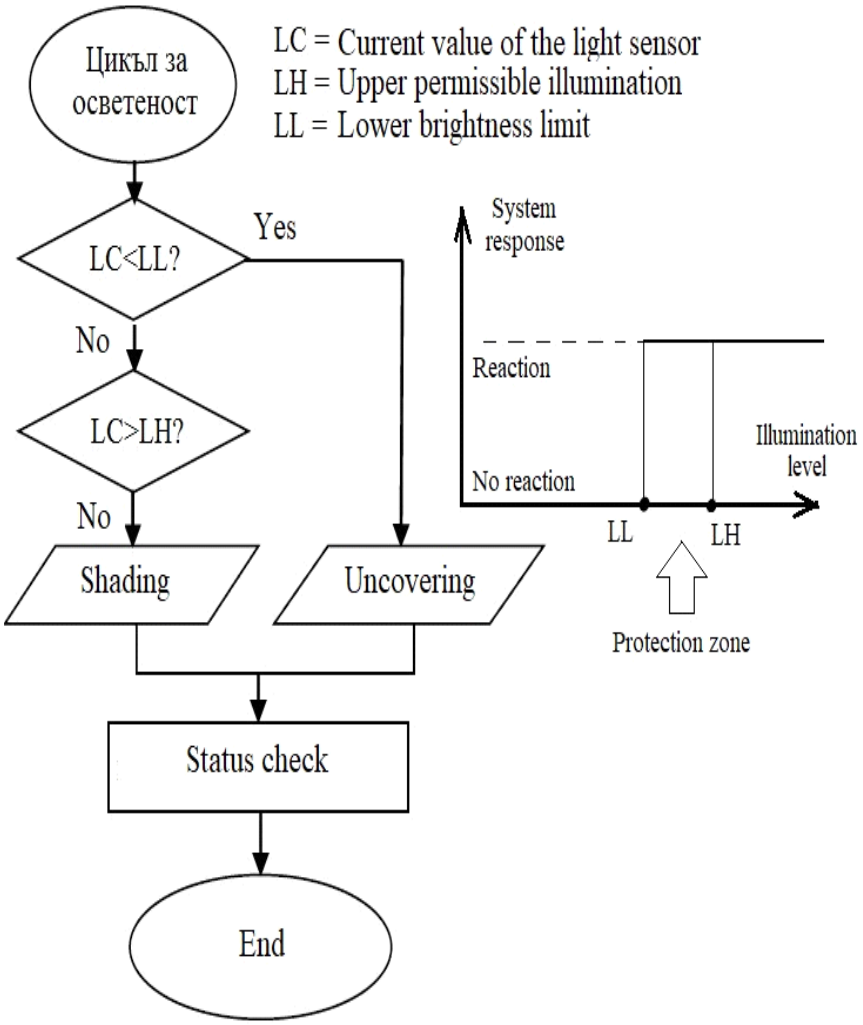


Fig. 8 Check of illumination and shading

Considerations in the introduction of hysteresis here are related to avoiding the frequent operation of the shading mechanisms in the event of momentary clouds and extending their life cycle, and this measure also leads to limiting the consumption of electricity from the system[9], [10].

In this cycle, in addition to the maximum permissible level, an intermediate level is stipulated with the objective of providing a zone without reaction

Conclusion

The proposed system and management algorithms eliminate the risk of human error in manual monitoring and maintaining the parameters of the environment within certain limits. The low cost and minimum own system consumption has been achieved. The survey and the made selection of the components, as well as the compilation of the controller's program algorithm guarantee reliability and applicability. At the same time, functionality, flexibility, upgrade capabilities and connectivity of the system are preserved.

Developed system AS, as proposed in this project, provides full functionality and is suitable for integration into SMALL MODULAR TYPE greenhouses GARDEN AND IN THE GREENHOUSE large industrial farms.

The system is flexible and can relatively easy and without major ADDITIONAL COSTS TO BE perfection and build on the specific needs of different manufacturers (e.g., keeping of exotic

animals required under the specific environmental conditions) with extra sensors and actuators if necessary.

References

[1]. Ponce P., Greenhouse Design and Control, 1st Edition, Taylor & Francis Group, London 2015

[2]. Iskren Nikolov, Development of a warehouse microclimate management system, Innovation and entrepreneurship, ISSN 1314-9253 ,Volume VI, number 3, 2018.

[3]. Health Canada, Environmental and Workspace Health: Residential Indoor Air Quality Guidelines, 2015

[4]. Vu Quan, Minh Automated Wireless Greenhouse Management System, 2011

[5]. Компютърно базирана система за мониторинг на параметри на въздушната среда в затворени помещения, Звездица Ненова, Георги Георгиев, Стефан Иванов, Научни трудове на русенския университет - 2015, том 54, серия 3.1

[6]. Air conditioning of production premises. Engineering review, vol.4, 2012, http://www.engineering- review.bg/bg/klimatizaciya-na-proizvodstveni-pomeshteniya/2/1932/ 2018, Bulgarian

[7]. Andonov, K., P. Daskalov, K. Martev. (2003). A new approach to controlled natural ventilation of livestock buildings. Biosystems Engineering, vol.84, pp. 91-100.

[8]. Binev, I. (2018). Possibilities for ground water use for conditioning of the faculty of technics and technologies building in Yambol. Applied Researches in Technics, Technologies and Education, Vol. 6, No. 2, pp.166-169.

[9]. Kartelov, Y., N. Katrandzhiev. (2016). Power Supply of Resource-Sensitive Smart Devices by Energy Management and Energy Collection. Scientific Works of the University of Food Technologies – Plovdiv (in Bulgarian)

[10]. Zlatev, Z. (2017). Analysis of data from automatic weather stations. Innovation and entrepreneurship, vol.5, No.4, pp.216-230

Contact details:
Associated prof. eng. Todorka Georgieva
Department of Communication Engineering and Technology
Technical University of Varna
e-mail: tedi_ng@mail.bg

RADIOWAVE PROPAGATION OVER SEA – MODELS AND DUCTS ASSESSMENT OVERVIEW

Nikolay Grozev

Abstract: The aim of this paper is to make a brief review of the models for radio wave propagation over sea surface developed and used so far and assessment of ducts occurrences especially for the Black Sea region. This article is not intended to cover every work published on this topic, because the subject is quite long and extensive, but rather to cover the main approaches and factors determining the radio waves propagation over sea surface. It is limited to reviewing scientific approaches mostly in Very high frequency and Ultra high frequency ranges. The Parabolic equation model and his numerical scheme Split-step Fourier Transform is most universal and used method. It is more capable than other models in computing loss, due to varying refractivity and terrain along the propagation path and has larger step size.
Keywords: duct models; duct measurements; duct occurrence; radio wave propagation over sea; tropospheric duct.

1. Introduction

The problems of tropospheric radio wave propagation over sea surface were investigated from many researchers over the years, because of impossibility of establishing stable radio communications links or various interferences caused by the abnormal spread of radio waves. The different weather conditions, occurring in the atmosphere over large water basins are the main reason for that phenomenon. Due to these constantly varying and different atmospheric conditions depending on the water basin, climate zone or season of the year, it is very difficult to define the most accurate model to describe radio waves propagation over sea. The water condition also has a significant influence of radio waves propagation. It is well known, that different radio waves have different propagation through the atmosphere, which makes the task of developing a reliable and accurate model more complex.

For standard atmosphere temperature, moisture and pressure decrease with altitude, which do not have a large impact on the refraction, but as an exception to this occurs where rapid change in temperature take place over relatively small change in altitude (temperature inversion). Also an increase in water vapor pressure with 1 milibar causes an increase in refractivity with 1 to 5 ratios. Pressure has a small effect. There are four main types of refraction *N*: Sub-refraction, Normal refraction, Super Refraction (positive refraction) and Trapping a special case of Super Refraction, as the weather conditions are the same but more intense. When *N* falls below the Critical Gradient, the radio waves bend in the direction of the Earth's surface, reflecting it or falling into a negative refraction layer and distorted upward into the atmosphere. This process can lead to a significant increase in the range of radio waves and can confine EM waves to a narrow height range called a duct. There are four main types of ducts: surface, surface-based, elevated and evaporation ducts. When trapping conditions are extending to the surface, the duct is called Surface duct. Surface-based duct is formed when refractivity of the surface is greater than the minimum at the trapping layer, keeping it to the surface and the last is slightly elevated. These ducts are formed when air at height is wormer and dryer then those at the surface. The Elevated Duct is differing from surface ducts only from that the base of the duct is aloft. Evaporation duct is formed because the air in immediate contact with the sea/ocean surface has a relative humidity of 100%. Water Vapor Pressure rapidly decreases with height in the few meters above the surface. Thus, the N values increase with height to the point where it again begins to decrease. The height of the maximum is called Evaporation Duct Height, which varies from 1m to 40 m at equatorial latitude summer days. Evaporation Duct is much weaker then Surface Duct. For near sea-surface LoS propagation, the

effects of evaporation duct are obvious dominant, because the Evaporation Duct exists over the ocean almost all of the time [1]. When the air temperature is higher than the temperature of water, the atmosphere is in stable condition and vice versa the atmosphere is unstable.

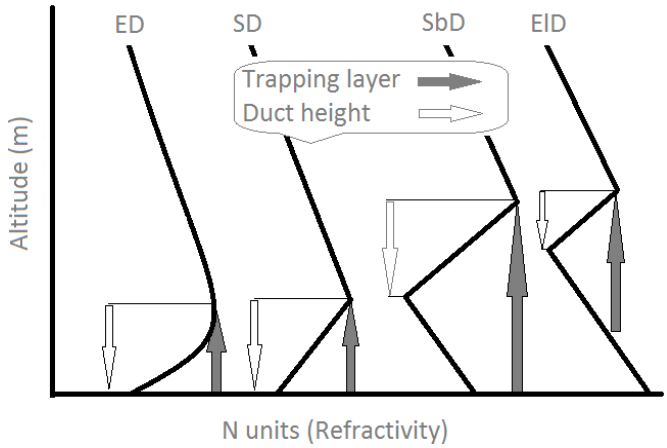


Fig. 1. Four main duct: ED - Evaporation duct; SD – Surface duct; SbD – Surface based duct; ElD-Elevated duct

Reflection and refraction of radio wave can result in changes in its polarization, so that changes in polarization can cause changes in the received signal strength. When propagation is accomplished via multihop refraction, energy is lost each time the radio wave is reflected from the Earth's surface. The amount of energy lost depends on the frequency of the wave, the angle of incidence, ground irregularities, and the electrical conductivity of the point of reflection.

As a general it can be said that normally, radio waves have major loss of energy caused by the spreading of the wave front as it travels away from the transmitter. The propagating signals in the atmospheric ducts are trapped between the ducting layers and/or the sea surface, so that the power of the propagating signal does not spread isotopically through the atmosphere. As a result, these signals travel over the horizon [2]. Interference may arise through a range of propagation mechanisms whose individual dominance depends on climate, radio frequency, time percentage of interest, distance and path topography.

To get enough information of what work has been done so far, a survey article is proposed that gathers information about various propagation models used already and proposed for maritime environment, statistics for duct height and occurrences especially for the region of Black sea coast.

2. Channel models for radio wave propagation

2.1. Parabolic equation method

The dominant technique to exemplary radio waves propagation in the troposphere regardless of the frequency ranges used and the territories over which the radio waves are propagate is Parabolic equation method (PEM). This model use paraxial approximation to the Helmholtz wave equation. The two-dimensional scalar parabolic wave equation in terms of u (scalar wave equation with reduced function associated with the paraxial direction x) is:

$$\frac{\partial u}{\partial x^2} + 2ik\frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial z^2} + k^2\left(n^2 - 1\right)u = 0, \tag{1}$$

where *u (x,z)* is the wave function of the electric or magnetic field; *k* is the wave number in vacuum; *n* is the atmospheric refractive index. Utilizing first order Taylor expansions of the square-root and exponential functions and some operations (1) can be approximated to standard parabolic equation [3]:

$$\frac{\partial u(x,z)}{\partial x} = \frac{ik}{2}\left[\frac{1}{k^2}\frac{\partial^2}{\partial z^2} + \left(n^2(x,z) - 1\right)\right]u(x,z). \tag{2}$$

The modelling of PEM is approached using different numerical schemes that include Split-step Fourier Transform (SSFM), Finite Element and Finite Difference methods. There are a lot of papers which have explained the theory of PEM and SSFM (as [3]) in detail and the interested reader may refer to them. However, we do not discuss the theory of PEM in this paper. Instead, we only make an observation of the basic numerical methods for solving.

2.1.1. Split-step Fourier Transform based on Parabolic equation

The split-step solution of the narrow-angle parabolic equation is given by [3]:

$$u\left(x+\Delta x, z\right) = e^{\frac{i\Delta x k(n^2-1)}{2}} F^{-1}\left\{e^{-\frac{in^2\Delta x p^2}{2k}} F\left[u(x,z)\right]\right\}, \tag{3}$$

where F and F^{-1} represents the forward and inverse Fourier transform, and $p = k\sin\theta$ is the transform variable (θ is the propagation angle, referenced from the horizontal). The wide-angle split-step solution is computed by [4]:

$$u\left(x+\Delta x, z\right) = e^{i\Delta x k(n-1)} F^{-1}\left\{e^{-i\Delta x p^2\left(\sqrt{k^2+p^2}+k\right)^{-1}} F\left[u(x,z)\right]\right\}. \tag{4}$$

2.1.2. Finite-Difference method based on Parabolic equation

Finite-Difference method is obtained by numerically solving (2) with the Crank-Nicolson schema, using rectangular grid [5]. The differentiations in (2) are present as:

$$\frac{\partial u_{i+1/2}^m}{\partial x} = \frac{u_{i+1}^m - u_i^m}{\Delta x}, \tag{5a}$$

$$\frac{\partial^2 u_{i+1/2}^m}{\partial z^2} = \frac{u_i^{m-1} - 2u_i^m + u_i^{m+1} + u_{i+1}^{m-1} - 2u_{i+1}^m + u_{i+1}^{m+1}}{2\Delta z^2}, \tag{5b}$$

$$u_{i+1/2}^m = \frac{u_i^m + u_{i+1}^m}{2}. \tag{5c}$$

The calculation is performed as a field in the next step $i+1$ subscript (on a vertical) is computed from the field in the previous step *i* with given boundary conditions at the bottom and top of the computational domain and with given field distribution at the initial vertical plane.

Substituting the above equations in (2) we obtain:

$$u_{i+1}^{m-1} + (-2 - a + b_i^m)u_{i+1}^m + u_{i+1}^{m+1} = -u_i^{m-1} + (2 - a - b_i^m)u_i^m - u_i^{m+1}, \tag{6}$$

where:

$$a = \frac{4ik\Delta z^2}{\Delta x}, \tag{7a}$$

$$b_i^k = k^2(n_{i+1/2}^m - 1)\Delta z^2, \tag{7b}$$

where Δx and Δz are step sizes in horizontal and vertical directions respectively.

2.1.3. The Finite-Element method based on Parabolic equation

The method is obtained by first multiplying equation (2) with a smooth test function v(x), while considering the boundary condition in:

$$\left(a_1 \frac{\partial}{\partial z} + a_2 \right) u(x,z) \Big|_{z=0} = 0 \ , \tag{8}$$

where a_1 and a_2 are constants, and $a_1 = 0 \ \ (i = 1,2)$ results in Dirichlet (horizontal polarization) and Neumann (vertical polarization) boundary conditions, respectively, for a perfectly electrically conducting surface. The Cauchy-type boundary condition is introduced with $a_1 = 1, \ a_2 = ik\sqrt{\gamma - 1}$ for the horizontal, and $a_1 = 1, \ a_2 = ik\sqrt{\gamma - 1}/\gamma$ for vertical polarization. $\gamma = \varepsilon_r + i60\sigma\lambda$ is the relative complex dielectric constant in terms of the relative permittivity (ε_r) and conductivity (σ) of the ground at range x. Integrating from $z = 0$ to $z = Z_{\max}$, and integration-by-parts rule [6] gives:

$$\begin{aligned} \int_0^{Z_{\max}} \left\{ -\frac{\partial u(x,z)}{\partial z} \frac{dv(z)}{\partial z} + k^2 \left[n^2(z) - 1 \right] u(x,z) v(z) + 2ik \frac{\partial u(x,z)}{\partial x} v(z) \right\} dz \\ + \frac{\partial u(Z_{\max})}{\partial z} v(Z_{\max}) - \frac{\partial u(0)}{\partial z} v(0) = 0. \end{aligned} \tag{9}$$

The last two terms of Equation (9) have no contributions for the homogeneous Dirichlet and Neumann boundary conditions must, but only for the Cauchy boundary conditions. Dividing the domain vertical profile $[0 - Z_{\max}]$ into subdomains (called elements) in which the basic functions are generally formed with the help of linear piecewise Lagrange polynomials:

$$B_1^e(z) = \frac{z_2^e - z}{z_2^e - z_1^e}, \tag{10a}$$

$$B_2^e(z) = \frac{z - z_1^e}{z_2^e - z_1^e}, \tag{10b}$$

where e stands for the elements between z_1^e and z_2^e. Approximated the field function $u(x,z)$ by $u_{ap}(x,z)$ at the discrete height points as:

$$u_{ap}(x,z) = \sum_{e=1}^{n_e} \sum_{j=1}^{2} c_j^e(x) B_j^e(z), \tag{11}$$

where e is the number of elements and $c_j^e x$ denotes the coefficients of unknown functions. By replacing test function with $B_m(z)$ for $m = 1,2$, equation (9) can be written as:

$$\begin{aligned} \sum_{e=1}^{n_e} \left\{ \sum_{j=1}^{2} c_j^e \int_{z_1^e}^{z_2^e} \frac{\partial B_j^e}{\partial z} \frac{\partial B_m^e}{\partial z} dz - k^2 c_j^e \int_{z_1^e}^{z_2^e} \left[n^2(z) - 1 \right] B_j^e B_m^e dz \right. \\ \left. -2ik \sum_{j=1}^{2} \frac{\partial c_j^e}{\partial x} \int_{z_1^e}^{z_2^e} B_j^e B_m^e dz = 0. \right. \end{aligned} \tag{12}$$

The conventional Finite Element/Difference methods are less accurate, have smaller step size and computationally more expensive, but permit more flexibility in the implementation of various boundary conditions. Split-step Fourier Transform has larger step size and computationally very efficient.

2.2. Two and tree ray path loss model

For the radio wave propagation in free space, the propagation loss can be predicted by the FSL model:

$$L_{FSL} = 20\log\left(\frac{4\pi d}{\lambda}\right) - PF, \tag{13}$$

where L_{FSL} is the free space loss in dB, λ is the wave length, and *d* is the propagation distance in meters and *PF* is propagation factor.

The needs to fit the local oscillations resulted from the destructive summation of sparse multipath signals the ray trajectory-based path loss models is present. This model can geometrically identify the trajectories of the most dominant rays arrived at the receiver, thus the phase shift of each ray is characterized and considered in the path loss calculation, therefore providing a better description of the local peaks and nulls of the received signal strength. When a reflected ray from the sea surface exists besides the LoS path (direct ray), the propagation loss could be predicted by a 2-ray path loss model. Therefore, the 2-ray path loss model can be simplified as [7]:

$$L_{(h_t,h_r)} = -10\log_{10}\left(\left(\frac{\lambda}{4\pi \mathrm{d}}\right)^2\left(2\sin\left(\frac{2\pi}{\lambda}\frac{h_t h_r}{d}\right)\right)^2\right), \tag{14}$$

where *L* is the 2-ray propagation loss in dB, and h_t, h_r are the heights of a transmitter and a receiver in meters. If a homogeneous evaporation duct layer exists pathlosss can be expressed as a simplified three-dimensional model [7]:

$$L_{(h_t,h_r,h_e)} = -10\log_{10}\left(\left(\frac{\lambda}{4\pi \mathrm{d}}\right)^2\left(2\left(1+\Delta\right)\right)^2\right), \tag{15}$$

where:

$$\Delta = 2\sin\left(\frac{2\pi h_t r}{\lambda d}\right)\sin\left(\frac{2\pi\left(h_e - h_t\right)\left(h_e - h_r\right)}{\lambda d}\right), \tag{16}$$

2.3. Ray optics model

The RO method consists of tracing a series of direct and reflected rays through selected control points, and then interpolating the magnitudes of these rays (computed from a spreading term relative to free space spreading for each ray) and the phase angle between (determined from the optical path length differences from the ground range for each ray) them at each desired receiver point. The total range *X*, spherical spreading *S*, and total optical path length difference *D* are computed at each ray trace step *i*. Then, they are summed over the whole ray path for each ray once the desired receiver point is reached and are given by [8]:

$$X = \sum_i (a_1 - a_0) g_i^{-1}, S = \sum_i \left(\frac{a}{a_1} - \frac{a}{a_0} \right) g_i^{-1},$$
$$D = \sum_i \left[\left(10^{-6} M_i - \frac{1}{2} a_0^2 \right) (a_1 - a_0) + \frac{1}{3} \left(a_1^3 - a_0^3 \right) \right] g_i^{-1}, \quad (17)$$

where a is the elevation angle at the transmitter, and a_0 and a_1 are the angles at the beginning and at the end, respectively, of each ray trace step and:

$$g_i = \frac{10^{-6} \left(M_{i+1} - M_i \right)}{z_{i+1} - z_i}. \quad (18)$$

These are obtained by the standard geometric ray trace formulas based on small angle approximations to Snell's law. Once X, S, and D are computed for each ray, the propagation factor for the direct ray F_d and reflected rays F_r, are given by:

$$F_d^2 = f_d^2 \left| \frac{X}{\beta_d S_d} \right|, F_r^2 = f_r^2 R^2 \left| \frac{X}{\beta_r S_r} \right|, \Omega = \left(D_r - D_d \right) k + \varphi, \quad (19)$$

where f_d and f_r are the antenna pattern factors for the direct and reflected ray elevation angles, respectively, at the transmitter, β_d and β_r are the propagation angles at the receiver point for the direct and reflected rays, R and φ are the magnitude and phase lag of the Fresnel reflection coefficient, and Ω is the total phase angle between the direct and reflected rays. The propagation factor F of the resultant field between both rays is:

$$F^2 = F_d^2 + F_r^2 + 2 F_d F_r \cos \Omega \quad (20)$$

RO model assumes a horizontally homogeneous refractive environment and should be used over a flat surface in regions with flat terrain profiles within the first few kilometers from the source. For non-flat terrain profiles, produced coverage diagrams are limited and loss will not be computed at large heights and near ranges.

2.4. C. Advanced propagation model (AMP)

APM is a hybrid model, consisting of four basic sub models: flat earth (FE), ray optics (RO), extended optics (XO), and the split-step Fourier PE algorithm. The primary model is the PE and the other three sub models are built around. The PE model is more capable than the other three in computing loss due to varying refractivity and terrain along the propagation path. Therefore, all parameter constraints and initializations are performed for the PE algorithm first, keeping the region over which it is applied to a minimum for the most efficiency. APM used other three models to compute loss in regions where the PE algorithm is not applied [8].

3. Examination of the atmosphere to assess the Tropospheric ducts

The one year radiosonde data are used in [9] to investigate the occurrence of elevated ducts above Arabian Sea. The results showed that in the lower atmosphere (100 m to 750 m) 6.82 % of time duct have occurred. The anomalous propagation due to ducting is more prevalent during spring season. Another 2 years radiosonde data are used in [10] to determine the surface duct conditions over Istanbul, Turkey. In that region most of the ducts are occurred in May (14%) and July (13%). The highest occurrence rate of surface ducts is observed in the summer season (33%) and the lowest

- in the winter season (17%). The median duct thickness (in meters) is found to be the highest in summer (35 m), whereas the lowest in winter (24 m), also the extreme duct thicknesses are 400 m observed in July. After separating the data into stable and unstable atmospheric subgroups, a clear seasonal differences of surface duct characteristics is observed. The percentage occurrence of surface ducts in the stable and unstable atmosphere is 51% and 46%, respectively. The results also showed that summer ducts is caused more by local effects and synoptic-scale weather systems mostly affect the surface ducts in winter. Surface ducts in a stable atmosphere are found to be stronger than those in an unstable atmosphere. Reported results on the tropospheric ducts occurrence and properties for the bays of Varna and Burgas situated along the Bulgarian Black Sea shore are shown in [11]. Meteorological data are used to reconstruct the refractivity profiles from ECMWF, operational model TL799L91, conducted in a two-year period 2007 and 2008. Statistics of essential duct parameters for surface, surface-based and elevated ducts are reported for the summer months and overall duct statistics are given for the other seasons. The radio refractivity N is computed from the meteorological parameters in ITU-R, Rec.P.453-9, 2003. The report showed no principal difference in the duct formation, occurrence and parameters for the two bays: the Varna bay indicated higher duct occurrence in all seasons, especially for the areas closest to the coast. The mean and median M-deficits of layers forming all types of ducts are commonly higher for Varna bay than for Burgas bay. The statistics for the studied period showed also, that spring season may be rich in trapping layers; this mostly refers to the bay of Varna. The changeability of the essential duct parameters in the 50 km littoral zone points showed the necessity of using range-dependent refractivity profiles, when predicting propagation conditions even for not-so-big distances.

4. Conclusion

A brief review of literature is proposed which aims is to summarize the existing models for radio wave propagation over sea environment, the statistics of duct occurrences and real performed experimental measurements. A brief overview of the basic channel model methods (Parabolic Equation Method and its three popular numerical methods for solving - split step Fourier (SSF), finite difference (FD), and finite element (FE) is observed. Using Two and Tree ray pathloss models we can identify the trajectories of the most dominant rays arrived at the receiver, thus the phase shift of each ray is characterized and considered in the path loss calculation. To describe the propagation characteristics in the spatial or delay domain, i.e., the AoA/AoD and the delay spread the ray optics (RO) method is introduced to find the trajectory of each ray by solving well known Snell's equation. Advanced Propagation Model APM is a hybrid model, consisting of four basic submodels which are flat earth (FE), ray optics (RO), extended optics (XO), and the split-step Fourier PE algorithm. After comparing the examined models it can be conlude that Parabolic equation model and his numerical scheme Split-step Fourier Transform is most universal and used method, computionaly more efective and has larger step size. Occurrence of elevated ducts in lower atmosphere near Arabian Sea, Surface Duct Conditions over Istanbul and over Bulgarian Black Sea cost have been also examined. The authors in most articles have proposed a different and/or new model for solving the problem of the abnormal propagation of radio waves in the troposphere above the water surface in order to accurately predict pathloss. But the examples and considerations reported in the most articles demonstrated the need of additional investigations, both theoretical and experimental, in order to improve the propagation models, which could improve the maritime (naval and civil) communications.

References

[1]. Y. H. Lee, F. Dong and Y. S. Meng, "Near Sea-Surface Mobile Radiowave Propagation at 5 GHz: Measurements and Modeling," Radioengineering, vol. 23, 2014.

[2]. O. B. Akan and E. Dinc, "Beyond-line-of-sight communications with ducting layer," IEEE Communications Magazine, vol. 52, no. 10, pp. 37 - 43, 2014.
[3]. M. Levy, “Parabolic Equation Methods for Electromagnetic Wave Propagation”, The Institution of Electrical Engi-neers, London 2000
[4]. O. Ozgun, G. Apaydin, M. Kuzuoglu, L. Sevgi “PETOOL: MATLAB-based one-way and two-way split-step parabolic equation tool for radiowave propagation over variable terrain”, Computer Physics Communications, December 2011.
[5]. P. Valtr, P. Pechač, “Tropospheric Refraction Modeling Using Ray-Tracing and Parabolic Equation”, Radioengineering 14, December 2005.
[6]. G. Apaydin and L. Sevgi “The Split-Step-Fourier and Finite-Element-Based Parabolic-Equation Propagation Prediction Tools: Canonical Tests, Systematic Comparisons, and Calibration”, IEEE Antennas and Propagation Magazine, Vol. 52, No.3, June 2010
[7]. J. Wang, T. Q. Quek, H. Zhou and Y. Li, "Wireless Channel Models for Maritime Communications," IEEE Access, p. 1, 2018.
[8]. Barrios, "Considerations in the development of the advanced propagation model (APM) for U.S. Navy applications," in Proceedings of the International Conference on Radar (IEEE Cat. No.03EX695), 2003.
[9]. Ullah, S. U. Rehman and N. Mufti, "Investigations into the occurrence of elevated ducts in lower atmosphere near Arabian Sea," in International Conference on Space Science and Communication (IconSpace), 2015.
[10]. S. Mentes and Z. Kaymaz, "Investigation of Surface Duct Conditions over Istanbul, Turkey," JOURNAL OF APPLIED Meteorology and Climatology, vol. 46, no. 3, p. 318, 2007.
[11]. Sirkova, "Duct occurrence and characteristics for Bulgarian Black sea shore," Journal of Atmospheric and Solar - Terrestrial Physics , vol. 135, pp. 107-117, 2015.

For contacts:
Msc, Nikolay Grozev
Department of Communication Engineering and Technologies
Technical University of Varna
E-mail: n.grozev@tu-varna.bg

ОБОБЩЕНА КЛАСИФИКАЦИЯ НА БЛОКЧЕЙН ТЕХНОЛОГИИТЕ

Антон П. Хулиян

Резюме: Интересът към използване на блокчейн технологиите е по-голям от всякога и за първи път големи корпорации позволяват на своите потребители разплащане с криптовалути, което е ключов момент към масово използване. Друга голяма стъпка е приложението му за оптимизиране и ускоряване на бизнес процеси. Тази статия разглежда съществуващите блокчейн технологии, тяхното приложение и предлага обобщена класификация.
Ключови думи: Блокчейн, Крипто валута, Разпределени бази

General Taxonomy of Blockchain Technologies

Anton P. Huliyan

Abstract: Interest in the use of blockchain technology is greater than ever and for the first time large corporations allow their customers to pay with cryptocurrencies, which is a key moment for mass adoption. Another big step is its application to optimize and accelerate business processes. This article discusses existing blockchain technologies, their application and offers a general taxonomy.
Keywords: Blockchain, Cryptocurrency, Distributed ledgers

1. Въведение

След масовия интерес към криптовалутите и блокчейн технологиите през 2017та година и последвалия спад в началото на 2018та година, през 2020та година отново се връща интересът към тях. Технологиите са изминали дълъг път, все повече компании започват да ги прилагат в реални проекти и да се вижда ползата от тях. Броят на транзакциите отново е на нивата от 2017та година и дори започва да го подминава (фигура 1).

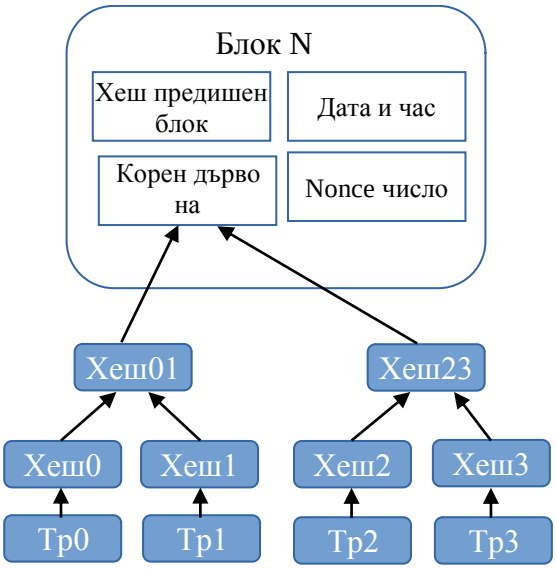
Фиг. 1. Среден брой транзакции за 24часа [4]

Една от най-големите стъпки към масово използване на блокчейн технологии за разплащане е решението на Paypal [1] да позволи тяхната покупка, съхранение и разплащане в своята мрежа, която разполага с 26 милиона търговци по целия свят, включително и една от най-големите платформи за търговия в Интернет eBay. Една от най-големите застрахователни компании AXA [2] стартира пробен проект за застраховка на закъснели полети посредством блокчейн. Световното положение на глобална епидемия поставя производството и търговията под много голямо натоварване. Блокчейн технологиите се оказват много добро средство за проследяване на производството, контрол на качеството и насочване към най-натоварените болници [3].

В тази статия се разглежда концепцията на блокчейн технологията, различни видове блокчейн мрежи и тяхното приложение. Предлага се обобщена класификация на блокчейн технологиите.

2. Същност на блокчейн технологията

За първи път блокчейн се появява в публикацията на Сатоши Накамото [5], като средство за реализация на виртуална валута. Тя се базира на вече познати технологии (разпределени бази данни, публичен-частен ключ, хеширане чрез дърво на Меркел, консенсусен механизъм, криптиране), чиято комбинация служи за разрешаване на проблема с двойния харч, който е съществувал в предишните предложения за виртуална валута [6]. Дървото на Меркел е патентовано от Ралф Меркел през 1979 като средство за комуникация в несигурни канали. Информацията в него се съхранява в хеширан формат, като хешироването е познато от 1950 година. Първите предложения, описващи криптовалути, се появяват през 90-те години. Блокчейн технологията, описана от Накамото, съхранява информацията в блокове, като всеки има максимален размер от 1MB. Когато бъде излъчена транзакция в мрежата, за да бъде валидна, тя трябва да влезе в блок. Всички транзакции в блока се оформят в дърво на Меркел. Към него се добавя часът и датата на създаване на блока, хеш кодът на предишния блок и nonce число. Структурата на блок от веригата на Bitcoin е показана на фигура 2.



Фиг. 2. Структура на блок при Bitcoin

Съдържанието на хеш кода на предишния блок гарантира, че вече влезли във веригата блокове няма да бъдат манипулирани. Ако даден блок бъде манипулиран, неговият хеш код ще бъде променен, тази промяна ще трябва да се отрази в следващия блок, което ще доведе и до промяна в неговия хеш на следващия блок и така до последния блок. Но дори и това да бъде извършено, всеки участник в мрежата има собствено копие на блок веригата и при

излъчване на невярната информация, няма да се получи консенсус от останалите участници и тя ще бъде отхвърлена. Nonce числото служи като доказателство за извършена работа и като механизъм за регулиране на времето за създаване на нов блок. В зависимост от броя на участниците в мрежата, които валидират транзакции и тяхната изчислителна мощ, дължината на nonce числото се променя, за да може винаги да има равни интервали между блоковете. Поради извършваната работа за намиране на nonce числото алгоритъмът за постигане на консенсус в Bitcoin мрежата се нарича Proof-of-Work(PoW). Освен PoW се появяват и други видове консенсусни механизми - Proof-of-Stake, Proof-of-Authority, Proof-of-Storage, Hybrid, Byzantine Fault Tolerance.

Благодарение на невъзможността за променяне на вече излъчени в мрежата данни, блокчейн технологията се смята за относително сигурна. Колкото повече време минава от записването на данните в мрежата, толкова по-сигурни са те. С всеки новодобавен блок работата, необходима за тяхната манипулация, се увеличава и усложнява. За успешна хакерска атака е необходимо влияние върху повече от 51% от мрежата. Но дори и при това условие би се получило разделяне на мрежата на паралелни вериги, като и двете биха продължили да съществуват, докато има участници в тях и биха били толкова достоверни, колкото участниците в тях имат доверие във веригата. Въпреки невъзможността за промяна на вече записани данни, в историята са познати няколко случая на заличаване на вече записвани данни, като мрежата се връща до състояние преди тяхното добавяне. Тези „изключения“ са били извършени след публично допитване до участниците в мрежата. В резултат на такива събития мрежата на Ethereum се разделя на паралелните вериги – Ethereum и Ethereum Classic. Ethereum Classic запазва оригиналната блок верига на Ethereum, без изтриване на данните, поради причината, че част от общността не подкрепя тази мярка и иска да запази оригиналната блок верига. Друг такъв случай е появата на Bitcoin Cash, като паралелна верига на Bitcoin.

Друга блокчейн мрежа е Ethereum [7], която позволява автоматичното изпълнението на умни договори (smart contracts). Чрез тях може да бъде описана бизнес логика, която да бъде автоматично изпълнявана в блокчейн мрежата при достигане на определени условия. Умните договори намират приложение в създаването на токени - единици виртуална валута, които се създават и използват със специфична цел. Основната криптовалута в Ethereum мрежата е Етер, съществува и спомагателна, която се използва за такси при излъчване на транзакции – GAS. Умните договори създават нов слой в блокчейн мрежата, различен от този за транзакции, като използват мета данните във всеки блок за съхраняване на информация.

3. Класификация на блокчейн технологиите

За реализация на блокчейн могат да бъдат използвани много подходи и набор от различни технологии за постигане на желания резултат. На база съществуващите технологии за изграждане на блокчейн трябва да се състави класификация, която да оценява техните предимства и недостатъци, и да служи като насока за избора им при реализация на проект или тяхното задълбочено изследване. Като сравнително скоро появила се технология, няма много разработки в областта на създаване на класификация на блокчейн технологиите.

В [10] са разгледани различните компоненти, които съставят един блокчейн и са разбити на подкомпоненти. Всеки компонент е представен подробно и са разгледани различните варианти за реализация, които са сравнени помежду си. В публикацията са представени следните главни компоненти – “Консенсус”, “Транзакции”, “Основна валута и токени”, “Разширяемост”, “Сигурност и Поверителност”, “Код за програмиране”, “Достъп до мрежата”, “Такси и Система за награди”.

Друга публикация на Лабазова и др. [11] представя класификация на приложението на блокчейн технологиите. Разгледани са шест области на приложение: “Финансови

транзакции”, “Умни договори”, “Мениджмънт на данни”, “Съхранение”, “Комуникация”, “Категоризиране”. За да бъдат оценени блокчейн технологиите във всяка от областите, се разглеждат 8 техни технически характеристики – “Четене на данни”, “Запис на данни”, “Механизъм за консенсус”, “Анонимност”, “Управление на събития”, “Обмяна на информация”, “Криптиране”, “Историческо менажиране”.

На база разгледаните публикации и други критерии, разгледани в тях, се предлага обобщена класификация на блокчейн технологиите, показана на фигура 3. Най-важните характеристики на една блокчейна мрежа са :

- Достъп на участници в мрежата – тази характеристика разделя блокчейн мрежите основно на два вида – частни и публични. Частните мрежи са ограничени за достъп, наричани още блокчейн с позволение (Permissioned), само от участници, които имат нужните права. Правата се определят и раздават от организацията, която има контрола над мрежата Обикновено при този тип мрежи участниците в мрежата не са анонимни. Те могат да имат различни права на достъп и извършване на операции, например само определена група имат право да валидират транзакции, а друга да излъчва транзакции в мрежата. Публичните блокчейни могат да бъдат с и без позволение (Permissioned, Permissionless). При публичните блокчейн мрежи без позволение няма ограничения на достъпа за участници в тях. Всеки с интернет достъп може да излъчва данни в мрежата, които в последствие да бъдат валидирани. Валидирането на транзакции е достъпно за всеки изпълнил условията за това, като те зависят от консенсусния механизъм, например да извършва изчисления със своя хардуер (PoW) [8] или да заключи определен брой единици валута (PoS) [9]. Участниците могат да бъдат анонимни, като е видим само техният публичен адрес. Всички транзакции в мрежата са видими от всички участници и те притежават копие на цялата блок верига. При публичните блокчейни с позволение отново всички участници могат да четат транзакциите в мрежата, но само на тези, на които са дадени права, имат право да излъчват транзакции и да участват в процеса на валидация. Докато при тези без позволение участниците в мрежата участват активно в развитието на мрежата, като имат право да гласуват за промените, при тези с позволение организацията зад системата решава това.
- Консенсусен механизъм – това е една от най-важните характеристики на една блокчейн мрежа, която до голяма степен определя нейната сигурност, бързодействие, разход на енергия, възможност за разрастване. Първият механизъм за консенсус, използван в блокчейн технология, е Proof-of-Work, който използва изчислителната мощ на хардуера (процесор, видео карта, ASIC) за извършване на математически изчисления, изчисляване на nonce число като доказателство за извършена работа. Това число се добавя към хеш кода на блока и сумата трябва да започва с точно определен брой нули. Друг механизъм за постигане на консенсус е Proof-of-Stake - при него участник, който иска да извършва валидиране на транзакции, заключва определена сума от валутата в мрежата. Ако той излъчва неверни данни в мрежата, губи заключената сума. Обикновено сумата, която трябва да се заключи, е значителна и по този начин се стимулира излъчването само на верни данни. Освен загубата на средства, участниците са стимулирани от това, че ако мрежата е смятана за несигурна, никой няма да иска да извършва транзакции в нея и наличностите на участниците в нея не биха имали никаква стойност. Както при PoW, така и при PoS валидаторите на транзакции получават като награда таксите, платени за излъчване на транзакция в мрежата.
- Основна обменна единица – основна характеристика на валутата в една блокчейн система е нейната крайност, дали има ограничен брой единици, които могат да съществуват или той е неограничен. При Bitcoin има ограничение от 21 милиона единици и след като всички влязат в обръщение, няма механизъм, по който да се имитират още. При Ethereum, няма краен брой етери, които могат да бъдат имитирани. Друга важна характеристика е първоначалният начин на разпределение на единиците. В Bitcoin мрежата нови единици се отключват от протокола при всеки нов блок, до достигане на 21 милиона. При други мрежи

това се случва чрез ICO, initial coin offering, в случая на Ethereum, а при други всичко зависи от хората, управляващи мрежата, например Neo [13] и Ripple [16]. При последните изцяло хората, притежаващи права над мрежата, създават и пускат в обръщение единици.

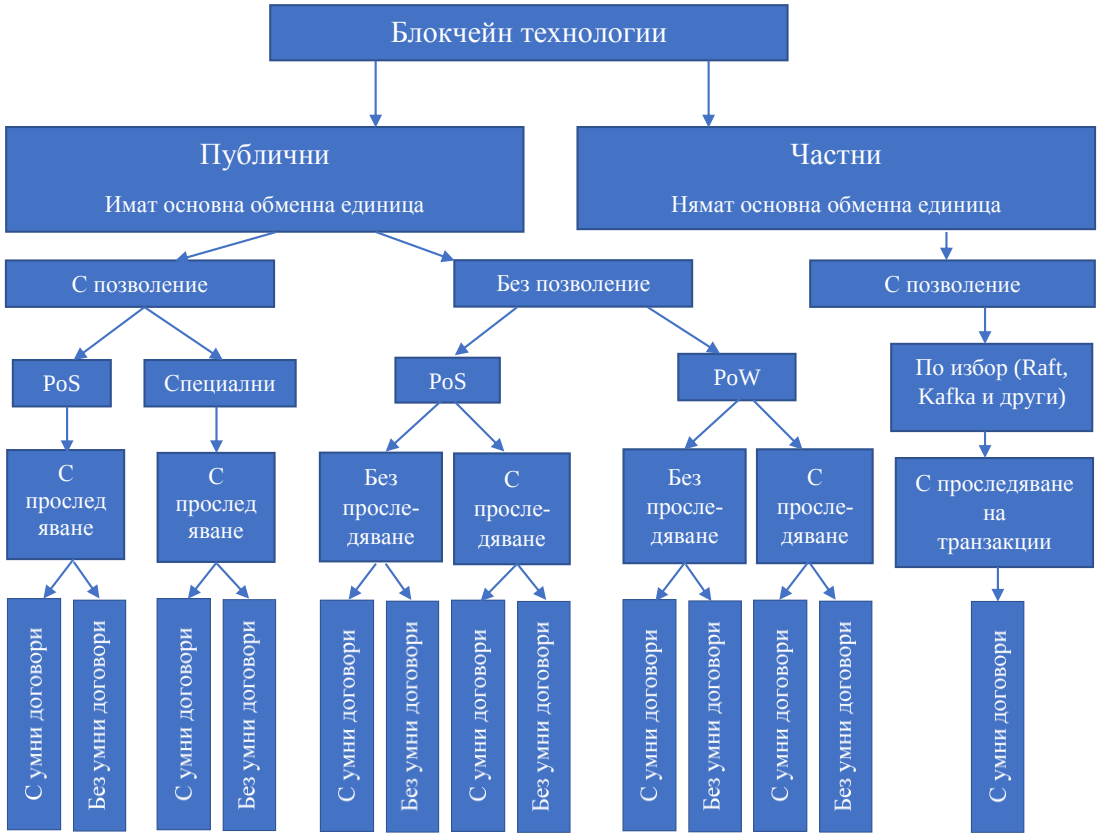
- Проследяване на транзакциите – в блокчейна на Bitcoin всяка една единица валута може да бъде проследена до момента на нейното откриване. Въпреки че може да се проследи, не може да се знае кое точно лице я притежава, освен ако то не пожелае, защото е видим само публичният ключ на притежателя и ако той не обяви публично това, адресите остават анонимни. Това дава възможност за защита от злоупотреби, като адреси, за които се знае, че участват в престъпна дейност, могат да бъдат маркирани и да бъдат отказвани операции с тях. От друга страна, ако бъде разкрита самоличността на притежател на портфейл, всеки може да види неговия баланс и той да стане цел на измами или друг вид последствия, които той не желае. Дори той да премести средствата в друг портфейл, с нов публичен ключ, трансферът ще бъде публично видим от кой към кой адрес е бил насочен. Съществуват блокчейн системи, при които не може да се проследи пътят, през който е минала дадена единица по време на транзакция. Най-популярните такива системи са Monero [14] и Zcash [15].
- Умни договори – това концептуално представлява отделен слой в блокчейн мрежата, отделен от този за транзакции, в който може да се опише бизнес логика. В нея могат да се описват условия, при достигането на които определени събития да се изпълняват. Най-честата им употреба е създаване на токени, те представляват единици валута, създадена с определена цел, съществуваща паралелно с основната. Умните договори набират голяма популярност за осъществяване на децентрализирани борси за крипто токени. Първият блокчейн, който развива тази концепция, е Ethereum. Когато умният договор се достъпва за четене, не се заплаща никаква такса, но при запис на данни се заплаща такса, размерът на която зависи от това колко бързо се изисква тази информация да влезе в блок. Един път качен в мрежата, договорът не може да бъде променян, единствено могат да се качват нови негови версии, като старите продължават да съществуват, и за да се гарантира коректна работа, винаги трябва да се работи с най-новата налична версия. При Ethereum се използва по-различна версия на дървото на Меркел, понеже умният договор се записва в определен блок, на по-късен етап, когато той трябва да бъде достъпен, трябва да се знае къде се намира той, затова се използва дърво на Меркел-Патриша (Merkel-Patrica Tree)[12]. Това е комбинация от дърво на Меркел и дърво на Патрища, наричано още дърво с префикс. При него се съхранява чрез префикс пътят до даден нод от дървото. По този начин се проследява пътят до умния договор, който трябва да бъде изпълнен. При Ethereum е разработен специален език за програмиране, Solidity, за имплементация на умни договори. Друг блокчейн, който ги поддържа, е Neo. При него се използват конвенционални езици за програмиране C++ и Java. Блокчейн, на който основната му цел е изпълнение на бизнес логика, е HyperLedger Fabric, който позволява изграждането и интегрирането на частни блокчейн решения. В Bitcoin протокола не е предвидено изпълнението на умни договори.

В Таблица 1 са представени едни от най-популярните блокчейни криптовалути и системи. За всяка от тях са представени данни според характеристиките описани по-горе.

Таблица 1. Сравнение на едни от най-популярните блокчейн технологии

Блокчейн	Достъп на участниците до мрежата	Консенсусен механизъм	Основна обменна единица	Проследяване на транзакциите	Умни договори
Bitcoin (BTC)	Публичен, без позволение	PoW	Да Максимум: 21млн	Да	Не
Ripple	Публичен, с позволение	RPCA	Да ∞	Да	Не
Ethereum(ETH)	Публичен, без позволение	PoW	Да ∞	Да	Да

Litecoin(LTC)	Публичен, без позволение	PoW	Да Максимум: 84млн	Да	Не
Cardano(ADA)	Публичен, с позволение	PoS	Да Максимум: 45млрд	Да	Да
EOS (EOS)	Публичен, с позволение	DPoS	Да ∞	Да	Да
Monero(XMR)	Публичен, без позволение	PoW	Да ∞	Не	Не
TRON (TRX)	Публичен, с позволение	PoS	Да ∞	Да	Да
Stellar(XLM)	Публичен, с позволение	The Stellar Consensus Protocol (SCP)	Да ∞	Да	Да
Neo(NEO)	Публичен, с позволение	dBFT	Да Максимум: 100млн	Да	Да
Zcash(ZEC)	Публичен, без позволение	PoW	Да Максимум: 21млн	Не	Не
Hyperledger Fabric	Частен	Raft, Kafka	Няма	Да	Да
Corda	Частен	По избор	Няма	Да	Да



Фиг. 3. Обобщена класификация на блокчейн технологиите

4. Предимства и недостатъци на частните и публичните блокчейн технологии

4.1 Публични блокчейн мрежи

Първата блокчейн технология, която се е появила – Bitcoin, е публична блокчейн мрежа, в която всеки има право да участва, като излъчва транзакции в мрежата, валидира транзакции, прави справки за отминалите транзакции в мрежата, дори и да има пълно копие на всички транзакции в мрежата от началото на нейното създаване. Всички участници в мрежата имат равни права, по този начин няма нужда от доверия към дадени участници, които да извършват определени действия. Получава се среда от участници, които помежду си нямат доверие, но имат доверие в мрежата, в която участват. Колкото повече участници има, толкова по-сигурна е тя. Децентрализацията предотвратява атаките за контрол върху мрежата, защото колкото повече участници има, над толкова по-голям брой трябва да се придобие контрол за нарушаване на консенсуса в мрежата.

Една от първите блокчейн технологии, насочени към бизнеса, по-точно между банковите трансфери и плащания от тип кредитна карта, е Ripple. Ripple е компания, която предоставя блокчейн решение с отворен код, който е базиран на кредитната система. Блокчейн системата, която функционира в момента, се нарича RippleNet и е от тип публична, с позволение, обменната единица се нарича XRP. Създаването на нови инстанции в мрежата е само с позволение, а не както при Bitcoin, всеки да има свободен достъп. Целевата група от потребители също е различна, докато Bitcoin е насочена към крайния потребител, Ripple цели B2B (Business-to-Business) [17] модел на работа. Консенсусният механизъм се нарича RPCA (Ripple Protocol consensus algorithm) [18] и е разработен от компанията. Той предоставя много по-малки времена за валидация на транзакция, както и по-голям брой отколкото при Bitcoin.

Основен недостатък на публичните блокчейн мрежи е тяхното бързодействие, поради факта че трябва да се получи консенсус между много несигурни участници, както и че трябва да има ясни правила на колко време и колко голям блок трябва да се създаде. Фиксираните времена на блок и неговия размер не позволят увеличаване на пропускателната способност на мрежата при увеличаване на участниците в нея или в моментите на по-голяма активност.

Анонимността на участниците в нея е предимство и недостатък. Участници в нея могат да я използват за криминална дейност и служителите на закона нямат достъп до лицето, което се крие зад даден крипто портфейл.

Невъзможността за променяне на вече записани данни блок е основно правило на блокчейн системите, но това води и до някои негативи. Ако бъде направена транзакция с грешен получател или ако някой придобие частния ключ и изпрати всички средства до своя адрес, това няма как да бъде променено. Докато при традиционната банкова система могат да се правят корекции по трансфери, както и злонамерени транзакции да бъдат отменени и клиента да получи обратно своите пари. Грешки в написването на умен договор, в който се съхраняват много големи суми, могат да бъдат използвани от хакери, за да ги придобият. Единственият вариант за връщането им е връщане на целият блокчейн до състояние преди кражбата и коригиране на несигурността в умния договор, но този вариант е много сложен и би довел до големи проблеми в мрежата и трябва да се получи съгласие от участниците в мрежата.

4.2 Частни блокчейн мрежи

Компаниите в почти всички сфери на бизнеса подлежат на много регулации, които целят да намалят злоупотребите, прането на пари, укриване на данъци и всякакви практики, които биха били в ущърб на клиентите им. Поради тази причина употребата на публични блокчейн мрежи е неприложимо, заради достъпа на всеки до тях, а в почти всички случаи и

анонимността. Нуждите на бизнеса включват регистриран достъп на всеки потребител, строго определени права за извършване на процеси от потребителите, сигурна комуникация между много отделни организации.

HyperLedger [19] е блокчейн решение под лицензa на Linux Foundation. За разлика от Ripple, които предлагат мрежа, към която бизнесът може да стане част, HyperLedger предоставя възможността бизнесът да изгради блокчейн мрежа, която да отговаря точно на неговите нужди. Предоставят се опции за избор на топология на мрежата, на консенсусен механизъм и различни начини и езици за програмиране на умни договори. Недостатък спрямо Ripple е нуждата от квалифициран персонал, който да проектира и изгради блокчейн системата посредством инструментите, предоставени от HyperLedger.

Основен недостатък на частните блокчейн мрежи е по-малкият брой участници и нуждата от доверие между тях. За делегирането на права и задължения на точно определени участници в мрежата за извършаването на определени операции, трябва всички участници в мрежата да им имат доверия. Намаляването на децентрализацията също води до намаляване на сигурността в мрежата - малко на брой участници в една бизнес мрежа биха валидирали транзакции от гледна точка на сигурността, но това всъщност увеличава риска, поради по-малкото участници, над които трябва да се придобие контрол за злонамерено повлияване на мрежата.

В публикацията на Katia Sayegh [20] е разгледано приложението на блокчейн системите в различни отрасли на бизнеса и по-специално в застрахователния бизнес, който е характерен с много дълги и бавни процеси, включващи много документи между различни организации. Изследвани са различните процеси, които се извършват и се разглежда потенциални начини за оптимизация чрез блокчейн технологии.

5. Заключение

След като началният ентусиазъм към блокчейн технологиите е преминал, сега се намира във фаза на изследване на реалната им употреба и изграждане на решения за оптимизиране на бизнес процеси. В тази статия са разгледани съществуващите блокчейн технологии и се предлага обобщена класификация. На база разгледаните предимства и недостатъци публичните блокчейн мрежи намират приложение като системи за разплащане и системи, при които е важна анонимността на участниците. Частните блокчейн технологии са насочени към бизнес решения, в които се изисква бързодействие, реализация на сложна бизнес логика и механизъм за контрол на правата на потребителите.

Разгледаните съществуващите модели за класификация за блокчейн технологиите изследват характеристиките им, но не предлагат обобщена класификация. Необходимо е по-детайлно изследване за класифициране на консенсусните механизми и отличаващите се характеристики на частните и публичните блокчейн мрежи

Литература

[1]. PayPal-Launches-New-Service-Enabling-Users-to-Buy-Hold-and-Sell-Cryptocurrency, Oct. 21, 2020, https://newsroom.paypal-corp.com/2020-10-21-PayPal-Launches-New-Service-Enabling-Users-to-Buy-Hold-and-Sell-Cryptocurrency, last visit on 30.10.2020

[2]. AXA goes blockchain with fizzy, Sep 13, 2017, https://www.axa.com/en/magazine/axa-goes-blockchain-with-fizzy, last visit on 30.10.2020

[3]. IBM using blockchain to connect pop up medical mask and equipment makers with hospitals, April 28, 2020, https://www.techrepublic.com/article/covid-19-ibm-using-blockchain-to-connect-pop-up-medical-mask-and-equipment-makers-with-hospitals/, last visit on 30.10.2020

[4]. Average Transactions Per Block, https://www.blockchain.com/charts/n-transactions-per-block, last visit on 30.10.2020
[5]. Nakamoto, S. “Bitcoin: A Peer-to-Peer Electronic Cash System.” (2008), https://bitcoin.org/bitcoin.pdf, last visit on 30.10.2020
[6]. Back A. Hashcash - Amortizable Publicly AuditableCost-Functions, 2002, http://www.hashcash.org/papers/amortizable.pdf, last visit on 30.10.2020
[7]. Buterin V., Ethereum: A Next-Generation SmartContract and Decentralized Application Platform, 2013, https://ethereum.org/en/whitepaper/, last visit on 30.10.2020
[8]. Gervais A., Ghassan K., Karl W. and others, On the Security and Performance of Proof of Work Blockchains, 2016, DOI: 10.1145/2976749.2978341
[9]. W. Y. Maung Maung Thin, N. Dong, G. Bai and J. S. Dong, "Formal Analysis of a Proof-of-Stake Blockchain," *2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*, Melbourne, VIC, 2018, pp. 197-200, doi: 10.1109/ICECCS2018.2018.00031.
[10]. Tasca P., Claudio T., A Taxonomy of Blockchain Technologies: Principles of Identification and Classification, 2019, DOI: 10.5195/ledger.2019.140
[11]. Labazova O, Dehling T., Sunyaev A., From Hype to Reality: A Taxonomy of Blockchain Applications, Scholarspace, 2019, ISBN: 978-0-9981331-2-6, DOI: 10.24251/HICSS.2019.552
[12]. V. Leis, A. Kemper and T. Neumann, "The adaptive radix tree: ARTful indexing for main-memory databases," *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, Brisbane, QLD, 2013, pp. 38-49, doi: 10.1109/ICDE.2013.6544812.
[13]. Neo Whitepaper, https://docs.neo.org/docs/en-us/basic/whitepaper.html, last visit on 05.11.2020
[14]. Monero Whitepaper, https://decred.org/research/saberhagen2013.pdf, last visit on 05.11.2020
[15]. Ben-Sasson E., Alessandro C., Christina G. and others, Zerocash: Decentralized Anonymous Payments from Bitcoin, *2014 IEEE Symposium on Security and Privacy*, San Jose, CA, 2014, pp. 459-474, doi: 10.1109/SP.2014.36.
[16]. Armknecht F., G. Karame, A. Mandal, E. Zenner and others, Ripple_Overview_and_Outlook, Trust and Trustworthy Computing, 2015, DOI:10.1007/978-3-319-22846-4_10
[17]. Ripple cost mode, https://ripple.com/files/xrp_cost_model_paper.pdf, last visit on 05.11.2020
[18]. Ripple consensus whitepaper, https://ripple.com/files/ripple_consensus_whitepaper.pdf, last visit on 05.11.2020
[19]. HyperLedger Whitepaper, https://www.hyperledger.org/learn/white-papers, last visit on 05.11.2020
[20]. Katia Sayegh “Blockchain Application in Insurance and Reinsurance“

За контакти:
Антон П. Хулиян
E-mail: anton.huliyan@gmail.com

ПРОГРАМНА СИСТЕМА ЗА ОЦЕНКА НА РИСКА ЗА ИНФОРМАЦИОННАТА СИГУРНОСТ

Петко Г. Генчев, Недялко Н. Николов

Резюме: Тази статия разглежда възможностите, на реализиран програмен продукт, който подпомага процеса на оценка на риска за информационната сигурност в една организация. При изграждането на продукта са използвани предварително направен анализ на проблемите при оценката на риска [1] и разработен подход за динамично следене, структуриране и обработване на рисковите фактори при оценка на риска [2]. В продукта са реализирани различни възможности за облекчаване на оценката на риска на СИ и за преодоляване на основни затруднения, свързани с определяне и остойностяване на рисковите фактори, облекчаване на избора на методи за контрол, събиране на информация за влияния между рисковите фактори и отношение на персонала към процеса на оценка на риска за СИ.
Ключови думи: Управление на риска за сигурността на информацията, оценка на риска за сигурността на информацията, информационни технологии и системи, методи за събиране на данни.

Information Support System to Tnformation Security Risk Assessment.

P. Genchev, N. Nikolov

Abstract: This article discusses the possibilities of a implemented software product that supports the process of risk assessment for information security in an organization. During the construction of the product, a previously made analysis of the problems in the risk assessment and an already proposed methodology for dynamic monitoring, structuring and processing of the risk factors in the risk assessment were used. The product has various opportunities to facilitate the assessment of the risk of Information Security (IS) and to overcome the main difficulties associated with identifying and evaluating risk factors, facilitating the choice of control methods, collecting information on influences between risk factors and staff attitudes to the IS risk assessment process.
Keywords: Dynamic systems and control, Information Technology and Systems, Methods of data collection, Risk management, Tracking.

1. Увод

В съвременния свят все повече организации разчитат на информационни технологии, за да им помогнат да постигнат своите бизнес цели. Ето защо сигурността на информацията е от първостепенно значение за организациите. Необходим е систематичен подход за управление на риска по отношение на сигурността на информацията, който да помогне да се определят изискванията за сигурност на информацията и да се създаде ефективна система за управление. Група стандарти на ISO/IEC 27xxx. определят изисквания за създаване, сертифициране и използване на Система за Управление на Сигурността на Информацията (СУСИ) и методите за сигурност при Управление на Риска за Сигурността на Информацията (УРСИ) (ISRM). На базата на анализа на проблемите [1] са изведени основни препоръки за подпомагане на процеса на оценка на риска [2]. Направените препоръки са оформени като задание за изработване на програмен продукт, чрез който да се решат основни проблеми при УРСИ.

Целта на тази статия е да представи възможностите на вече изграден програмен инструмент, който е реализиран на базата на формулиран подход [2] за преодоляване на някои проблеми, свързани със събиране и обработка на информация, необходима за оценка на риска при системи за УРСИ. Същността на използвания подход дава възможност за

динамично следене, структуриране и обработване на рисковите фактори при оценка на риска.

Статията е структурирана в 5 раздела, като във втори раздел, на базата на описания подход, се формулират основните цели, които се поставят за решаване от програмната система. В трети раздел са описани основните акценти от използвания в програмата подход за динамично следене, структуриране и обработване на рисковите фактори при оценка на риска. В четвърти раздел са описани някои от реализираните функционалности на продукта в светлината на решаваните проблеми. Петият раздел представлява заключение, в което се обобщават възможностите на изградения програмен продукт и се правят препоръки за допълване на неговите възможности.

2. Цели за изграждане

Анализът на констатираните проблеми води до извода, че основната част от тях са свързани с подценяване, неизпълнение или заобикаляне на препоръките и изискванията, които дават стандартите за изграждане на СУСИ и УРСИ. Различни причини влияят на неизпълнението на организационните и техническите препоръки и изисквания на стандартите. Една част са свързани с финансови причини, а други с некомпетентност и субективни причини. Налага се изводът, че за повишаване на ефективността от въвеждане на система за УРСИ е необходимо да се въведат средства, които да въведат по-строги отговорности на всички нива на управление [1]. Освен това може да се направи заключението, че повечето проблеми са свързани с нуждата от въвеждане на единен инструмент за събиране и обработване на информация за изграждането, обработването и адаптацията на системата към бързо променящата се среда. При по-внимателен анализ се вижда, че голяма група проблеми са свързани с отчитането и съобразяването със сложни взаимовръзки от информационен, организационен, технически и субективен характер.

В изградения подход за динамично следене, структуриране и обработване на рисковите фактори при оценка на риска се предлагат начини за преодоляване на някои проблеми, които възникват в процеса на експлоатация на УРСИ. Основното в подхода е заложеният механизъм за следене и оценка на активността на собствениците на активи и предоставяне на по-добри възможности за структуриране и използване на натрупваната информация за рисковите фактори за сигурността на информацията. Във връзка с препоръките от подхода могат да се изведат основните цели за изграждане на програмен продукт за подпомагане на оценката на риска:

- Изграждане на инструментални средства за структурирано въвеждане, редактиране и използване на възможно по-пълна информация за рисковите фактори;
- Подходящо структуриране на информацията с цел отчитане на взаимовлиянията между рисковите фактори;
- Осигуряване на даннови структури и инструментални средства за проследимост на въвеждане и проверка на рисковите фактори;
- Осигуряване на йерархична структура за достъп, въвеждане и контрол;
- Осигуряване на справочна информация за ръководството на организацията, отговорниците по сигурността и експерти по Сигурност на Информацията (СИ);
- Изграждане на средства за филтриране на чувствителната информация и даване на възможност за натрупване на информация за рисковите фактори от УРСИ системи на различни организации;

3. Акценти от използвания подход

3.1 Алгоритъм за следене и оценка на активността на собствениците на активи

Предлага се изграждането на динамична система за формиране на добавъчно ниво на риска, което е свързано с активността на проверките на рисковите фактори за даден актив, които се извършват от собственика на актива. Тази добавка може да участва в крайното формиране на нивото на риска на даден актив. По този начин при неактивен собственик на актив, с времето крайното ниво на риска на актива ще нараства. При старателен собственик на актив, който стриктно спазва дефинирани срокове за проверка, нивото на риска за актива се запазва или след определено време спада до начално изчисленото ниво на риска за актива. Такава система за динамично актуализиране на нивото на риска за даден актив трябва да работи с два параметъра – стойност на рисковата добавка и зададен максимален период за проверка и актуализиране на рисковите фактори на актива. При просрочване на периода за проверка, към общото ниво на риска се добавя рисковата добавка. Обратно - при дълготрайна тенденция на спазване на зададения период за проверка да се изважда рисковата добавка, докато се достигне първоначално изчисленото ниво на риска за актива (без отчитане на активността на собственика) [2].

3.2 Организиране на йерархична структура на въвеждане и контрол на рискивите фактори

Като основание за йерархичната структура се използват понятията актив и собственик на актива, както и обстоятелството, че актив може да бъде и човек. При правилно разпределение на активите между собствениците ще се получи йерархия на проследяване на риска, която е повторение на административната йерархия в организацията. Така на най-ниското ниво ще стоят собственици, които имат отношение към материални и информационни активи, но не и към служители, които имат отговорности за СИ. В средата на структурата ще се намират собственици и на активи, които представляват служители, имащи отношение към СИ, т.е. това ще бъдат ръководни кадри от административната структура, които имат подчинени служители. На върха на информационната структура ще стоят информационни активи, които нямат собственик и представляват административни ръководители от най-високо ниво, които не са подчинени пряко на друг ръководител. Такава нивова информационна структура ще позволява контрол на събирането, обработването и цялостно отношение към информационните активи, което съответства на административния контрол за осъществяване на целите на организацията.

3.3 Отчитане на влиянието на рисковете между активите

Това отчитане трябва да е съпроводено с допълнителна информация, която гарантира проследимост. Необходима е методика за отчитане на влиянието. Отчитането на всички рискови фактори и третирането им трябва да доведе до лесна проследимост на динамиката на изменението на риска за даден актив и влиянието му върху риска за другите активи.

3.4 Предложение за експорт на основна информация за риска

Това е предложение да се реализира възможност за филтриране на чувствителната информация, която е събрана при изграждането на УРСИ за една организация и предоставяне на възможност за пренасяне на основни данни, свързани с оценката на риска към изграждане на УРСИ за друга организация. Миграцията трябва да включва информацията за уязвимостите на различни активи, заплахите към активите, съответстващите двойки заплаха-уязвимост, информация за влиянията на приложените методи за контрол върху нивото на риска и информация за влиянията между различните активи.

4. Описание на някои реализирани възможности

Създадена е информационна система, в която са реализирани повечето от предписанията на описания подход. Системата за управление на бази данни е реализирана на SQL сървър и съдържа 21 таблици и 39 съхранени процедури за управление на данните.

Основен елемент в изградената Информационна Система (ИС) е информационният актив. Информационният актив може да бъде човек, машина, документ, инфраструктура, програма, данни или всичко, което има информационна стойност за фирмата или има отношение към информация, ценна за фирмата.

За всеки актив трябва да бъде идентифициран собственик, за да се осигури отговорност и отчетност за актива. Собственикът на актива може да няма право да притежава актива, но е отговорен за неговото производство, разработка, експлоатация, използване и защита, когато е уместно. Собственикът на актива е често най-подходящото лице за определене на стоиността на актива за организацията [3].

Тъй като информационният актив може да бъде човек, то собственикът на информационен актив може да се явява актив на някой друг собственик.

В съответствие с предложената в използвания подход йерархична структура на въвеждане и контрол на рискивите фактори е реализирана схема на права за достъп до данните за активите и реализираните справки. В Таблица 1 е илюстрирана схема за осигуряване на достъп до данните от различните потребители на системата.

Таблица 1. Права за достъп до данните

Данни / Потребител	Данни за актив на собственик	Данни на подчинен собственик	Данни за текущия собственик	Данни за всички собсвеници	Данни за всички активи	Справки за актив	Справ-ки за всички активи
Ръководство				Четене	Четене	Четене	Четене
Отговорник по ИС, Експерт	Пълен	Четене	Четене	Четене	Четене, писане	Четене	Четене
Личен състав			Пълен	Пълен			Четене
Собственик на актив	Пълен	Четене	Четене	-	-	Четене	-

4.1 Основна информация за информационния актив.

Данните за информационния актив се въвеждат от собственика на актива и от неговите ръководители.

4.1.1 Въвеждането на нов актив

Въвеждането се осъществява на базата на вид актив и име на актива. При това вида на актива се избира на базата на таблица с предварително въведени вид и подвид на актива, които съответстват на препоръчаните от ISO/IEC 27005[3] видове. За всеки избран вид на актива може да се избира име на вече въведен актив от този вид, или да се въведе ново име на актив, което се записва в данните за този актив и се добавя към списъка с вече въведените имена за този вид. Имена на собственика на актива могат да се въвеждат или да се генерират от системата на база на автентикацията на въвеждащия. Предвидена е възможност информацията за всеки актив да може да се въвежда и на базата на вече въведен актив с възможност за модификация. Проверката за съвпадение с вече въведен актив се извършва на

базата на проверка за съвпадение на двойката име на актива / име на собственика. Изисква се за един собственик на актив да не се въвеждат два актива с изцяло съвпадащи имена.

4.1.2 Съхраняване на дати

В данните за актива се съхраняват два типа дати, които се генерират автоматично от системата. В първият тип е датата на първото въвеждане на данни за този актив. Вторият тип дати е свързана с датата на последната актуализация на данните за актива, както и на датите на предишни актуализации.

4.1.3 Съхраняване на нива на риска

На етапа на определяне на контекста на организацията се определя колко степенна е скалата за оценка на риска, както и приемливото ниво на риска за този актив. Тази информация се въвежда в системата от потребител с по-високо ниво на привилегия от това на собственика на актива.

4.1.4 Съхраняване на данни за извършени проверки

Важна група данни е свързана с алгоритъма за динамична промяна на нивото на риска за актива. В тази група са данните за периодите за препоръчителна проверка на състоянието на актива, датите на проверка на състоянието на актива, добавъчното ниво на риска от недобросъвестното поведение на собственика, както и общото натрупано добавъчно ниво на риска за този собственик. Съхранява се и информация за датите на проверка на състоянието на рисковите фактори за актива от по високо ниво в йерархията на организацията, от гледната точка на управление на риска.

4.1.5 Данни за влияния между активи

Друг проблем, който се опитва да реши програмният продукт, е свързан с взаимните влияния между активите. Този проблем е сложен и многопластов. Има две основни посоки за влияния между активите. Едната причина за влияния между активите е, когато един сложен актив е декомпозиран на по-малки активи, които си взаимодействат и си влияят по отношение на рисковите фактори. Другият случай на влияние е, когато два независимо работещи активи променят условията и / или ефективността на работа един на друг и променят рисковите фактори за другия актив. Това влияние може да е пряко върху работата на даден актив или влияние върху резултатите от неговите действия. Не е възможно да се отчетат и вземат предвид всички взаимовлияния между рисковете на различните активи.

За всеки влияещ актив в системата се запомня информация за вид актив, име на актив, ниво на риска и коефициент на влияние. Това е помощна информация, която може да се вземе предвид при определянето на окончателното ниво на риска за текущия актив. При дефиниране на алгоритъм за установяване на влиянието между активите, в системата могат да се добавят възможности за автоматично изчисляване на влиянието.

4.1.6 Данни за използвани методи на оценка на риска

Поради използване на много различни методи на оценка на риска и субективния характер на голяма част от оценките, автоматизирането на процеса е трудна задача с неоднозначно решение. По тази причина към момента в системата не е предвидено автоматично извършване на оценка на риска за актив. Предвидено е само въвеждане на свободен текст за характеризиране на използвания метод. Самата оценка се извършва извън системата, а резултатите от оценката се въвеждат, следят и актуализират в системата от собственика на актива.

4.2 Въвеждане и използване на рисковите фактори в системата

Основните рискови фактори, които участват в оценката на риска, са нивата на въздействието, заплахата и уязвимостта. Има различни формули, по които може да бъде изчислена стойността на риска за информационен актив, но в повечето от тях се използват тези рискови параметри. По тази причина в продукта се запомнят стойностите на тези променливи.

4.2.1 Въвеждане на данни за въздействието

Въздействието е израз на усилията и средствата, които фирмата трябва да приложи за възстановяване на нормалната функционалност на актив след реализиран инцидент с информационната сигурност. От тази гледна точка въздействието е съставено от стойността на актива плюс стойността на загубите от инцидент с този информационен актив. Втората съставна на въздействието може да бъде сведена до три основни стойности, които се отнасят до загуба на конфиденциалност, загуби от цялостност на актива и загуби от достъпност до актива, или до информацията, която е във връзка с актива. Системата позволява съхранение и обработка на три обобщени стойности на актива, които позволяват (участват в) последващо намиране на три стойности на риска, свързани с трите критерия – конфиденциалност, цялостност и достъпност.

4.2.2 Въвеждане на данни за заплахите

Според ISO/IEC 27000:2016 [4] заплахата е потенциална причина за нежелан инцидент, който може да доведе до увреждане на дадена система или организация. При оценка на риска заплахата представлява вероятността да се появи причина за инцидент и честотата, с която заплахата ще се опитва да атакува уязвимостта.

В системата заплахите за разглеждания актив се характеризират с вид на заплахата и описание. Като помощна информация се съхранява информация за произхода („преднамерена", „случайна", „природна"), това дали тя е предизвикана от външен или от вътрешен източник. За анализа е важно да има информация дали причинителят е човек и какви са мотивите му, както и какви са възможните последствия от действието на заплахата. За всяка заплаха се въвежда и използва информация за нивото на заплахата (при зададено максимално ниво). За опростяване на обработките приемаме, че ефектът от заплахата е еднакъв върху трите основни въздействия върху информационния актив - конфиденциалност, цялостност и достъпност. Приема се също, че влиянието на методите за контрол върху заплахите е много по-малко отколкото влиянието им върху уязвимостта и въздействието и от тази гледна точка считаме, че контролите не променят нивото на заплахата. Тези допускания намаляват значително броя на комбинациите на параметрите на взаимодействащите заплаха, уязвимост и въздействие.

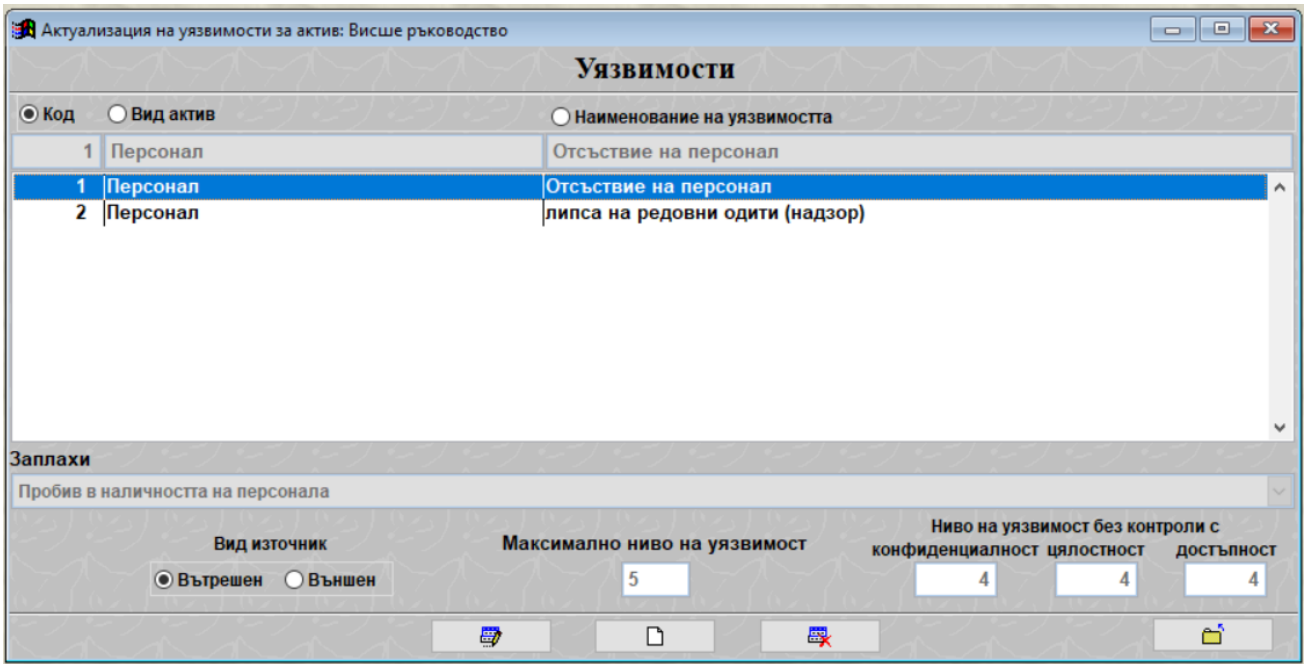
4.2.3 Въвеждане на данни за уязвимости

Уязвимостта е слабост на даден актив, която може да бъде използвана от една или повече заплахи [4]. Тя се характеризира с лекотата да бъде използвана. Колкото по-малка е слабостта, толкова по-малка е вероятността някоя заплаха да открие и използва уязвимостта.

В системата за всяка уязвимост се въвежда име и описание на уязвимостта. Името на уязвимостта е в пряка връзка с вида на актива, като на потребителя се предлага избор на вече въведени уязвимости за този вид актив, или въвеждане на ново име.

Връзката между уязвимост и заплаха се установява при въвеждане на всяка уязвимост, чрез избор на някоя от въведените заплахи за този актив. Възможен е изборът на опцията „няма заплаха". Такава уязвимост не участва в анализа на риска, но информацията за всички възможни уязвимости води до пълнота на данните и при следващи проверки на рисковите фактори, подсказва да се провери за поява на заплаха към тази уязвимост. Същото се отнася

и за възможността някоя заплаха да остане без съответстваща уязвимост, което означава, че тя не влияе на риска за актива. Възможностите за въвеждане на информация за уязвимости са показани на фигура 1.



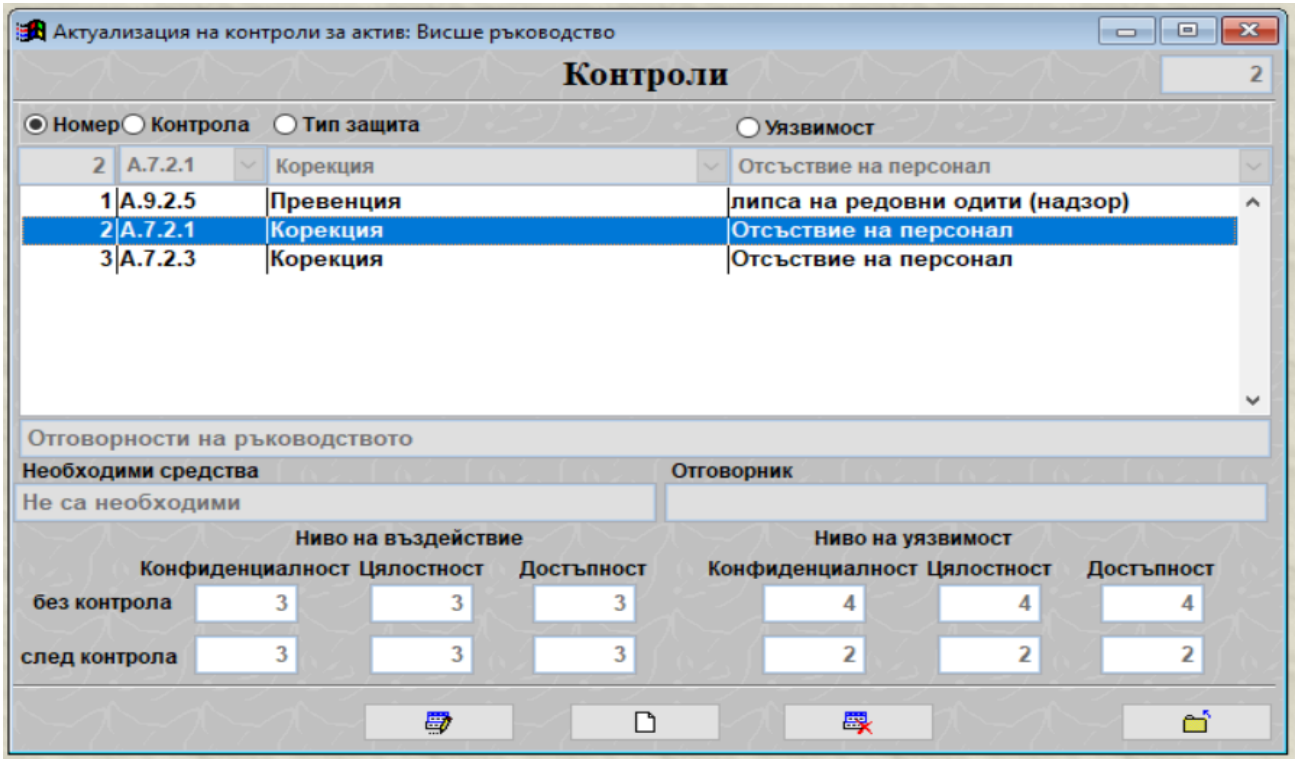
Фиг. 1. Въвеждане на информация за уязвимости

Въвежда се и ниво на уязвимостта, което има смисъл на степен на лекота за използване от заплахи, за предизвикване на неблагоприятни последствия, свързани с конфиденциалност, цялостност и достъпност. Има възможност за въвеждане и използване на три нива на уязвимост – преди прилагане на методи за контрол.

4.2.4 Въвеждане на данни за използвани механизми за контрол

Третиране на риска или въздействие на риска представлява процесът на изменение на риска [5]. Според ISO/IEC 27001:2017 [6] трябва да се определят всички механизми за контрол, които са необходими за реализиране на избраната опция за въздействие върху риска за сигурността на информацията [6]. Затова в продукта е предвидено да се въведе информация за механизмите за контрол, които се използват за защита на активите чрез намаляване на тяхната уязвимост и въздействието от инцидент. Въвежда се информация за номер на приложения механизъм за контрол (според „Приложение А” на използваната версия на стандарта ISO/IEC 27001) , типа на защита, необходимите финансови средства за реализция на предвидените мерки и кой отговаря за прилагане на механизъма за контрол.

Приехме, че механизмите за контрол не променят вероятността от заплаха. Следователно механизмите за контрол трябва да се прилагат към сцернарий за инцидент и въздействието за даден актив. За тази цел е предвидено да се съхранява информация за нивата на въздействие и описанието и нивата на уязвимост след прилагане на мерките за контрол. Предвидени са по три стойности, свързани с неблагоприятните последствия от загуба на конфиденциалност, цялостност и достъпност. Съхраняваната информация за използваните контроли е показана на фигура 2.



Фиг. 2.Въвеждане на информация за контроли

4.2.5 Данни за нивата на риска

Последната група данни за актива се отнася до нивата на риска за актива. Реализирано е автоматично изчисляване на нивата на риска на базата на нивата на въздействие, нивата на заплахи и нивата на уязвимост. Реализирана е възможност за пресмятане на нивата на риска преди и след прилагане на механизмите за контрол за всяка валидна двойка заплаха - уязвимост. За всеки използван метод за контрол се изчисляват по три стойности в зависимост от загубата на конфиденциалност, цялостност и достъпност. Така диференцирани данните за нивата на риска позволяват лесна преценка на влиянията на отделните рискови фактори и ясно отчитане на резултата от прилаганите мерки за контрол. Освен това има възможност за въвеждане или пресмятане на две интегрални величини на нивата на риска за този актив, преди и след прилагане на мерките за контрол. Това дава възможност за лесно съпоставяне на нивата на риска между различните активи и създава удобства в етапа на „преценка на риска”.

4.3 Реализирани справки

4.3.1 Справки за въведените активи

Реализирани са няколко справки, които са предназначени за ръководителите на организацията и дават информация за състоянието на най-стойностните и най-рисковите активи на организацията.

Направена е подобна справка за активите, която е предназначена за отговорника по сигурността на информацията в организацията и експерти, които съдействат на организациата. В тази справка има информация за заплахите, уязвимостите, нивата на риска, въздействие на риска, чрез въведените механизми за контрол и нивата на риска след приложеното въздействие.

4.3.2 Данни за идентифицирани заплахи

Справката, свързана с информация за заплахите, повтаря структурата на примерите за типични заплахи в ISO/IEC 27005 [3], като към вида, описанието и произхода на заплахата е добавена информация за източник на заплахата, произход на човешка намеса, мотивация, възможни последствия, вид и име на актива, към който е насочена, ниво на заплахата (%) за този актив. Справката извежда данни за всяка открита заплаха в базата данни на организацията.

4.3.3 Данни за въведени уязвимости на активите.

За уязвимости на всеки въведен актив се генерира справка, която съдържа информация за вид и име на актива; описание на уязвимостта; открити заплахи; ниво на уязвимоста (%) при загуба на конфиденциалност, цялостност и достъпност; приложени мерки за контрол.

Извежда се и справка за открити двойки заплаха-уязвимост с информация за използването и влиянието на приложените мерки за контрол. Тази справка е оформена като „План за въздействие върху риска за сигурността на информацията", който се изисква от ISO/IEC 27001:2017 [6] .

4.3.4 Справка за влияния между активите.

Натрупаната в системата информация за влиянията между различните активи може да бъде изведена във вид на справка, която съдържа информация за вид и име на влияещ актив; вид и име на повлиян актив; коефициент на влияние (какъв процент от стойността на външния риск се добавя към риска на текущия актив).

4.3.5 Справка за собственици на активи.

Във връзка с йерархия на нивата на достъп, въвеждане и контрол се изработва справка за имената на всички собственици и имената на всички активи от тип „персонал", за които отговаря всеки собственик. Така се описва йерархията на провеждане на проверки. Организационен интерес представлява и справката, в която за всеки актив се извеждат датите на проверка от собственика и датите за проверка от ръководителя на собственика.

4.3.6 Справка за следене на рисковите фактори и нивото на риска

Една от функционалностите на системата е свързана с възможността за проследяване на активността на всеки собственик на актив по отношение на изменения в рисковите параметри на работата на поверения му информационен актив. Активността на собственика се следи на базата на периодите, през които се извършват проверки на рисковите фактори за актива. На базата на съхранената история на извършените проверки системата извежда справка за всички собственици на активи и периодите, през които са извършени проверки, спрямо зададения граничен период. Оценката на спазването на зададените периоди на проверка се изразява чрез натрупания риск от поведението на собственика и прибавянето му към общото ниво на риска за актива. В генерираната справка се дава информация за нивото на добавъчния риск от поведението на собственика и динамиката на изменение на добавъчния риск. Изчисляването на натрупания добавъчен риск се извършва при всяка проведена проверка от собственика на актива и потвърждаване на статуса на риска за актива. Актуализация на този риск както и на общия риск за актива се извършва и при всяка генерирана справка за общия риск за актива.

4.3.7 Справка за натрупваната информация между системи на различни организации

Извеждат се справки за активи, нива на риск за актива, заплахи, уязвимости и влияния на контроли без чувствителна за организацията информация. Тази справка показва вида на

информацията, която може да мигрира към система за УРИС на друга организация. Тази справка може да се използва и като помощна информация при идентифициране на риска.

5. Заключение

Предложената програмна система предоставя значителни улеснения за процеса на УРСИ. Натрупването на по-пълна информация, необходима за УРСИ и предоставените справки могат да служат за лесно генериране на много от по-важните документи на СУСИ и да се използват от отговорниците и специалистите по информационна сигурност. Освен това системата създава условия за по-лесен контрол, като така мотивира персонала да спазва изискванията и политиките на Организацията в областта на ИС. Използваният подход за преодоляване на някои проблеми, свързани със събиране и обработка на информация [2], способства за по-лесно извършване на много от дейностите в УРСИ и за облекчено нарупване на полезна информация както за организацията, така и за външни експерти. Предложеният продукт може да стане по-ефективен, ако се разработи и внедри в него методика за изчисляване на зададените периоди за проверка, като функция на нивото на риска за актива и натрупаната информация за активността на собственика на актива, както и методика за отчитане на влиянието на активността на собственика на актива към цялостното ниво на риска за актива. Добър принос ще бъде и разработване на методика за изчисляване на влиянията между рисковете на различни активи.

Литература

[1] P.G.Genchev, M.N.Karova, Analysis of some issues in risk assessment for information security, Computer science and technologies (ACT 2019), pp. 62-70, Sept. 2019 (Conference proceedings)

[2] P.G.Genchev, Ideas (main guidelines) for building a methodology for organizing the process for assessment and monitoring of information security risk, submitted for publication. (Pending publication).

[3] Information technology - Security techniques - Information security risk management (ISO/IEC 27005:2018), // BDS_ISO_IEC_27005-2018-en.pdf

[4] Information technology — Security techniques — Information security management systems — Overview and vocabulary (ISO/IEC 27000:2016), // BDS_EN_ISO_IEC_27000-2017-bg.pdf

[5] Risk management – Guidelines (ISO 31000:2018), // BDS_ISO 31000-2018–bg.pdf

[6] Information technology — Security techniques — Information security management systems — Requirements (ISO/IEC 27001:2017), // BDS_EN_ISO_IEC_27001-2017-bg.pdf

За контакти:
ас. инж. Петко Генчев
Катедра „Компютърни науки и технологии”
Технически университет – Варна
email: pgenchev@tu-varna.bg

СРАВНИТЕЛЕН АНАЛИЗ НА CART, ID 3, C4.5 И C5.0 АЛГОРИТМИ. ПРЕДИМСТВА И НЕДОСТАСТЪЦИ

Мая П. Тодорова

Резюме: Машинното обучение е подраздел на изкуствения интелект, включващ методи за изграждане на алгоритми, които се обучават от данни. Дървото на решения (Decision trees) е начин за представяне на правила в йерархична последователна структура и е един от методите на машинното обучение за решаване на класификационни и регресионни задачи чрез обучение на класификатор. В статията са представени четири алгоритъма за създаване на модели чрез метода Дърво на решения - CART, ID 3, C4.5 и C5.0. Посочени са предимствата и недостатъците им. Описана е процедурата по изграждане на дърво на решения.
Ключови думи: Дърво на решения, CART, ID 3, C4.5, C5.0

Comparative Analysis of CART, ID 3, C4.5 AND C5.0 Algorithms. Advantages and Disadvantages

Maya P. Todorova

Abstract: Machine learning is a subdivision of Artificial Intelligence that includes methods for building algorithms capable of learning from data. The decision tree is a way of presenting rules in a hierarchical sequential structure. It is one of the methods of machine learning, for solving classification and regression tasks through classifier training. The article presents four algorithms for creating models using the Decision Tree method - CART, ID 3, C4.5 and C5.0. Their advantages and disadvantages are indicated. The decision tree construction procedure is presented.
Keywords: Decision tree, CART, ID 3, C4.5, C5.0

Увод

Дърветата на решения са мощен инструмент за машинното обучение. Класификационните дървета са метод, позволяващ определяне на принадлежност на обект към определен клас на категориалноно зависима променлива на база стойностите на една или повече предикаторни променливи.

Процесът по изграждане на дърво на решения се състои в последователно, рекурсивно разделяне на обучаващото множество на подмножества с използването на правила за вземане на решения във възлите. Започвайки от коренния възел, който представлява целия набор от данни, алгоритъмът избира функция, която е най-предсказуемата от целевия клас. Примерите се разделят на групи от различни стойности на база тази характеристика. Решението формира първия набор от клони на дървото. Алгоритъмът продължава да разделя и завладява възлите, да избира най-добрата характеристика, докато се достигне критерий за спиране.

Критериите за спиране са:

- Всички или почти всички примери във възел да са от един и същи клас;
- Да не съществуват характеристики, които да различават примерите;
- Нарастването на дървото до предварително определена граница за размер, като достигане на минимален брой примери в даден възел или максимална дълбочина [1].

Дървото на решения е алгоритъм за логическа класификация, основан на търсенето на конюнктивни закономерности, при която всеки вътрешен възел е тестова процедура, всеки клон е резултат от теста, а листата на дървото са търсеният клас [2].

За изграждане на дървото на решение в машинното обучение се използват различни алгоритми - ID3, CART, C4.5, C5.0, CHAID и др. Основни характеристики, които отличават алгоритмите, са критериите, по които се разделят данните във възел и използваните методи за намаляване на размерността на дървото. Алгоритмите на дървото за вземане на решения представляват контролирано обучение и като такива изискват предварително класифицирани целеви променливи. Наборът от данни за обучение трябва да бъде разнообразен и с малко на брой липсващи стойности. Целевите променливи трябва да имат дискретни стойности [3].

Конструиране на дърво на решение

Процесът на създаване на дърво на решения преминава през следните етапи:

- *Избор на атрибут, разделящ множеството*

Избраният атрибут трябва да разделя множеството от наблюдения във всеки възел, така че получените подмножества да съдържат примери от един и същи клас или да са максимално близки до класа. Броят на обектите, принадлежащи на други класове, във всяко от тези подмножества трябва да бъде възможно най-малък.

Основни критерии за избор на атрибут са:

- Ентропия на Шенон (Shannon Entropy)

Теоретико-информационният критерий се основава на концепцията на теорията на информацията. Ентропията е мярка, характеризираща количество информация, въведена от Клод Шенон [4]. Ентропията е мярка за „неопределеност“ за появата на дадено събитие [5].

Ентропията на Шенон за система с N възможни състояния се определя чрез формулата:

$$Entropy = -\sum_{i=1}^{n} \frac{N_i}{N} \log\left(\frac{N_i}{N}\right) \quad (1)$$

където:

- n – брой класове в първоначалното подмножество;
- N_i - брой примери в $i^{-ти}$ клас;
- N – общ брой примери в подмножеството.

Ентропията на дискретно събитие е максимална при равномерно разпределение и характеризира априорната неопределеност на дадена статистическа система. Най-добрият атрибут за разделяне ще бъде този, който ще осигури максимално намаляване на ентропията на полученото подмножество спрямо родителя.

- Индекс на Джини (Gini Index)

Атрибутът се избира на база разстояние между класовете [6].

$$Gini(D) = 1 - \sum_{i=1}^{j} p_i^2 \quad (2)$$

където:

- D – множество от данни;
- j – брой класове в D;
- p_i - вероятност за $i^{-ти}$ клас (вероятността в множеството D да има клас i).

- Информационна печалба

Дадено е множество D, което се разделя на подмножества D_1, D_2,….,Dn по стойностите на атрибут X. С |D| се обозначава броят на примерите в множеството D. Съществува

неопределеност относно принадлежността на всеки обект по определена стойност на атрибута X във възлите за разделяне.

Ако Info(D) е ентропията на множеството D, а Info(D_i) е ентропията на всяко от множествата D_i, $i \in [1, n]$, тогава критерият за избор на най-подходящ атрибут, по който ще се извършва проверката, се изчислява чрез формулата [7]:

$$Gain(X) = Info(\mathrm{D}) - Info_X(\mathrm{D}) \quad (3)$$

където:

$$Info_X(\mathrm{D}) = \sum_{i=1}^{n} \frac{|\mathrm{D}_i|}{|\mathrm{D}|} * Info(\mathrm{D}_i) \quad (4)$$

- Коефициент на печалба (Gain Ratio)

Коефициентът на печалба отчита не само количеството информация, необходимо за записване на резултат, но и количеството информация, необходимо за разделяне на текущия атрибут.

$$IntI(\mathrm{D}, X) = -\sum_{i=1}^{n} \frac{|\mathrm{D}_i|}{|\mathrm{D}|} log\left(\frac{|\mathrm{D}_i|}{|\mathrm{D}|}\right) \quad (5)$$

$$GainRatio(\mathrm{D}, X) = \frac{Gain(\mathrm{D},X)}{IntI(\mathrm{D},X)} \quad (6)$$

където:

IntI(D) - потенциалната информация, генерирана от разделянето на обучаващото множество D на подмножества D_i, използвайки атрибут X.

- *Избор на критерии за спиране*

Основните подходи, определящи дали даден възел да бъде разделен или отбелязан като листо, са:

- Ранно спиране - Използват се статистически методи за оценка на необходимостта от последващо разделение. Задава се прагова стойност като критерий за спиране.
- Ограничаване на дълбочината на дървото - определя се максимален брой възли в клоните на дървото.
- Задаване на минимално допустим брой примери в даден възел;
- Пълното дърво да има нулева грешка.

Задава се стойност на допустима величина за грешка и последващо действие - отсичане на клони.

- *Избор на метод за отсичане на клон*

Този етап е приложим, ако е създадено пълно дърво и са определени стойности за показателите точност и абсолютна грешка. Отсичат се клони от дървото, чието премахване не води до нарастване на грешката или до съществено намаляване на точността на модела [6].

- *Оценка на точността на построеното дърво*

Алгоритъм CART (Classification and Regression Tree)

Алгоритъмът създава двоични дървета за решаване на класификационни или регресионни задачи [7]. При конструирането на дървото на всяка стъпка правилото, образувано във възела, разделя обучаващия набор от данни на две части - частта, в която

правилото е изпълнено, е десен наследник и частта, в която правилото не е изпълнено, е ляв наследник. Изборът на оптимално правило се реализира чрез изчисляване на оценка за качеството на разделянето.

Изискванията за структурата на възел са:

- Всеки възел трябва да има препратки към двама потомци - вляво и вдясно, които са подобни структури;
- Възелът трябва да съдържа идентификатора на правилото, информация за броя или съотношението на примери от всеки клас от обучителната извадка, които са удовлетворили условието.

Оценъчната функция се базира на идеята за намаляване на примесите в един възел, реализирана чрез индекса на Джини [9]. Разделянето във всеки възел се осъществява на база на една променлива. Алгоритъмът за създаване на дървото, на всяка стъпка от изграждането му, последователно сравнява всички възможни разделяния за всички атрибути и избира най-добрия атрибут и най-доброто разделяне за него. Механизмът за подрязване на дърво е една от основните разлики между CART алгоритъма и другите алгоритми за изграждане на дървета. При CART алгоритъма изрязването е вид компромис между получаването на дърво с „подходящ размер“ и получаването на най-точната оценка за класификация[6].

Предимства на алгоритъма:

- Бързо изграждане на модел;
- Вграденият в алгоритъма механизъм за разделяне на дървото поставя данните без стойност в отделен възел и премахва шума от налични данни [10];
- Лесно се интерпретира - поради простотата на модела лесно се визуализира дървото и се проследяват всички възли;
- Позволява създаване на класификационно или регресионно дърво [6].

Недостатъци на алгоритъма:

- Алгоритъмът е нестабилен. Конструираното дърво, получено на база от данни от една извадка, е неприложимо към друга извадка [8,10].
- Неподходящ за изграждане на дърво с по-сложна структура.

Алгоритъм ID3 (Iterative Dichotomizer)

Идеята на алгоритъма се основава на рекурсивно разделяне на обучаващото множество, разположено в коренния възел на дървото на подмножества, използващи правила за решения. Коренният възел съдържа всички примери на обучаващото множество. Разделянето продължава, докато в получените подмножества не се съдържат само примери от един клас. Процесът на обучение спира и подмножествата се декларират като листа от дървото, съдържащи решенията. Всеки атрибут на обучаващото множество отразява свойство на класифицираните обекти.

Изборът на атрибут за разделяне във всеки възел се основава на увеличаване на информационната печалба или намаляване на ентропията [8,13].

Предимства на алгоритъма:

- Простота и интерпретируемост на класификацията [10];
- Бърза процедура за изграждане на дърво на решенията [8];
- Сложността на алгоритъм е линейна.

Недостатъци на алгоритъма:

- При голямо разнообразие на стойностите на атрибутите се наблюдава тенденция на преобучение [8];
- Когато всички стойности на атрибутите са уникални, се създава дърво, на което броя

на възлите е равен на броя на примерите;
- Не съществува възможност за обработка на числови атрибути и липсващи стойности [8,9,10];
- Целевата променлива трябва да бъде дискретна;
- Дърветата на решения, конструирани с помощта на този алгоритъм, са класифициращи.

Алгоритъм C4.5

Алгоритъмът е подобрена версия на алгоритъма ID3. C4.5 предоставя възможност за работа с числови атрибути [14], подрязване на клони и способност да се изгради дърво от непълна обучаваща извадка - някои атрибути нямат зададени стойности. Използва се за решаване на класификационни задачи [3].

Критерият $GainRatio(D)$ позволява да се оцени делът на полезната информацията, получена в резултат на разделянето и подобрява класификацията. При разделяне на множеството D трябва да има поне две подмножества, които да съдържат определен минимален брой примери. При неизпълнение на правилото се преустановява разделянето и съответният възел се обявява за листо. Ограничението може да породи ситуация, в която примерите, които попадат във възела, принадлежат към различни класове. Листото се асоциира с класа, който най-често се среща в него. При равни дялове на класовете като решение се определя най-често срещаният клас за непосредствения прародител на това листо.

Липсващите стойности за атрибута се разпределят пропорционално на честотата на възникване на съществуващите стойности.

Предимства на алгоритъма:
- Простота и интерпретируемост на класификацията;
- Сложността на алгоритъм е линейна;
- Алгоритъмът позволява използване на различни критерии за разделяне на възел и спиране;
- Позволява обучаващата извадка да съдържа липсващи стойности [13, 14];
- Възможност за работа с числови и дискретни променливи [14].

Недостатъци на алгоритъма:
- Когато атрибутите имат близки стойности, малки промени в данните могат да доведат до изграждането на напълно различно дърво на решения;
- При малък обем данни се получават лоши резултати.

Алгоритъм C5.0

Алгоритъмът е подобрена версия на C4.5. Използва се за на класификационни задачи в различни области. Може да създава двоично дърво или дърво с много разклонения [11]. Преди да бъде създадено дървото, е необходимо да се избере признакът, въз основа на който ще се реализира разделянето. Признакът трябва да осигури разделяне, което да доведе до получаване на възможно най-много групи, състоящи се от елементи от един и същи клас. Алгоритъмът C5.0 използва ентропията като мярка за чистотата на групата. Минималната стойност на ентропията е нула и показва, че извадката е напълно хомогенна. Максималната стойност за ентропията е единица и показва максималното количество от различни стойности. Алгоритъмът използва ентропия, за да изчисли промяната в хомогенността в резултат на разделяне на всяка възможна характеристика. Печалбата на информацията се изчислява като разлика между ентропията в сегмента преди разделянето и получените дялове

след разделянето. За намиране на общата ентропия във всички дялове след разделянето е необходимо да се извърши претегляне на ентропията на всеки дял спрямо съотношението записи, попадащи в този дял [1].

При възникне на ситуация, когато групите са твърде малки или има прекалено много точки на разклонение, се извършва подрязване на дървото. Резултатът от подрязването е намаляване на размера на дървото на решения.

Съществуват два вида подрязване: предварително и последващо. При предварително подрязване процесът на разклоняване ще бъде спрян след достигане на определен брой решения или възлите за решение съдържат само малък брой примери. Недостатък на този подход е, че дървото може да пропусне важни модели, които би научило, ако беше нараснало до по-големи размери. Алгоритъмът C5.0 използва последващо подрязване [15]. Премахват се възли и клони, които имат малък ефект върху класификационните грешки. В някои случаи цели клони се преместват по-нагоре по дървото или се заменят с по-прости решения.

Предимства на алгоритъма:

- Използва се за анализ на цифрови и номинални данни;
- Осигурява обработка на липсващи данни;
- Работи с малко количество данни в обучаващата извадка, но може да работи и с голямо количество;
- По-ефективен в сравнение с другите алгоритми за дърво на решенията.

Недостатъци на алгоритъма:

- Предубеденост на модела към функции с голям брой нива;
- Създаденият модел може да бъде преобучен или недоучен;
- Чувствителен към обучителната извадка. Малки промени в данните за обучение могат да доведат до големи промени в логиката на решение.
- Големите дървета могат да бъдат трудни за интерпретация [1].

В сравнение с C4.5 алгоритъмът C5.0 създава дървета на решение с по-малка размерност. Алгоритъмът е по-бърз, използва по-ефективно паметта на компютъра, има по-висока точност на резултатите и позволява автоматично премахване на незначителни атрибути [12,15].

Заключение

В статията са представени четири алгоритъма за създаване на дърво на решения - CART, ID 3, C4.5 и C5.0, като са посочени основните им предимства и недостатъци. Дървото на решение, като метод на машинното обучение, може да се използва за решаването на широк клас задачи в различни области на човешката дейност и има широк спектър от приложения: медицина, икономика, биоинформатика, техническа диагностика, компютърна лингвистика и др. Изборът на алгоритъм за създаване на дърво на решения зависи от вида на модела, който трябва да бъде създаден - класификационен или регресионен, от типа и обема на данните.

Литература

[1] Lantz Br., “Machine Learning with R”, ISBN 978-1-78216-214-8, pp.1 – pp.396, 2013

[2] Уткин Л.В., “Машинное обучение. Деревья решений “, стр.1 – стр.28

[3] Larose D, Larose Ch., „DISCOVERING KNOWLEDGE IN DATA An Introduction to Data Mining“, ISBN 978-0-470-90874-7 (hardback), Published by John Wiley & Sons, Inc., (2014)

[4] Замятин А., “ВВЕДЕНИЕ В ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ”, ISBN 978-5-94621-531-2, стр.1 – стр. 119, 2016

[5] Николенко С., Тулупьев А., „ Самообучающиеся системы”, ISBN 978-5-94057-506-1, стр.1 – стр. 289, 2009
[6] Богатырёв С., Симонова А., "Введение в добычу данных (Data Mining)", стр.1 – стр.34, 2006
[7] Rokach L., Maimon Od., "Data Mining and Knowledge Discovery Handbook” pp 165-192, 2005, DOI: 10.1007/0-387-25465-X_9
[8] Mohankumar M., Amuthakkani S., Jeyamala G, “COMPARATIVE ANALYSIS OF DECISION TREE ALGORITHMS FOR THE PREDICTION OF ELIGIBILITY OF A MAN FOR AVAILING BANK LOAN”, International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST) Vol. 2, Special Issue 15, 2016, pp. 360-366
[9] Hssina B., Merbouha Abd., Ezzikouri H., Erritali M., “A comparative study of decision tree ID3 and C4.5”, International Journal of Advanced Computer Science and Applications, Special Issue on Advances in Vehicular Ad Hoc Networking and Applications, pp. 13-19
[10] Singh S., Gupta Pr., „COMPARATIVE STUDY ID3, CART AND C4.5 DECISION TREE ALGORITHM: A SURVEY”, International Journal of Advanced Information Science and Technology (IJAIST) Vol.27, No.27, ISSN: 2319:2682, 2014, pp. 97 - 103
[11] Bernal G., Villegas L., Toro M., “SABER PRO SUCCESS PREDICTION MODEL USING DECISION TREE BASED LEARNING”, pp.1117–1121
[12] Nasaramma K., Lakshmi M., Kiranmayi D., “Prediction and Comparative Analysis of Students Placements Using C4.5 &C5.0”, International Journal of Computer Science and Information Technologies, Vol. 8 (3), 2017, ISSN: 0975-9646, pp. 367-368
[13] Priyama A. at al., “Comparative Analysis of Decision Tree Classification Algorithms”, International Journal of Current Engineering and Technology, Vol.3, No.2, ISSN 2277 – 4106, 2013, pp. 334-337
[14] Sharma H, Kumar S., “A Survey on Decision Tree Algorithms of Classification in Data Mining”, International Journal of Science and Research, Volume 5, Issue 4, ISSN (Online): 2319-7064, 2016, pp. 2094-2097
[15] Prof. Nilima Patil, Prof. Rekha Lathi, Prof. Vidya Chitre, “Comparison of C5.0 & CART Classification algorithms using pruning technique”, International Journal of Engineering Research & Technology, Vol. 1, Issue 4, ISSN: 2278-0181, 2012, pp. 1-5

За контакти: ас. Мая П. Тодорова
катедра „Софтуерни и Интернет технологии“
Технически университет –Варна
e-mail: maya.todorova@tu-varna.bg

ОБЗОР НА ПУБЛИКАЦИИ СЪС СРАВНИТЕЛЕН АНАЛИЗ НА МЕТОДИ И АЛГОРИТМИ ЗА КЛАСИФИКАЦИЯ НА ТЕКСТ В МАШИННОТО ОБУЧЕНИЕ

Нели Ан. Арабаджиева – Калчева

Резюме: В статията е представен обзор на публикации, свързани със сравнителен анализ на методи и алгоритми за класификация на текст в машинното обучение. Анализът на публикуваните научни изследвания показва, че класификацията на текст в машинното обучение е важна и актуална задача, намираща приложение в различни реални практически задачи. В голяма част от изследванията с най-добри характеристики на избрани метрики е методът на опорните вектори, нерядко следван от Наивния Бейсов класификатор.
Ключови думи: обзор на публикации, сравнителен анализ, методи и алгоритми за класификация на текст в машинното обучение, класификация на текст, машинно обучение

Overview of Publications with Comparative Analysis of Methods and Algorithms for Classification of Text in Machine Training

Neli An. Arabadzieva – Kalcheva

Abstract: An overview of publications related to comparative analysis of methods and algorithms for text classification in machine learning is presented in the article. The analysis of published scientific research shows that the classification of text in machine learning is an important and topical task, finding application in various real practical tasks. In most of the studies with the best characteristics of selected metrics is Support Vector Machines, often followed by the Naive Bayesian classifier.
Keywords: review of publications, comparative analysis, methods and algorithms for text classification in machine learning, text classification, machine learning.

1. Увод

През последните години голям брой изследователи прилагат различни методи и алгоритми за класификация в машинното обучение, сравняват ги с цел намиране на най-точния и ефективен от тях при работа както със структурирани, така и с неструктурирани данни.

Благодарение на засиления интелектуален интерес, наличието на по-мощен хардуер, увеличената наличност на документи в електронен вид и произтичащата от това необходимост от тяхното организиране, е отчетена нарастваща активност към класификацията на текст.

2. Класификация в машинното обучение

2.1. Същност на класификацията в машинното обучение

Класификацията като дейност разпределя предмети, процеси, обекти, видове, типове според някакви съществени признаци и ги разполага в определен ред, отразяващ степента на сходство помежду им.

Основна задача на класификацията е приобщаването на даден обект към предварително определен клас в съответствие с набор от признаци, които го описват – цена, тегло, качество, размери, количество и други характеристики. Признаците, които определят съществените за класификацията характеристики, се наричат предиктори. За да се реши задачата за класификация, е необходимо да се разработи алгоритъм за построяването на обобщен модел, който еднозначно съпоставя стойностите на предикторите с класа, към който принадлежи обектът.

Задачата за класификация на текст се състои в определяне на принадлежността на текста към даден клас въз основа на анализ на набор от предиктори, характеризиращи дадения текст.

2.2. Постановка на задачата за класификация

Формална постановка на задачата за класификация:

Нека:

- $X \in R^n$ – множество на обектите (входни данни)
- $У \in R$ - множество на резултатите (изходни данни)

Законът на разпределение $P_{XY}(x,y)$ не е известен, известна е само обучаваща извадка. Двойката (x,y) се представя като реализация на $(n+1)$ – мерни случайни величини $(X,У)$, зададени във вероятностното пространство.

$$\{x_1, y_1, x_2, y_2, \ldots\ldots, x_N, y_N\}, \quad (1)$$

където (x_i, y_i), $i=1, 2, \ldots N$ се явяват независими реализации на случайна величина (X, Y).

Целта е да се намери функция $Ф: X \rightarrow У$, която, изхождайки от стойностите на x, да предскаже y. Функцията $Ф$ се нарича решаваща функция или класификатор [4].

Преформулирана, формалната постановка на задачата за класификация на текст може да се представи така: нека е дадено крайно множество от класове $C=\{c_1,c_2,\ldots,c_k\}$ и крайно множество от документи $D=\{d_1,d_2,\ldots,d_k\}$. Целевата функция $f: D \times C \in \{0,1\}$ за всяка двойка <документ, клас> е неизвестна. Необходимо е да се намери класификатор f ‘, т.е. функция, максимално близка към функцията f. [3]

Основна задача на класификацията на текст е приобщаването на даден документ към определен клас в съответствие с набор от признаци. Традиционно в качеството на признаци се използват честотата на срещане на думите.

2.3. Оценяване на класификатори

Оценяването на класификатора се осъществява чрез сравняване на предсказания от модела клас за всеки обект от тестовата извадка с неговата действителна стойност. При съвпадение на двете стойности класифицирането се приема за правилно, при несъвпадение се отчита класификационна грешка.

Класификационният модел разпределя съответния обект в един от възможните класове: вярно класифициран (True) или невярно класифициран (False). При този модел са възможни четири варианта на класификация:

True Positive (TP) – действителен е клас Positive и вярно класифициран като клас Positive;

$$TP\ Rate = \frac{TP}{TP+FN} \quad (2)$$

True Negative (TN) – действителен е клас Negative и вярно класифициран като клас Negative;

$$TN\ Rate = \frac{TN}{TN+FP} \quad (3)$$

False Positive (FP) – действителен е клас Negative, а невярно класифициран като клас Positive;

$$FP\ Rate = \frac{FP}{FP+TN} \quad (4)$$

False Negative (FN) – действителен е клас Positive, а невярно класифициран като клас Negative.

$$FN\ Rate = \frac{FN}{FN+TP} \quad (5)$$

Мярката *точност* (*Accuracy*) се изчислява като отношението на правилно класифицираните примери към общия брой обекти от тестово множеството. Обикновено се представя в проценти.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (6)$$

Класификационната грешка (Error Rate) се изчислява като отношението на неправилно класифицираните примери към общия брой обекти от тестово множеството. Обикновено се представя в проценти.

$$Error\ Rate = \frac{FP+FN}{TP+FP+TN+FN} \quad (7)$$

Прецизността (*Precision*) е мярка, която показва доколко е точен класификаторът и дава процента на правилно класифицираните примери от съответния клас.

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

Мярката *чувствителност* (*Recall*) показва пълнотата или възвращаемостта на класификатора, т.е. колко от всички примери от даден клас е успял да определи като принадлежащи на този клас. Обикновено се представя в проценти.

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

Мярката *F-measure*, която комбинира Precision и Recall е тяхното претеглено хармонично средно. Обикновено се представя в проценти.

$$F\ measure = \frac{2}{{}^{1}/_{Recall} + {}^{1}/_{Precision}} \quad (10)$$

3. Обзор на публикации със сравнителен анализ на методи и алгоритми за класификация на текст в машинното обучение

Авторите Bo Pang and Lillian [12] сравняват точността на три класификатора: Мултиномиален Наивен Бейсов класификатор, максимална ентропия и метод на опорните вектори. За изследването са използвани случайно избрани 700 документа с позитивни настроения и 700 документа с отрицателни настроения от филмови ревюта. Данните са разделени на три равни по големина групи, като се е поддържало балансирано разпределение на класовете във всяка група. Получените резултати са средните трикратни стойности на точността от кръстосаното валидиране на тези данни. В това изследване изследователите са се съсредоточили върху характеристики, базирани на униграми (с отрицателно маркиране) и биграми. Резултатите при униграми са в полза на Мултиномиалния Наивен Бейсов класификатор (78.7%). Случаите, когато се отчитат биграми, резултатите на трите алгоритъма са сходни с разлики под 1 процент. Но когато са съчетали униграми с биграми,

методът на опорните вектори (82.7%) е отчел преднина пред другите два алгоритъма с около 2 процента.

Авторите в [2] изследват два основни проблема: анализ на текст при обработка на естествен език и намиране на най-добрия алгоритъм в машинното обучение. Разгледани са няколко от най-известните метода за класификация: метод на Бейс, метод на опорните вектори и метод на К най-близкия съсед. Проведените експерименти на класификация на колекция от текстове на английски, китайски и руски език доказват, че методът на опорните вектори има преимущество пред другите методи по точност и чувствителност. Методът на К най-близкия съсед е най-бърз, но с най-лоша точност и чувствителност. Методът на Бейс дава средни временни характеристики, както и връща като резултат средна точност и чувствителност.

Авторите на [10] правят обзор на следните класификатори на тест в машинното обучение за периода от 1980 година до 2010 година: Бейсов класификатор, дърво на решенията, метод на К най-близък съсед, метод на опорните вектори, невронна мрежа, алгоритъм на Рочио, хибриден алгоритъм и класификатор на логистичната регресия. Представен е анализ на методите за избор на характеристики и алгоритми за класификация. От проучването е потвърдено, че най-често използваните алгоритми: метод на опорните вектори, Бейсов класификатор, метод на К най-близък съсед и тяхната хибридна система с комбинация от различни други алгоритми и техники за избор на характеристики са най-подходящи в съществуващата литература като:

- Бейсовият класификатор се представя добре във филтрирането на спам и категоризацията на имейли, изисква малко количество данни за обучение, за да се изчислят необходимите параметри за класифициране. Бейсовият класификатор работи добре с цифрови и текстови данни, лесни за изпълнение в сравнение с други алгоритми, но допускането за условна независимост се нарушава от данни в реалния свят и се представя много слабо, когато функциите са силно корелирани.

- Класификаторът метод на опорните вектори е признат за един от най-ефективните текстови класификационни методи при сравнения на алгоритми за машинно обучение.

- Ако с метода на К най-близък съсед се използва подходяща предварителна обработка, тогава този алгоритъм може да постигне много добри резултати. Наивният Бейсов класификатор също постига добра производителност с подходяща предварителна обработка.

Авторът в [7] сравнява алгоритмите Наивен Бейсов класификатор и метод на К най-близкия съсед, използващ Евклидово разстояние при класификация на мнения на ползватели на интернет по различни въпроси. Резултатите показват много по-добра точност и чувствителност на Наивния Бейсов класификатор с повече от 10% от метода на К най-близкия съсед при векторен модел с униграми. При изследването с биграми тази разлика е повече от 12%.

Анализ на тоналността на текстове на руски и английски език на основата на методи на машинното обучение представят авторите в [9]. Целта на изследването е тестване и сравнение на различни методи на машинното обучение, от които: метод на Бейс, метод на К най-близкия съсед, метод Рочио, метод на опорните вектори, метод на класификация на основата на ключови думи и неговата комбинация с метода на опорните вектори. Във всички подходи теглата на думите в текста се определят по схемата TF.IDF и са изключени англоезични и руски „стоп-думи“. Използваните метрики за сравнения са точност, прецизност, чувствителност и F-мяра. Изследванията показват, че методът на опорните вектори и комбинираният метод дават най-добри резултати.

Разгледаният период на автора в [1] е от 2011 до 2016 година и са изследвани едни от най-разпространените методи за построяване на класификатори. Разгледаните методи са: метод на Бейс, метод на К най-близкия съсед, дърво на решенията, метод на опорните вектори, логистична регресия, метод на основата на изкуствените невронни мрежи, чиито

недостатъци и предимства са изложени. За сравняване на качествата на алгоритмите са използвани метрики: точност, чувствителност и F-мяра. Обучаващата и текстовата извадка са избрани в отношение 70/30. На основата на проведеното изследване авторът е стигнал до извод, че с най-добри характеристики на избраните метрики е методът на опорните вектори и невронната мрежа.

Авторите в [5] отделят следните подходи за определяне на тоналността на текст:

- на основата на използване на шаблони - Подходът се основава на генериране на правила, на основата на които се определя тоналността на текста. Способът се състои в разбиване на текста на думи или последователност от думи (N-grams). След това получените данни се използват за отделяне на често срещаните шаблони, на които се приписва положителна или отрицателна оценка. Отделените шаблони се приемат чрез правила от вида: if <условие> then <заключение>:

- машинно обучение без учител - Подходът се основава на идеята, че в текста с най-голяма тежест са термините, които често се срещат в него и в същото време присъстват и в неголеми количества във всички колекции. От отделените термини може да се направи извод и за тоналността за целия текст.

- машинно обучение с учител - В този подход е необходим статистически или вероятностен класификатор (например бейсов).

- хибриден метод - Даденият подход съчетава всичките или няколко от по-горе описаните принципи и се заключава в приложението им при създаването на класификатори при определяне на тоналността на текста.

В [5] авторите представят метод за класификация на текст по тоналности, основан на речник, съдържащ емоционална лексика. За изследването са използвани 15718 отзиви на ползватели на филми. Методът използва като оценка на работа на класификатора F-метрика и отчита резултат малко повече от 3% от метода на опорните вектори.

Емпирично проучване на три алгоритъма за класифициране на текст представят авторите в [14], използвайки два набора от данни: Наивен Бейсов класификатор, метод на опорните вектори и дърво на решенията (C4.5). Резултатите са сравнени въз основа на стойностите на прецизност и чувствителност за всеки един от алгоритмите. Изследванията показват, че от трите класификатора методът на опорните вектори е най-ефективен. Авторите доказват, че работата на метода на опорните вектори се влошава при класификация с малки набори от данни. Поради това авторите предлагат използването на хибриден метод на опорните вектори, който да подобрява съществуващите недостатъци на метода на опорните вектори.

Изследване на авторите в [15] има за цел да намери подходящия алгоритъм за автоматично класифициране на новинарска статия на индонезийски език. Документите първоначално са подложени на лематизация и премахване на стоп-думи, с цел минимизиране на шума в документите. Алгоритмите Мултиномиален Наивен Бейсов класификатор, Бернулиев Наивен Бейсов класификатор и метод на опорните вектори са сравнени и оценени преди и след прилагане на TF.IDF функция. Въз основа на резултатите от теста, комбинацията TF.IDF и Мултиномиален Наивен Бейсов класификатор дава най-добър резултат за класификация на новини в индонезийски език (98.4%), следвана от TF.IDF + Бернулиев Наивен Бейсов класификатор (98.2%).

Авторите в [11] определят като най-значими за изследване следните пет класификатора: дърво на решенията, метод на опорните вектори, метод на К най-близък съсед, Наивен Бейсов класификатор и скрит Марков модел. Всички тези алгоритми са модифицирани от много изследователи, за да се получи висока точност. Целта на тяхното изследване е цялостно проучване за почти всички изменения, които са направени по тези пет алгоритъма за класификация на текст. Трудността за сравнението между тях произтича от факта, че много изследователи са ги прилагали на лични данни. Тези изменения са

класифицирани според обучаемия (модификация), основен алгоритъм (модификация и добавяне) и характеристики (извличане и редукция). Изследователите осъществяват сравнение между тези модификации за всеки алгоритъм. Проучването показва, че най-лесният начин за подобряване на класификацията е чрез намаляване на броя на характеристиките, което води до повишаване на ефективността.

Според авторите в [13] най-добрите алгоритми за класификация са: Наивен Бейсов класификатор, Случайна гора и метод на опорните вектори и именно поради това много изследователи ги комбинират за да подобрят точността. Изследователите са тествали тези алгоритми със софтуера Rapid Miner с 500 примерни данни. Резултатите, които получават, показват, че методът на опорните вектори е с най-висока точност.

Авторите в [6] провеждат експерименти с цел изследване приложимостта на алгоритмите за машинно самообучение за учене на многоетикетни класификатори на емоционално зареден текст. Изследвани са и корелациите между различните емоции с идеята, че те могат да доведат до подобряване на точността на класификация. Данните представляват мнения, които са събрани от една от големите социални мрежи в уеб – Twitter. За оценяване на модела са използвани основните метрики в машинното самообучение – прецизност, чувствителност и мярката F-measure, която е претеглено хармонично средно на предходните две. При експеримента за трансформация на задачата са използвани следните алгоритми за машинното самообучение от тип учене с учител: Наивен Бейсов класификатор, логистична регресия и метод на опорните вектори. Използван е вариантът на полиноминалния Наивен Бейсов класификатор, който е подходящ за дискретно множество от атрибути. При оптимизирането на класификаторите методът на опорните вектори дава най-добри резултати, а за двоична проверка класификаторът с най-високи резултати е Наивният Бейсов класификатор.

Авторите в [3] сравняват точността на девет алгоритъма на машинното обучение при класифициране на потребителски мнения на английски език. За изследването е избран набор от данни, взет от стандартния модул NLTK - Natural Language Toolkit, който е open source библиотека на Python. Данните, които представляват филмови рецензии, са предоставени през 2004 година от Bo Pang и Lillian Lee (Department of Computer Science. Cornell University. Ithaca, NY). Представен е нов алгоритъм, базиран на Наивния Бейсов класификатор, използващ Лапласово разпределение и именуван Лапласов Наивен Бейсов класификатор. Проведените изследвания показват, че точността на Лапласовия Наивен Бейсов класификатор е най-висока и е съизмерима с точността на метода на опорните вектори. Всички алгоритми, използвани в сравнението: Лапласов Наивен Бейсов класификатор, Гаусов Наивен Бейсов класификатор, Бернулиев Наивен Бейсов класификатор, Мултиноминален Наивен Бейсов класификатор, метод на К най-близък съсед, метод на опорните вектори, дърво на решението, ансамблов алгоритъм Случайна гора и ансамблов алгоритъм Адаптивен усилвател (AdaBoost), увеличават точността си при увеличаване обема на данните.

Авторът в [8] осъществява сравнителен анализ на точността на алгоритми за класификация на български текст в машинното обучение. Данните, използвани в изследването, представляват извадка от стихотворения на български език на автори, живели и творили през 19 и 20 век. Изследваните алгоритми са: Бернулиев Наивен Бейсов класификатор, Мултиномиален Наивен Бейсов класификатор, дърво на решенията (C4.5), метод на К най-близкия съсед и метод на опорните вектори, използващ оптимизация с два Лангранж множителя и полиномиална функция на ядрото. Проведените изследвания показват, че при класификация на два класа с най-висока точност е алгоритъмът дърво на решенията, която намалява при по-голям обем от данни. При четири и осем класа най-точен е Мултиномиалният Наивен Бейсов класификатор. При 20 класа Бернулиевият Наивен Бейсов класификатор и методът на опорните вектори с оптимизация са с най-високи

стойности спрямо другите алгоритми. Методът на К най-близкия съсед показва най-ниски резултати в цялото изследване.

Заключение

Анализът на публикуваните научни изследвания показва, че класификацията на текст в машинното обучение е важна и актуална задача, намираща приложение в различни реални практически задачи, което води до следните изводи:

- Изследванията показват, че няма цялостен универсален метод или алгоритъм за класификация на текст в машинното обучение. Всеки метод или алгоритъм работи добре в зависимост от спецификата на поставената задача и от използваните данни.
- При задачата за класификация е важно предварително да бъдат определени критериите и мерките, с които ще бъдат оценявани използваните методи и алгоритми.
- Целта на сравнителния анализ на повечето изследвани научни статии е да се провери кой от алгоритмите класифицира с най-висока точност при различните данни и при различни условия.
- Някои изследователи използват собствен набор от данни за тестване на методи и алгоритми за класификация в машинното обучение, което прави сравнението по-трудно.
- В голяма част от изследванията с най-добри характеристики на избрани метрики е методът на опорните вектори, нерядко следван от Наивния Бейсов класификатор.

Литература

[1] Батура, Т. Методы автоматической классификации текстов. Программные продукты и системы 30, no. 1, 2017.

[2] Зайцев, В., Л. Линь. Способы повышения эффективности классификации документов для конечного множества языков. Вісник НТУУ «КПІ» Інформатика, управління та обчислювальна техніка №50, 2009, с. 100-104.

[3] Калчева-Арабаджиева Н., Н. Николов, Класифициране на мнения от потребителски коментари на английски език чрез алгоритми на машинното обучение, V-та научна конференция с международно участие "Компютърни науки и технологии", 28-29 септември 2018 г., Списание Компютърни науки и технологии, Варна, България, том 1, стр. 120-128, 2018, ТУ – Варна, България

[4] Калчева Н., Матева З., Бейсова теория в машинното обучение, Годишник на ТУ Варна 2016, стр. 130-133

[5] Клековкина, М., Е. Котельников. Метод автоматической классификации текстов по тональности, основанный на словаре эмоциональной лексики. Труды, 2012, с. 118-123.

[6] Новаков Т., И. Койчев. Автоматично разпознаване на емоции в текст. XI Национална конференция „Образованието и изследванията в информационното общество” 2018, Пловдив, 2018.

[7] Хоан, Н. Анализ тональности текстов на основе методов машинного обучения. https://docplayer.ru/60280325-Mashinnogo-obucheniya-nguen-zy-hoan.html

[8] Arabadzieva – Kalcheva, Neli An., Comparative Analysis of Algorithms for Classification of Text in the Bulgarian Language in Machine Learning, International Conference “Applied Computer Technologies” ACT 2018 – Ohrid, Macedonia, pp. 6 - 11

[9] Feng, M., G. Wu. A distributed chinese Naive Bayes classifier based on word embedding. International Conference on Machinery Materials and Computing Technology (ICMMCT 2016), 2016, pp. 1121-1127.

[10] Khan, A., B. Baharudin, L. Lee, K. Khan. A review of machine learning algorithms for text-documents classification. Journal of Advances Information Technology, vol. 1, 2010, pp. 4-20.
[11] McCallum, A., K. Nigam. A comparison of event models for Naive Bayes text classification. Papers from the 1998 AAAI Workshop, 1998, pp. 41-48.
[12] Pang, B., L. Lee, S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing, vol. 10, Association for Computational Linguistics, 2002, pp. 79-86.
[13] Sheshasaayee, A., G. Thailambal. Comparison of classification algorithms in text mining. International journal of pure and applied mathematics 116, no. 22, 2017, pp. 425-433.
[14] Trivedi, M., S. Sharma, N. Soni, S. Nair. Comparison of text classification algorithms. International Journal of Engineering Research & Technology (IJERT) 4, no. 02, 2015.
[15] Wongso, R., F. Luwinda, B. Trisnajaya, O. Rusli, Rudy. News article text classification in indonesian language. Procedia Computer Science, vol. 116, 2017, pp. 137- 143.

За контакти:
д-р инж. Нели Ананиева Арабаджиева - Калчева
катедра „Софтуерни и интернет технологии”
Технически университет - Варна
E-mail: n_kalcheva@tu-varna.bg

СОФТУЕРНИ И ИНТЕРНЕТ ТЕХНОЛОГИИ

OVERVIEW OF THE METHODS USED FOR OPINION MINING AND SENTIMENT ANALYSIS AND THEIR APPLICATION IN BULGARIAN LANGUAGE SO FAR

Daniela Iv. Petrova

Abstract: In recent years there is a lot of attention towards the web documents as a new source for people's opinion and experience. This leads to a growing interest in the automated extraction and sentiment analysis of web documents, such as people's feedback on products or services, blogs or comments on news. The purpose of this paper is to give an overview of the different methods and approaches for opinion mining and sentiment analysis for English language texts. It also tries to cover some techniques and approaches that are used in sentiment analysis in Bulgarian language documents.
Keywords: opinion mining, sentiment analysis, text mining, text classification, Bulgarian language

1. Introduction

Human life is filled with emotions and opinions and they define the way people act and interact with each other. Emotions influence the way people think and react. But these feelings and beliefs are foreign for computer programs and they cannot define them alone. Due to the accumulation of a huge amount of user reviews, impressions, inquiries and questions in social media, blogs and web applications, opinion mining and sentiment analysis have become very popular. The analysis of such texts and documents is very important in marketing, advertising, as well as assessment of people's moods and attitudes on politics and even on security point of view. In the present opinion mining and sentiment analysis are scientific fields in the crossroad between information retrieval (IR) and natural language processing (NLP), as well as some other disciplines as text mining and information extraction. [6]

The term "opinion mining" is first introduced by Dave, Lawrence and Pennock. They define it as "a process of processing the search results for a certain element and generating opinions about its product characteristics". Sadegh, Ibrahin and Othman evolve the idea that those are methods for finding and extracting subjective information from texts. Opinion mining is done through natural language processing by defining the perceptions, views and understandings on a given topic, while the automated approach aims to extract the features of the object and thus to determine whether the text is positive, negative or neutral.[8]

Dash, Chen and Tong are the ones to use the term "sentiment analysis" in automatic text evaluation for the first time. In the majority of researches opinion mining and sentiment analysis are used as synonyms and very often they overlap.[8]

2. Overview and classification of the methods used for sentiment analysis and opinion mining

A thorough research on the different methods, used for sentiment analysis is done by Bing Liu. In his book Sentiment Analysis and Opinion Mining (2012) he divides the approaches based on the parts that are analyzed – a word, a sentence or a whole document. (Fig.1)

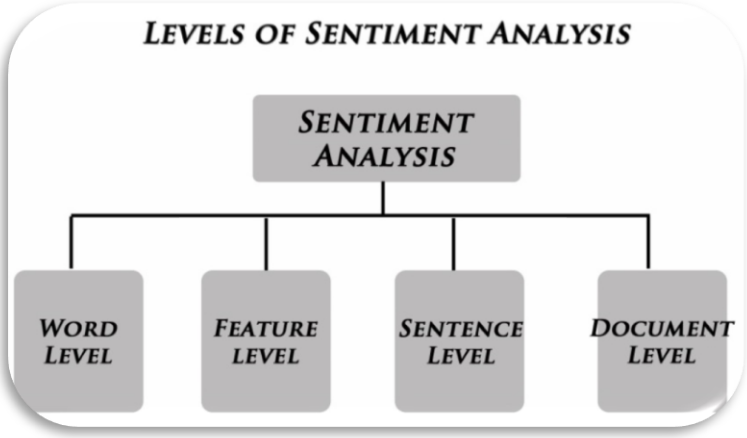


Fig. 1. Levels of Sentiment Analysis

- **Document level** - On a document level the task of the methods is to classify the overall sentiment of the document as positive or negative. This level of analysis assumes that every text or document gives and opinion only on one subject. For this reason it is not applicable for texts that give opinion or compare a couple of products.
- **Sentence level** – On this level of analysis the task is to define the sentiment (positive, negative or neutral) in every sentence. This level is closely linked to subjectivity classification, which divides the sentences on such that have factual information on the subject and those that have the subjective opinion of the users. There isn't an obvious line that separates the different sentences, as a sentence that has factual information could also imply an opinion. For example: "We bought the umbrella this month, but it was thorn in the first wind."
- **Entity/aspect level** – This level of analysis is required due to the fact that the analysis on document and sentence level cannot define precisely what people like or don't like. The analysis on an aspect level is far more precise. Here, instead of considering the different language constructions such as document, paragraph or sentence, is searched for the opinion itself. This analysis perceives the idea that the opinion consists of a sentiment (positive or negative) and an object of the sentiment. For instance, although the sentence "Although the cleaning isn't good enough, I still love this hotel." obviously has a positive tone, it cannot be stated as totally positive. It is positive for the hotel, but negative for the cleaning. The task of this level of analysis is to find the sentiment for every subject. Another example is the sentence "The computer is fast, but the life of its battery is short". Here the subjects are two – the computer and the battery and the sentiments are respectively - positive and negative. The result of this level of analysis is a structured summary of the sentiments for each subject, which turns the unstructured text into structured data that can be used for all kids of quality or quantity analysis.
- **Word level** – The words are the most important indicators of sentiment. The words that express feelings are common positive or negative sentiment words such as: good, great, excellent for positive and bad, terrible, awful for negative. In addition to single words, there can be phrases and idioms. Those phrases and idioms are gathered in lexicons, where they are classified in accordance to their orientation: positive or negative.

There are three types of methods to create a sentiment lexicon: manual, corpus-based methods and dictionary- based methods. The manual construction of a lexicon is time-consuming and hard work that should be always combined with some of the other methods to enhance their precision. Corpus-based methods always start with a seed set of sentiment words with defined polarity and use the syntactic models or search for their reappearance in order to identify new sentiment words and their polarity and widen the corpus. In their work Hatzivassiloglou and McKeown are the first to

speak about defining the orientation of the words. They develop a graph-based technique for lexicon learning using a corpus. Another algorithm for creating a lexicon is the one of Turney and Littman. They define whether a word is positive or negative and how strong is the opinion by computing the point wise mutual information of the words for their appearance in a small set of positive and negative seed words. The corpus-based methods can create relatively accurate lists of positive and negative words. But these methods need large sets of labeled data for learning. The dictionary-based methods can overcome those disadvantages by using lexicographic sources for their primary source for semantic information of the words. They don't need a huge data base or special search engines, but use the available resources, like WordNet for instance. These methods use a manually constructed list of sentiment words and their polarity, and then it searches the dictionary for synonyms and antonyms and thus widens the start list. After that the new list is used for generating new sentiment words. The major disadvantage of these methods is that it cannot make distinction in the orientation of the word according to the context, in which it is used. A word can have a positive sentiment in one context and negative – in another. For instance the word "large" can be positive, when speaking of a "large TV screen" and negative, when speaking of the dimensions of a phone. [6]

Although sentiment words and phrases are very important for the analysis, they are not sufficient for the final result, because of the complexity of the problem. Parts of the reasons are that: a positive or negative word may have another meaning depending on the context; sentences that have a word from the lexicon may not express sentiment at all. For example the sentences "Can anybody tell me which food dehydrator is better?" and "Does someone know from where to buy a tray for this awful food dehydrator?". Both sentences have sentiment words, but only the second one carries some sentiment for the object. There is also the problem with sarcastic sentences, where positive words can have negative meaning and vise versa – "Incredible dehydrator! It stopped working on the second day!". Furthermore many factual sentences that don't contain words from the lexicon may have sentiment – „we used the pool only once and it started leaking!". There are no sentiment words in this sentence, but its overall sentiment is negative. These are only some of the problems that sentiment analysis faces.

In the base of all researches, the most important for sentiment classification is the choice of effective features, the so called feature extraction. They can be the following (fig. 2)

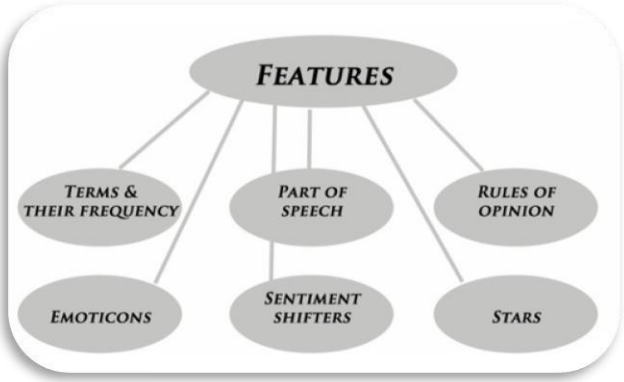


Fig. 2. Feature extraction

- **Terms and their frequency**. These are individual words (unigrams) and their n-grams with the respective frequency counts. They are the most commonly used features in text classification at all. Terms turn out to be highly effective for sentiment classification as well. [1]
- **Part of speech**. It is of great relevance what part of speech is the given word. Words can be treated differently depending on their type. Adjectives, for example, are important mood carriers: good, wonderful, excellent or bad, terrible, awful. Some researchers take into

consideration only the adjectives. Others analyze all types of part of speech. Nouns like "garbage" or verbs like "love, like, hate" can also be mood carriers. Apart from individual words, some phrases and idioms can also be used to express feelings. They are very specific for every language and cannot be unified. Such expression in Bulgarian can be: „Струва майка си и баща си!" [1]

- **Rules of opinion**. Aside from individual words and phrases, there can be other expressions and language constructions that can express sentiment or opinion. They can be simple – of one word, or composite expressions, for which should be used common sense or knowledge in the field in order to evaluate their sentiment. One way to present these rules is the idea for compositional semantics (Dowty, Wall и Peters, 1981), where the meaning of one composite expression is a function of the meanings of its parts and the syntax rules that connect them. [1]
- **Sentiment shifters**. These are expressions that are used for shifting the sentiment. They should be treated with caution since they do not all shift the sentiment. Such expressions are negative words like "never, no one, nowhere, barely" and others. In the sentence "The machine barely works" "works" is a positive word, but "barely" shifts its sentiment to negative nuance. The sarcasm could also change the mood- "What an amazing dishwasher, it doesn't work from the first day!". Although it is not difficult to spot such sentiment shifting phrases, their detection and processing by automated systems is a challenge. [1]
- **Emoticons**. Very often in social media posts are used emoticons that can also be used for classification of texts into positive or negative. In his work, Ashgar (Twitter sentiment analysis framework using hybrid classification scheme) offers a hybrid system for sentiment analysis of twitter posts and finds out that using emoticons can increase the accuracy of the classification with up to 6%. [1]
- **Stars in customers' reviews**. Some applications allow their users to give opinion using stars. They can be used relatively easy to categorize the opinions in three: positive (4 or 5 stars), neutral (3 stars) and negative (1 and 2 stars). [1]

Another way of classifying the sentiment analysis methods is the way the system is trained: 1. Supervised learning and 2. Unsupervised learning.

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. The most commonly used supervised learning algorithms in sentiment analysis are:

- Naïve Bayes classifier – despite its simplicity, the Naïve Bayes classifier is a popular method for classification of texts and is very effective in different spheres. During working, Naïve Bayes takes a stochastic model for generating a document. Using the rule of Naïve Bayes this model is inverted to predict the most probable class of the new document.[10]
- Support vector machines (SVM) – it is a statistical classification method, first introduced by Vapnik (1995). This model could be used for binary and multi categories classification. SVM seeks for the hyperplane, represented by vector w that divides the positive from the negative vectors with an optimal margin. [10]
- Cross-domain sentiment classification –it is applying a sentiment classifier trained using labeled data on a particular domain to classify sentiment of reviews on a different domain. This often results in poor performance because words that occur in the train domain might not appear in the test domain.
- Cross-language sentiment classification - aims to utilize annotated sentiment resources in one language (typically English) for sentiment classification of text documents in another language.

- N-gram based character language model – this is a new model in natural language processing that originates from the N-gram language models. For the basic feature of its algorithm it uses signs (letters, spaces and symbols) instead of words.

Despite its way of work, all supervised learning models have the following algorithm: (fig.3)

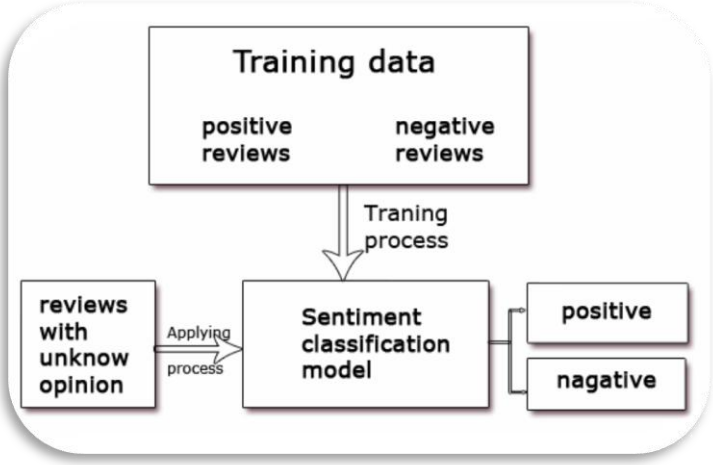


Fig. 3. Algorithm of supervised learning methods

In their work Bo Pang и Lillian Lee, compare the effectiveness of a couple of supervised learning techniques for sentiment classification of documents. They use Naïve Bayes classifier, maximum entropy classification and support vector machines, to categorize movie reviews. In the course of their work they try out using unigrams and bigrams, as well as different parts of speech and its position in the text to determine which base gives the best results. From their research can be concluded that Naïve Bayes classifier gives the worst results, while SVM gives the best results – over 81% accuracy. They determine that for more precise results should be explored the focus of every sentence in the document. [4]

Rudy Prabowo and Mike Thelwall offer a hybrid approach for sentiment classification in a couple types of documents – movie reviews, product reviews and comments in MySpace. Before using SVM they apply different classifiers, based on rules and study the effectiveness of the different combinations. There are different results for the different types of data, but the optimal turns out to be RBC (rules based classifier) ->SBC (statistics based classifier) followed by SVM. [5]

As opposed to supervised learning, which has proven its effectiveness, but requires a large set of labeled data and takes a lot of time and resources, unsupervised learning methods could avoid the flaws of supervised learning. Unsupervised learning is in fact an output function that describes the hidden structure of the unlabeled data. The approaches used are cluster based:

- Spectral clustering,
- k-means clustering,
- Hierarchical clustering.

Clustering is close to classification (supervised learning), but the clusters are not predefined and are determined by the data itself. **Spectral clustering** is used in many different spheres as statistics, machine learning, pattern recognition, data mining and image processing. It works on the base of Eigen structure of similarity matrix. The clusters are formed by dividing the dots of data using a similarity matrix. In **k-means clustering**, the data set is divided in k clusters (K is defined by the user and is a fixed number). This algorithm is easy to understand, but its results depend on the original centroid which not always brings to high efficiency. The **hierarchical clustering** algorithm groups that data in a tree. It can be two types: agglomerative or separating. In the agglomerative approach every observation starts in its own cluster and the couples are merged when

the observation is moving up in the hierarchy. In the separating approach, all observations start in one cluster and are separated recursively when moving down in the hierarchy. [9]

All the methods and approaches stated above could be summarized in the following figure: (fig.4)

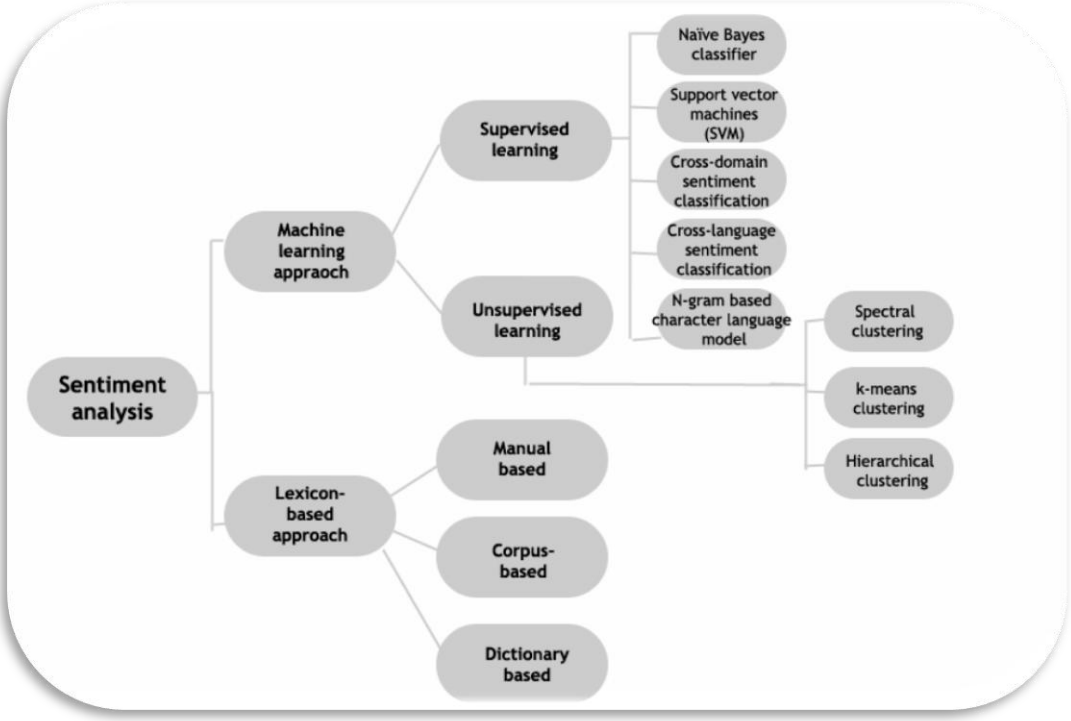


Fig. 4. Sentiment analysis methods classification

3. Different approaches used for Bulgarian language texts

To the knowledge of the authors there are not many researches done of sentiment analysis on Bulgarian texts. The complexity of Bulgarian language makes it difficult but interesting object of research and there is still necessity for research in this direction. This section of the overview makes an attempt to summarize the previous studies.

Maybe the first to automatically classify Bulgarian adjectives are Boris Kraychev and Ivan Kochev. In their work they are presenting a method for automatic classification using pre-selected emotional axes like love-hate, generosity-greed, goodness-evil and others. They use an unsupervised learning method with the primary data source – the frequency of occurrence of words in documents from the index of the popular search engine Bing. They choose Bing before Google, because it provides information for the number of documents that contain a given word and the number of documents where two words are found in a nearby. [3]

There are several works on Bulgarian language texts that are not exactly connected with opinion mining but could be used as a basis for a sentiment analysis. Such is the semantic classification of adjectives in the Bulgarian WordNet by Tsvetana Dimitrova and Valentina Stefanova. Their classification is based on the information that is already available in WordNet from other synsets (noun, verb and others) that are linked to the adjective synsets via lexicon semantic relations [2]. Another work is the dissertation of Ivelina Stoyanova about automatic detection and tagging of phrases in Bulgarian language. According to her, multiword expressions are a major part of the lexical system of the language and in the same time they are difficult to detect by an automatic system. Dealing with the problems connected with multiword expressions will enhance significantly the results of different natural language processing applications, including sentiment analysis. [7]

4. Conclusion

This paper attempts to make an overview of the different approaches and methods used for sentiment analysis and opinion mining and their classification. It also makes a review of the previous studies made on Bulgarian language. To the authors knowledge they are not many and it shows that there is place and need for research in the field of sentiment analysis and opinion mining regarding texts in Bulgarian language.

References

[1]. Bing L. Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012, p.31-35

[2] Dimitrova Ts., V. Stefanova "The semantic classification of adjectives in the Bulgarian WordNet: towards a multiclass approach" , Cognitive Studies 18, Warsaw 2018

[3]. Kraychev B., I.Koychev. "Automatic classification of Bulgarian adjectives on emotional semantic axes", VII national conference "Education and ……"p.121-130

[4]. Pang Bo, L.Lee. Thumbs up? Sentiment Classification using Machine Learning Techniques, Proceedings of EMNLP, 2002, p.79-86

[5]. Prabowo R., M. Thelwall. Sentiment analysis: A combined approach, Journal of Informetrics 3, 2009, p.143-157

[6]. Sadegh M., I.Roliana, Z.Othman. Opinion Mining and Sentiment Analysis: A survey, International Journal of Computers and Technology, June 2012, p.171-175

[7].Stoyanova I.."Автоматично разпознаване и тагиране на съставни лексикални единици в българския език", БАН, Sofia, April 2012

[8].Todorka Atanasova, Nadezhda Filipova, Snezhana Sulova, Yanka Alexandrova, Julian Vasilv. Intelligent analysis of students dataset, Publishing house "Knowledge and business", 2019

[9]. Unnisa M., A.Ameen, S.Raziuddin. Opinion Mining on Twitter Data using Unsupervised Learning Technique, International Journal of Computer Applications, August 2016,p.12-19

[10]. Ye Q.,Z.Zhang, R.Law. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches, Expert Systems with Applications 36, 2009, p.6527-6535

For contacts:
Daniela Petrova
Software and Internet Technologies
Technical University of Varna
E-mail: daniela.petrova@tu-varna.bg

МАШИННОТО ОБУЧЕНИЕ ЗА ИЗСЛЕДВАНЕ НА ДИСТРЕС ПРИ ПАЦИЕНТИ С ОНКОЛОГИЧНИ ЗАБОЛЯВАНИЯ

Гинка К. Маринова

Резюме: Ранното диагностициране на дистрес е от важно значение в първоначалния етап на лечение и е свързан с подобряване качеството на живот на пациентите. Техниките на машинното обучение са подходящи за анализ на медицински данни и изследване на поставената задача. В настоящата статия е направен сравнителен анализ на получените резултати на четири метода за класификация на набор от медицински клинични психологични диагностични данни за разпознаване на дистрес при пациенти с онкологични заболявания. Наборът от данни е разпределен за обучение и тестване на ефективността на избраните техниките, извършен е анализ на признаците, изследвани в Matlab. Оценката е извършена въз основа на показателите Точност (Accuracy), Прецизност (Precision), Пълнота (Recall) и Ф-мярка (F-measure).
Ключови думи: дистрес, признаци, Matlab, точност, прецизност, пълнота, Ф-мярка

MACHINE LEARNING FOR RESEARCH OF DISTRESS IN PATIENTS WITH ONCOLOGICAL DISEASES

Ginka K. Marinova

Abstract: Early diagnosis of distress is important in the initial stage of treatment and is associated with improving the quality of life of patients. Machine learning techniques are suitable for analyzing medical data and researching the task. In the present article, a comparative analysis of the obtained results of four methods for the classification of a set of medical clinical psychological diagnostic data for the recognition of distress in patients with oncological diseases is made. The data set is distributed for training and testing the effectiveness of the selected techniques, an analysis of the features studied in Matlab was per-formed. The estimation was performed based on the indicators Accuracy, Precision, Recall and F-measure.
Keywords: distress, features, Matlab, accuracy, precision, recall, F-measure

Увод

Един от факторите за напредъка на медицината през последните години е използването на методи и алгоритми на машинното обучение за изследване на различни задачи от онкологията и клиничната психологична диагностика. Приложението на машинното обучение е свързано преди всичко с анализ на медицинските данни. То се явява мощен инструмент за разработване на различни модели и тяхното прилагане върху набори от данни на пациенти с онкологични заболявания с цел повишаване достоверността на диагностичната оценка. Реакцията при диагнозата карцином зависи от личността на пациента, психологическата структура, семейството, социална среда и др. В основата си всеки пациент преживява дистрес, свързан с поставянето на диагнозата и провежданото лечение [1, 5].

Дистресът е неприятно преживяване от емоционално, психологическо и духовно естество, което пречи на способността за справяне със заболяването [6]. Тревожността, дистресът и депресията са срещани психични състояния, които допълнително усложняват антитуморното лечение и грижите за болните [2]. Ранният скрининг на дистрес и навременното справяне с него подобряват медицинското лечение [3, 4]. През последните години в научните и изследователски среди се изследват психо-социалните фактори и влиянието на дистреса върху пациенти с онкологично заболяване, като те могат да варират в зависимост от локализацията, стадий на болестта, възраст и пол на пациента [1].

Диагностиката на психичното здраве е многоетапен процес на изследвания и последователност от различни психологически тестове с цел намирането на подходящи скринингови инструменти и съответно начините за тяхното прилагане, осигуряващи необходимите интервенции според специфичните нужди на пациента.

Скрининг инструмент за самооценка на дистрес („Screening Tools for Measurement Distress“) [1, 6] е систематичен подход за преценка на психичното здраве на пациентите, свързан с техните индивидуални психосоциални потребности. За целта се използва валидиран въпросник за оценка на нивото и причините за дистрес. Инструментът е разработен от National Comprehensive Cancer Network (NCCN) [6], този метод е за бърз скрининг за измерване на психологичните симптоми при пациентите.

В това изследване се оценяват четири метода и четиридесет и три признака за задачата за откриване на дистрес за пациенти с онкологични заболявания (раздел Експериментални резултати). За решението на поставената задача са избрани определени класификатори от машинното обучение, като ключов аспект е направеното сравнение по отношение на точността на класификацията, прецизност, пълнота и Ф-мярка (раздел Експериментални резултати).

Методи

Методът на опорните вектори [9] (Support Vector Machines-SVM) е въведен от Vapnik през 1963 г. Той се базира на модел, при който данните се разделят на области в n-мерно пространство по начин, позволяващ в последствие да бъдат класифицирани с помощта на оптимален, линеен класификатор. При линейно разделими области се намира максимално възможната междина между тях и се построява разделяща равнина.

Дърво на решенията [10] (Decision Tree-DT) е метод, който широко се използва при вземането на решения и анализ на данни. Основава се на разработването на структура /дърво/, представена с корен, клони и листа. В дървото на решения всеки вътрешен възел представлява тест, всеки клон – съответно резултата от данните на теста и всеки листен възел представлява етикет на клас.

При метод Багинг (Bagging) моделите се изграждат паралелно, като всеки отделен модел е различен от другите, извършва се комбиниране на резултатите от прогнозирането на различните класификатори, които са обучени на произволни подмножества, следвайки някакъв детерминиран процес на усредняване [11].

При метод Бустинг (Boosting), моделите се създават последователно по адаптивен начин, като базовият модел зависи от предишните и ги комбинира, следвайки определена стратегия. Целта е да се получи силен обучаем модел [11].

Метрики за класификация

Изборът на метрика е важен процес за провеждане на дадено изследване. Матрица на грешките (Confusion matrix) се използва за оценка на модел за класификация, състои се от брой редове и колони, съответстващи на класовете [7]. Редовете на матрицата отговарят на действително наблюдаваните попадения в съответен клас, колоните – на класовете, получени при прилагането на класификационния модел.

Таблица 1. Матрица на грешките [7]

		Предсказан клас	
		Положителен	Отрицателен
Действителен клас	Положителен	Действително положителен True Positive (TP)	Действително отрицателен False Negative(FN)
	Отрицателен	Лъжливо положителен False Positive(FP)	Лъжливо отрицателен True Negative(TN)

На базата на матрицата на грешките са известни множество показатели за ефективност на класификацията [7], [8]. В проучването фокусът е поставен върху четири показателя, обобщени в Таблица 2 [7], [8].

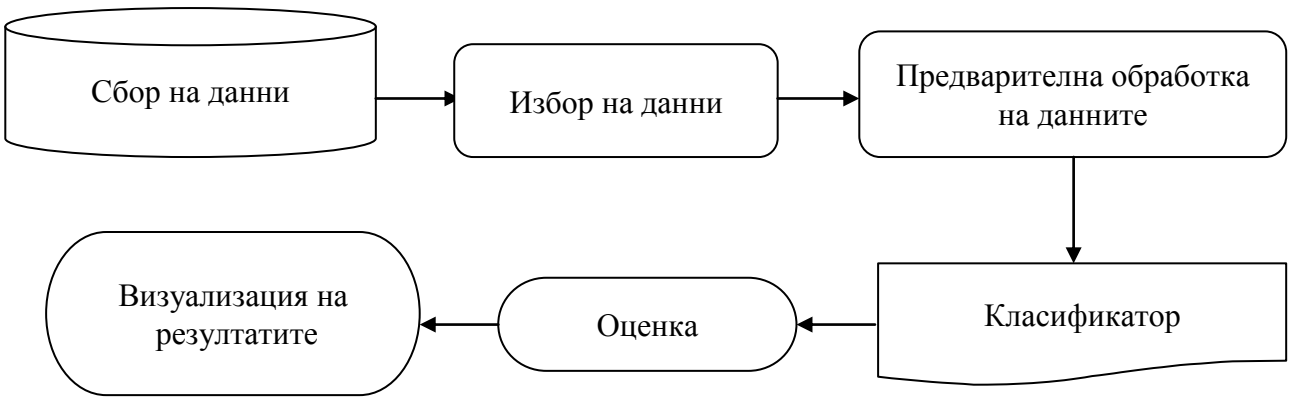
Таблица 2. Показатели за оценка на класификацията

Метрика на български	Метрика на английски	Формула
Точност	Accuracy	$\frac{TP+TN}{TP+FN+TN+FP}$
Прецизност	Precition	$\frac{TP}{TP+FP}$
Пълнота	Recall	$\frac{TP}{TP+FN}$
Ф-мярка	F-measure	$\frac{TP}{TP+\frac{1}{2}(FP+FN)}$

Експериментален протокол

Данните обикновено се съхраняват в база от данни и се търсят начини за намиране на методи за техния анализ с цел генериране на знания. Настоящите изследвания са проведени върху данните от диагностицирането и оценката на дистресово състояние на 6000 пациенти. Събраните данни се обобщават и анонимизират с цел запазване на личните данни на пациентите. Данните, получени в резултат на събирането, трябва да отговарят на определени критерии за качество. Извършена е предварителна обработка на данните като почистване [12]. Почистването на данните е процес, при който се отстраняват фактори, за да се подобри тяхното качество.

Генералната съвкупност е съвкупност от всички психически възможни обекти от даден тип, над които се извършват наблюдения, за да се получат конкретни стойности на определена величина. Блокова схема на процеса класификация за решение на поставената задача е показана на фигура 1.



Фиг.1. Блокова схема на класификация

Данните са представени в табличен вид, редовете отговарят за проведен клиничен преглед на отделен пациент, като е въведено характерно описание, набор от характеристики за всеки пациент. В колоните на таблицата са записани признаците, получените резултати от проведения тест. Скрининг инструмент за самооценка на дистрес се състои от 43 признака, описани в таблица 1 [6].

Таблица 3. Скрининг инструмент за самооценка на дистрес

№	Код	Практически проблеми
1	П1	Грижа за децата
2	П2	Домакинство
3	П3	Финансови
4	П4	Транспортни
5	П5	Работа/училище
6	П6	Решение за лечение
		Семейни проблеми
7	С1	Проблеми в партньорството
8	С2	Възможност за потомство
9	С3	Интимност с партньора
10	С4	Общуване с децата
		Емоционални проблеми
11	Е1	Депресивност
12	Е2	Страх
13	Е3	Нервност
14	Е4	Тъга
15	Е5	Тревожност
16	Е6	Гняв
17	Е7	Загуба на интерес към обичайните
18	Е8	Нежелание за планиране
		Духовни/религиозни проблеми
19	Д1	Загуба на смисъл и цел в живота
20	Д2	Загуба на вяра

№	Код	Физически проблеми
21	Ф1	Външен вид
22	Ф2	Къпане/обличане
23	Ф3	Дишане
24	Ф4	Промяна в уринирането
25	Ф5	Запек/диария
26	Ф6	Хранене
27	Ф7	Умора/изтощение
28	Ф8	Усещане за подпухналост
29	Ф9	Температура
30	Ф10	Лошо храносмилане
31	Ф11	Памет/концентрация
32	Ф12	Рани в устата
33	Ф13	Проблеми със зъбите
34	Ф14	Гадене/повръщане
35	Ф15	Сухота в носа/претовареност
36	Ф16	Болка
37	Ф17	Проблеми със
38	Ф18	Суха кожа/сърбеж
39	Ф19	Сън/Безсъние
40	Ф20	Злоупотреба с лекарства
41	Ф21	Изтръпване на ръце/крака
42	Ф22	Кървене
43	Ф23	Високо кръвно налягане

Следвайки методиката за самооценка на дистрес [1, 6], обект на изследване са данните, съдържащи години на пациента, пол, списък с признаци, разпределени в пет категории: практически проблеми; семейни проблеми; емоционални проблеми; духовни/религиозни проблеми; физически проблеми, свързани с болестта. При избора на признак се присвоява двоично тегло, където 1 означава, че е избран и 0 - не е избран.

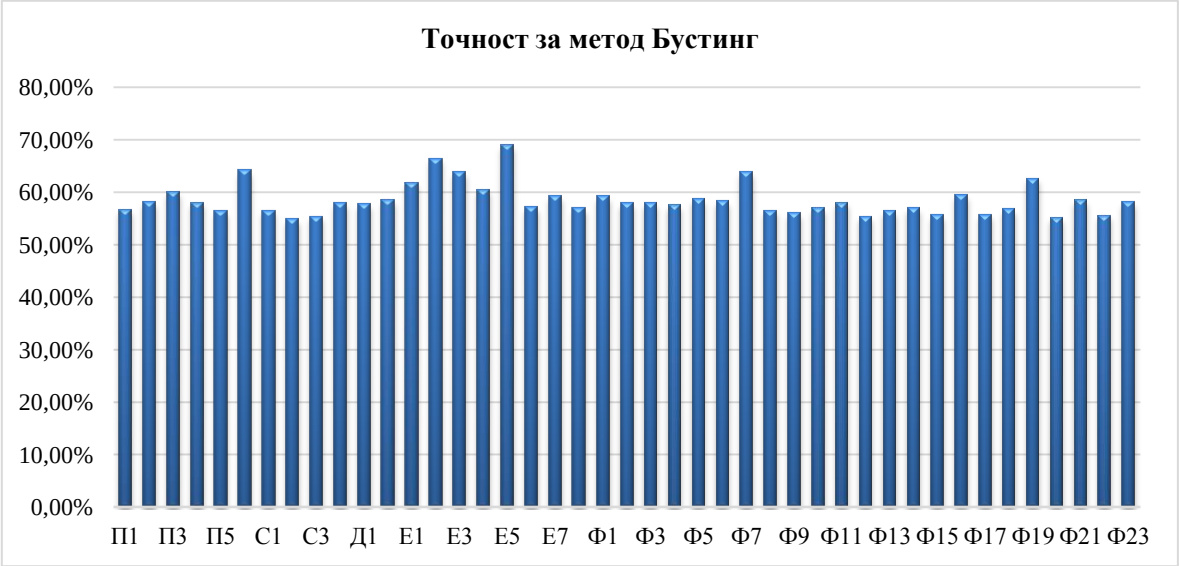
Експериментални резултати

От анализа на данните може да се направи заключението, че разпространените причини за психологически дистрес сред пациентите са: за практически проблеми - домакинство, финансови, решение за лечение; за емоционални проблеми - депресивност, страх, нервност, тревожност; за физически проблеми - загуба на интерес към обичайните дейности, болка, сън/безсъние, изтръпване на ръце/крака, високо кръвно налягане.

Фиг. 2. Разпределение на данните по признаци

В този раздел са представени проведените експерименти и е извършен анализ на ефективността на четири алгоритъма за класификация на медицински клинични психологични диагностични данни, като общият набор от данни е разпределен както следва: 80% са в обучителното множество, а 20% - в тестовото, броят на записите е 9240 за 6000 пациента с туморно заболяване. За всички реализирани експерименти се използва k-кратно кръстосано валидиране, като k=10 и всички експерименти са направени в MATLAB (R2019a). Матрицата за грешки е представена в таблица, обектите са разпределени в четири категории със свързаност от комбинацията от верния отговор и отговора на алгоритъма. Класификаторите са изпълнени посредством включване на всички признаци (43), идентифицирани от обработената таблица. Сред известните мерки четири са най-важни за сравняване на класификаторите. Те са Точност (Accuracy), Прецизност (Precision), Пълнота (Recall) и Ф-мярка (F-measure).

На фигура 3 са показани получените резултати за всички признаци по метод Бустинг. Най-висока точност се получава за Е5-Тревожност 68.44%.

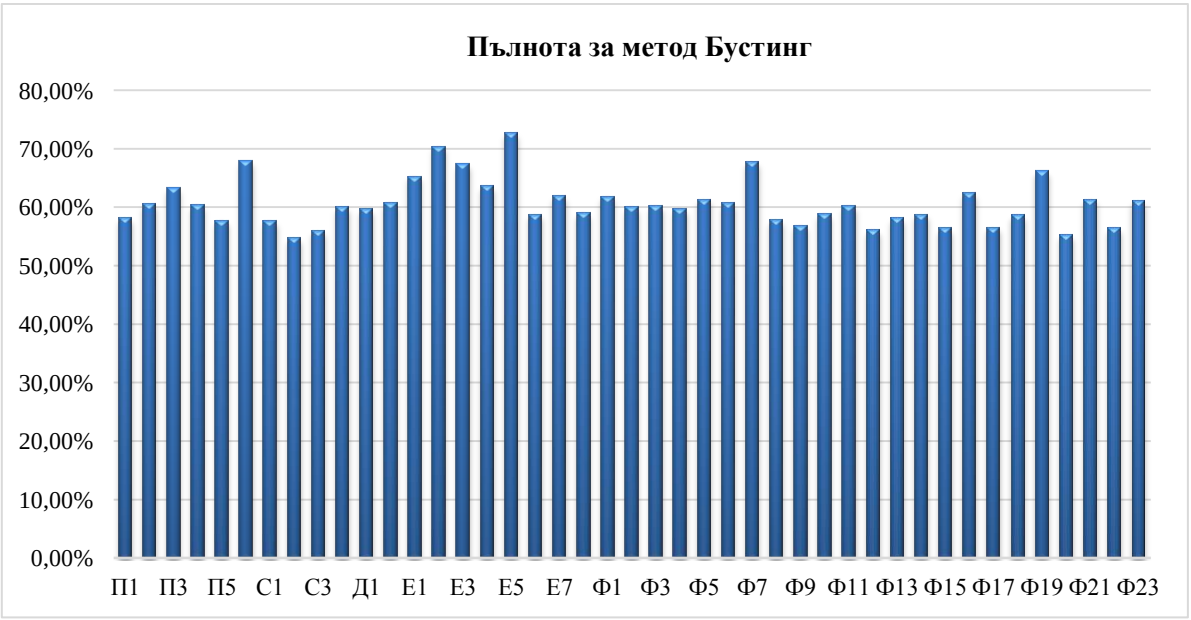


Фиг. 3. Сравнителен анализ по отношение на точност

Прецизността е степента на близост един до друг на независимите резултати от измерванията, които са получени при конкретни условия. Получената прецизност по метода на опорните вектори е визуализирана на фигура 4 и е 72.72% за показател Е3-Нервност.

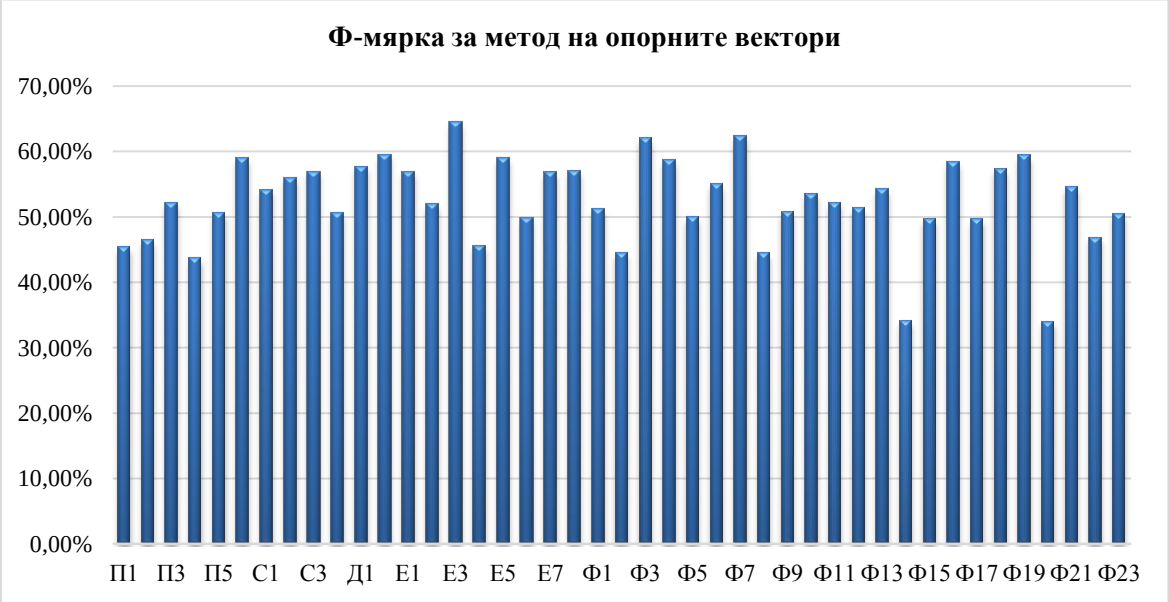
Фиг. 4. Получени резултати по отношение на прецизност

От получените експериментални данни и построената графика, показана на фигура 5 се вижда, че най-висок процент за пълнота се получава за Е5- Тревожност 72.57% за метод Бустинг.



Фиг. 5. Сравнителен анализ по отношение на пълнота

Резултатите от експеримента са визуализирани на фигура 6. Най-висок процент за Ф-мярка се получава за Е3 - Депресивност 64.47%.



Фиг. 6. Получени резултати по отношение на Ф-мярка

Заключение

Ранното диагностициране и ефективният контрол на нивата на дистрес подобряват качеството на живот на пациентите с онкологични заболявания. Технологията за извличане на данни осигурява ориентиран към анализатора подход към нови и скрити модели в данните. В статията са анализирани получените резултати за четири метода за машинно обучение, като се изследвани четиридесет и три признака на двоични данни. От направеното сравнение на експерименталните резултати базирани на показателите за оценка на точност показват, че за Е5-Тревожносте е 68.44%. За прецизност по метод на опорните вектори се получава 72.72% за Е3-Нервност. За пълнота е признак Е5- Тревожност 72.57% за метод Бустинг. Най-висок процент за Ф-мярка се получава за Е3-Депресивност 64.47% за метод на опорните вектори. Предмет на допълнителни изследвания за подобряване на ефективността на тези класификационни техники са синтезирани характеристични описатели за откриване на дистрес [13]. Тези знания могат да се използват за по-добро вземане на клинични решения от лекарите и психоонколога към дадено лечебно заведение.

Благодарности

Авторът би искал да признае за подкрепата на проект NP5 „Изследване на възможностите за интегриране на машинно обучение и Blockchain технологии за Internet of Things“.

Литература

[1] Власаков, В., и колектив, Психосоциална подкрепа и рехабилитация в онкологията, национален експертен борд клинично ръководство, основано на доказателства Атр Трейсър 2015, стр. 228

[2] Mehnert, A. et al. “One in Two Cancer Patients Is Significantly Distressed: Prevalence and Indicators of Distress.” Psycho-Oncology 27.1, 2017, pp.75–82. (DOI 10.1002/pon.4464)

[3] Linden W, et al. Anxiety and depression after cancer diagnosis: Prevalence rates by cancer type, gender, and age. J Affect Disord 2012; 141, pp. 343-351

[4] Sumathi M.R, Poorna B., Prediction of Mental Health Problems Among Children Using Machine Learning Techniques (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 7, No. 1, 2016, pp. 552-557
[5] Civilotti C., et al. The use of the Distress Thermometer and the Hospital Anxiety and Depression Scale for screening of anxiety and depression in Italian women newly diagnosed with breast cancer *Supportive Care in Cancer* volume 28, 2020, pp 4997–5004
[6] Riba, Michelle B**.** et al., Distress Management, Version 3.2019, NCCN Clinical Practice Guidelines in Oncology, 2019, pp. Vol. 17, no. 10. https://doi.org/10.6004/jnccn.2019.0048
[7] Luquea A., et al. The impact of class imbalance in classification performance metrics based on the binary confusion matrix, Pattern Recognition, Volume 91, July 2019, pp. 216-231 https://doi.org/10.1016/j.patcog.2019.02.023
[8] M. Hossin, M.N. Sulaiman, A review one valuation metrics for data classification evaluations, International Journal of Data Mining & Knowledge Management Process, 5(2), 2015, pp. 1-11, DOI: 10.5121/ijdkp.2015.5201
[9] Золотых, Н. Машинное обучение и анализ данных, 2013, http://www.uic.unn.ru/˜zny/ml.
[10] Kaur R., Ginige J., Comparative Evaluation of Accuracy of Selected Machine Learning Classification Techniques for Diagnosis of Cancer: A Data Mining Approach, International Journal of Biomedical and Biological Engineering Vol:12, No:2, 2018 pp.19-25
[11] Rising O. Odegua, Ensemble methods: bagging, boosting and stacking, Conference: Deep Learning IndabaX, 2019 pp.1-10
[12] Hellerstein J., Quantitative Data Cleaning for Large Databases, EECS Computer Science Division, 2008, pp.1-42
[13] Marinova, G., Ganchev, T., & Nikolov, N., (2020) Synthesis of characteristic descriptors for the detection of distress, Proc. of the International Conference on Biomedical Innovations and Applications, BIA-2020, Sept. 24-27, 2020, Varna, Bulgaria. Paper #23.

За контакти: ас. Гинка К. Маринова
катедра „Компютърни науки и технологии“
Технически университет –Варна
e-mail: gmarinova@tu-varna.bg

ОБЗОР НА ТЕХНИКИ ЗА ИЗВЛИЧАНЕ НА ДАННИ В ОНКОЛОГИЯТА И ПСИХООНКОЛОГИЯТА

Гинка К. Маринова, Мая П. Тодорова

Резюме: Извличането на данни е един от съвременните аналитични методи, използващ се в медицината – онкологията като интердисциплинарна област на изследвания. Този метод е надежден за откриване на модели и взаимоотношения в големи масиви от данни. Задачата за извличане на знания от медицинските клинични данни е предизвикателство и е сложна задача. Анализирани са различни етапи за извличане на данни в онкологията. В статията е направен обзор на публикации в областта, като е акцентирано върху извличането на данни и тяхното приложение в медицинската онкология и психоонкология.
Ключови думи: онкология, психоонкология, извличане на данни, медицински клинични данни

A Review on Data Mining Techniques in Oncology and Psycho-Oncology

Ginka K. Marinova, Maya P. Todorova

Abstract: Data mining is one of the modern analytical methods used in medicine - oncology as an interdisciplinary field of research. This method is reliable for detecting patterns and relationships in large data sets. The task of extracting knowledge from medical clinical data is a challenge and a complex task. Different stages for data mining in oncology are analyzed. The paper is done a review of publications in the field, as is emphasized on data mining and their application in medical oncology and psycho-oncology.
Keywords: oncology, psycho-oncology, data mining, medical clinical data

Увод

Развитието и внедряването на Информационните технологии (ИТ) в онкологията и клиничната психологична диагностика предоставят възможността за създаване на електронни бази от данни. Необработените медицински данни обикновено са значителни по обем и различни по своята същност.

Основни източници на информация за онкологично болни пациенти са:

- медицински електронни досиета;
- лабораторни изследвания и тестове;
- приложено медикаментозно лечение;
- снета анамнеза за пациент;
- направени ехографии, томографии, лъчетерапии, таргентерапи и др.;
- наблюдения и оценки от онколози, психоонколози;
- данни от проведени медицински процедури и манипулации;

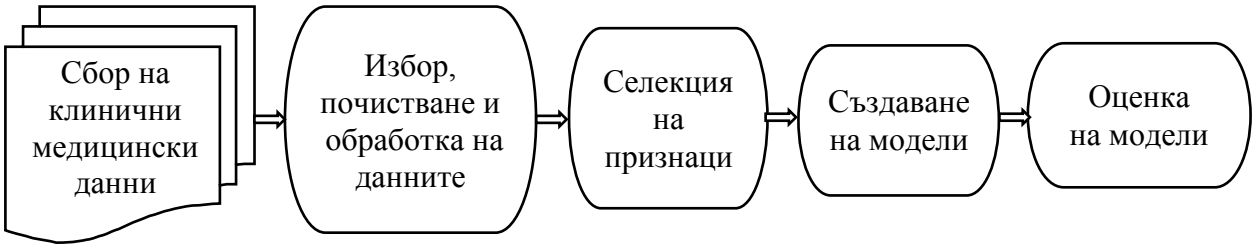
Използването на новите ИТ предоставят възможност за бърза и ефективна обработка на предоставените клинични данни и формулиране на нови цели за научно-изследователска работа. Анализът на данните и методите на машинното обучение подпомагат научното развитие в онкологията и психоонкологията.

Приложените методи за обработка на постъпилите данни за онкологично болни пациенти целят да подпомогнат медицинските специалисти при поставянето на диагноза, избор на лечение и проследяване на развитие на заболяването. Съвременните аналитични методи за извличане на данни се базират на надеждни модели, основаващи се на обработката на големи масиви от данни. Сложността на моделите, тяхната адекватност, надеждност и

високата степен на достоверност на получените резултати, позволява тяхното използване за прогнозиране развитието на заболяването.

Етапи за извличане на данни

Извличането на данни е задача, свързана със селекция, проучване и моделиране на големи количества данни. На фигура 1 е представен процесът за извличане на данни в онкологията и психоонкологията.



Фиг. 1. Процес за извличане на данни

Основна цел е откриване на взаимовръзки между данните и създаване на модели, чрез които да се осигури точен резултат и възможност за прогноза от специалистите в областта. Етапите за извличане на данни са [2], [3]:

- *Събиране и подготовка на данните*

Събирането и организацията на информацията, анализът и интерпретацията на данните е важен процес и има първостепенно значение, за да може да стане възможно извличането на знания. Данните, получени в резултат на събирането, е необходимо да отговарят на определени критерии за качество. Качеството на данните е мярка, която определя точността и способността за тяхната интерпретация. Правилният подбор на качествени данни е от важно значение за получаване на добри резултати.

- *Избор и почистване на данните*

Чрез процедурите за почистване на данните се отстраняват фактори, пречещи на работата на класификаторите. Почистването включва коригиране на неточности в данните, обработка на дубликати и почистване на данните от шум. Прилагат се следните методи за обработка на липсващи данни:

- изключване на обекти от обработка;
- изчисляване на нови стойности;
- игнориране на пропуски;
- замяна на пропуските с възможни стойности.

Възможно е наличието на шум в данните, причинен е от дисперсията на наблюдаваните стойности. Използван метод за контрол на шум е спектралният анализ.

- *Техники за извличане на данни*

Четирите класа задачи, известни от литературата [1], [2], [3], за извличане на данни в областта на онкологията, са:

- клъстеризация,
- класификация,
- регресия,
- асоциация.

Клъстеризацията е групиране на обекти на базата на свойства, описващи същността на тези обекти. Групирането на данни е разделяне на подбор от обекти на еднородни групи - клъстери или класове. Класовете данни не са предварително определени. В машинното обучение задачата за клъстеризация се разглежда като задача за обучение без учител.

Класификацията е процес на извличане на данни, позволяваща организиране на обекти в определени категории и изисква наличието на признаци, характеризиращи групата, към която принадлежи даден обект. Разделят се наборите от данни на целеви класове. Класовете са предварително известни. Според броя на изходите класификацията е бинарна и многомерна. Създаването и обучението на класификатор изисква предварително разделяне на входните данни на обучаваща и тестова извадка.

Регресията е предсказване на числова стойност на изходна величина на базата на даден набор от входни величини. Тя е статистически метод, използван за прогнозиране на числови стойности. Задачата за регресия е подобна на задачата за класификация. Основна разлика между двата типа задачи е, че при регресионната задача предсказваната променлива е числова величина, която приема непрекъснати стойности.

Асоциацията е една от известните техники за извличане на данни. При този тип задача се търси обединение или свързване на обекти в големи бази от данни. Връзката е известна като правило за асоцииране. Асоциацията разкрива връзки, съществуващи между обектите. Основната цел е откриване на корелационни зависимости, съществуващи между обектите.

На фигура 2 са визуализирани техниките за извличане на данни, използвани в онкологията и психоонкологията.

Фиг. 2. Техники за извличане на данни

- *Изграждане на модели*

Изграждането на класификационни и регресионни модели е ефективен метод за извличане на данни. В процеса на моделиране се идентифицират повтарящи се взаимовръзки

между входни променливи, които описват обектите, принадлежащи на един и същи клас. С помощта на създадените модели се открива интерпретируема информация, използвана за вземане на решения.

- *Тестване и оценка на знанията*

Тестване и оценка на знанията осигурява по-добро разбиране на набора от данни. Наборът от входните данни се разделя на обучаващо и тестово множество. Оценката на конструирания модел се реализира на база приложението му върху тестовото множество от данни, включващо проверка на неговата коректност.

Обзор на научни публикации в областта

Извличането на знания от големи масиви от данни подпомага както лекари, специалисти и учени-изследователи в областта на медицинската онкология, така и пациентите да направят правилен и информиран избор. Представен е кратък обзор на публикации, свързани с извличане на медицински данни за пациенти с онкологични заболявания, като са използвани методи от машинното обучение, приложени върху тези данни за решаване на различни класификационни задачи, с цел подпомагане на медицинската онкология и психоонкология.

- *Рейтингова система на лечебни заведения*

В [4] е създадена рейтингова система на лечебни заведения. В проучването са включени и анализирани данни за регистрирани пациенти с онкологични заболявания на гърдата, кожата, бронхите и белите дробове, за период от 2010 до 2014 година. Предмет на изследване са здравни заведения, които са приложили лечение на над сто пациента за петгодишен период. За формиране на цялостния рейтинг на дадено лечебно заведение е отчетена преживяемостта на всеки пациент и спецификата на заболяването. Създадени и обучени са класификационни модели Дърво на решения, К-най-близък съсед, Наивен Бейсов класификатор, Метод на опорните вектори и Ансамблови алгоритми. Използваните метрики за оценка на създадените класификатори са точност, прецизност, пълнота и класификационна грешка. Решаването на класификационната задача цели да подпомогне пациентите, за да могат да направят информиран избор относно лечебното заведение, в което да им бъде приложено лечение.

- *Рак на гърдата*

В [5] е направен сравнителен анализ на четиринадесет различни алгоритъма за класификация, определящи вида на тумора (доброкачествен или злокачествен), както и да се предскаже вероятността от рецидив. Анализът е оценен посредством два набора диагностични данни от пациенти с тумори на гърдата. Ефективността на класификаторите зависи от набора от данни и съществено от броя на признаците. Нито един от изследваните класификатори няма доминиращо значение спрямо останалите. В резултат на проведените експерименти авторите не успяват да идентифицират алгоритъм, който да работи еднакво добре и за двата набора от данни.

При изследването [6] на алгоритмите за прогнозиране състоянието на пациенти с рак на гърдата е използвана базата от данни на SEER (Surveillance Epidemiology and End Results) [11]. Изградени са три модела за прогнозиране на преживяемостта при рак на гърдата: на базата на Невронни мрежи, Дърво на решения и Наивен Бейсов класификатор. Критерий за оцеляване е пациентът да е преминал в ремисия. Направена е сравнителна оценка на точността на създадените модели. Най-добра точност от 93,6% е постигната чрез модел,

създаден с алгоритъма C4.5 по метода Дърво на решения, в сравнение с модели, създадени с методите Наивен Бейсов класификатор и Невронни мрежи [6].

В [7] е извършен анализ на факторите за преживяемост на рак на гърдата с помощта на технологията за машинно обучение. Данните са предоставени от университетския медицински център в Малая, Малайзия, като броят на изследваните пациенти е 8066. Наборът от данни съдържа двадесет и три предикаторни променливи и една категориална променлива, определена на база преживяемост на пациента. Основни предикаторни променливи са: стадия на рак, размер на тумора, броят на всички аксиларни отстранени лимфни възли, видът на първично лечение и методи за диагностика. Моделите, създадени с метода Дърво на решения, са с най-малка точност - 79,8%. Най-голяма е точността при използване на алгоритъм Случайна гора. В статията е идентифицирана степента на преживяемост на рак на гърдата [7].

- *Рак на белия дроб*

В [8] са разгледани многобройните подходи за откриване на рак на белите дробове. Цифровата обработка на изображения и извличането на данни са важни, тъй като за прогнозиране се използва или набор от изображения, или статистически набор от данни. След предварителна обработка, сегментиране и извличане на функции, са приложени различни алгоритми за прогнозиране на рак на белия дроб. В конкретното изследване наборът от данни се състои от 1018 случая на възли с диаметър по-голям от 3 мм. Изследването е проведено с помощта на два типа набори от данни - първият е набор от изображения, вторият се базира на статистически набор от данни. За етапа на класификация са използвани осем алгоритъма. По отношение на точността най-добре се представят моделите създадени с Невронни мрежи и Метод на опорните вектори.

- *Психоонкология*

В [9] се прилага йерархичен клъстерен анализ за идентифициране на подгрупи на пациенти с онкологично заболяване. Задачата е да се определят подгрупи на онкологичните амбулаторни пациенти, идентифицирани въз основа на оценките им за тежестта на умора, нарушение на съня, депресия и болка. Определянето на пациентите в подгрупи е направено по избрани демографски, болестни и лечебни характеристики. Техниките на клъстерния анализ са полезни за изследване на механизмите, оказващи влияние върху симптомите от преживяванията. Използването на приложената статистическа методология подпомага за идентифициране на ниско, умерено и високорискови групи на пациенти, като са приложени различни дози или по-целенасочени интервенции за управление на симптомите. Пациентите с високи нива на умора, нарушение на съня, депресия и болка се нуждаят от интервенции за подобряване на тяхното състояние.

В [10] е изследвана връзката на коморбидна тревожност и депресия с различни фактори (социално-демографски фактори, социално-икономически статус, поведение в живота, здравословни условия, умора, социална подкрепа, тревожност), използващи логистична регресия и съответно три алгоритъма от машинно обучение. Направено е сравнение на различни модели за предсказване на коморбидна тревожност и депресия, основаващи се на данни на 1546 китайски пациенти с онкологично заболяване. Сравнителният анализ включва получената точност, чувствителност, прецизност и Receiver Operating Characteristic (ROC) кривите за обучителното и тестово множество. Най-добри резултати са получени за Случайна гора. Авторите препоръчват психологическата помощ да бъде включена в редовните здравни грижи при пациентите. Освен това сътрудничеството между онколог, психоонколог, и семейство са необходими за осигуряване на подкрепящи съвместни грижи [10].

Заключение

Методите за извличане на данни предоставят нови възможности на лекуващите лекари, онколози, психоонколози, научни и практикуващи анализатори, както и за самите пациенти. В резултат на това се повишава и рейтингът на лечебните заведения.

Използвайки знанията като резултат от извличането на данни, лекарите- специалисти имат възможност да повишат качеството на диагностициране и съответно провеждането на ефективно лечение. Аналитичните и описателни способи за извличането на данни предоставя широки възможности за научни изследвания и практическо приложение в областта на онкологията. В направения обзор на публикации за извличане на данни са анализирани техниките и тенденциите, използвани в областта на онкологията и психоонкологията.

Литература

[1] Chen MS., et al., "Data mining: an overview from a database perspective," Knowledge and Data Engineering, IEEE Transactions on, vol. 8, 1996, pp. 866-883,

[2] Pandey S. "Data Mining Techniques for Medical Data: A Review" International conference on Signal Processing, Communication, Power and Embedded System-2016 DOI: 10.1109/SCOPES.2016.7955586

[3] U. Fayyad, et al., "Knowledge discovery and data mining: towards a unifying framework," presented at the Proc. Second Intl Conf. Knowledge Discovery and Data Mining, Aug 1996.

[4] Marinova G., Todorova M., „Classification Tasks Solving with Machine Learning Methods“, 2020 XXIX International Scientific Conference Electronics DOI: 10.1109/ET50336.2020.9238218

[5] Nookala G., et al., „Performance Analysis and Evaluation of Different Data Mining Algorithms used for Cancer Classification“, International Journal of Advanced Research in Artificial Intelligence, Vol. 2, No.5, 2013 pp.49-55

[6] Bellaachia A., Guven E., „Predicting Breast Cancer Survivability Using Data Mining Techniques“, Department of Computer Science The George Washington University Washington DC 20052 pp.1-4

[7] Ganggayah M. et al., „Predicting factors for survival of breast cancer patients using machine learning techniques“, BMC Medical Informatics and Decision Making (2019) pp.1-17, https://doi.org/10.1186/s12911-019-0801-4

[8] Banerjee N., Das S., „Machine Learning Techniquesfor Prediction of Lung Cancer“, International Journal of Recent Technology and Engineering (IJRTE)ISSN: 2277-3878, Volume-8 Issue-6, March 2020 pp.241-249

[9] Pud D., et al. „The Symptom Experience of Oncology Outpatients Has a Different Impact on Quality-of-Life Outcomes“, Journal of Pain and Symptom Management Volume 35, Issue 2, February 2008, pp. 162-170

[10] Yan R., et al. „Prediction of comorbid anxiety and depressionusing machine learning models in cancer survivors“, Version 1, Posted 11 Jun, 2020, DOI: 10.21203/rs.3.rs-32449/v1

[11] Hankey BF, Ries LA, Edwards BK. The surveillance, epidemiology, andend results program: A national resource. Cancer Epidemiol BiomarkersPrev 1999; pp.1117–1121

За контакти:
ас. Гинка К. Маринова
катедра „Компютърни науки и технологии“
Технически университет –Варна
e-mail: gmarinova@tu-varna.bg

ПРИНОСЪТ НА СЪВРЕМЕННИТЕ ИНФОРМАЦИОННИ ТЕХНОЛОГИИ В СОЦИАЛНИЯ ЖИВОТ НА ХОРАТА

Галина Т. Найденова

Резюме: В съвременния свят технологиите все по-често навлизат в ежедневието на хората. В условията на пандемия Covid 19 технологиите са най-използваното средство за свързване между неправителствени и правителствени организации, училища или пък се използват за лична комуникация между близки и познати. Какво влияние оказват те върху психо-емоцианалното поведение на децата? Дали технологиите са от полза за развитието на децата или оказват отрицателно влияние и отклонение в поведението им? Настоящата статия дискутира точно тези две различни гледни точки. Пред какви проблеми са изправени в условията на пандемия Covid 19 родители и психолози за работа с деца с потребности? Ще бъде направен обзор на съществуващи мобилни приложения, предназначени за деца с кортикални зрителни увреждания и дискутирано дали те са приложими за българските деца с този проблем. На базата на съвети за дизайн за приложения за деца с кортикални зрителни увреждания ще бъде представено проектиране и подходи за разработка на мобилни приложения.
Ключови думи: мобилни приложения, интерфейси, UX, кортикални зрителни увреждания (CVI), увреждания, деца със специални потребности

The Contribution of Modern Information Technologies to the Social Life of People

Galina T. Naydenova

Abstract: In today's world, technology is increasingly entering people's daily lives. In the context of the Covid 19 pandemic, technology is the most widely used tool of non-governmental and government organizations, schools, or used for personal communication between relatives and acquaintances. What influence do they have on children's psycho-emotional behavior? Are technologies beneficial to children's development or do they have a negative impact and deviation in their behavior? This article discusses exactly these two different points of view. What problems do parents and psychologists face in the context of the Covid 19 pandemic in working with children with needs? An overview of existing mobile applications intended for children with cortical visual aids will be made and it will be discussed whether they are applicable for Bulgarian children with this problem. Based on design advice for applications for children with cortical visual impairments, design and approaches for mobile application development will be presented.
Keywords: mobile applications, interfaces, UX, cortical visual impairment (CVI), disabilities, children with special needs

1. Увод

Невронауката поставя началото на революционна промяна в начина, по който мислим за детското развитие и има голямо значение за бъдещето на деца със специфични образователни потребности (СОП) и тяхната общност. А това позволява да се разбере по-добре взаимодействието между средата и генетичните фактори, както и позитивните и негативните въздействия върху развиващия се мозък.

Днес е всеизвестно, че мозъкът се влияе еднакво както от генетичните фактори, така и от факторите на средата. Схемата на развитие на мозъка се осигурява от гените, а средата оформя мозъка. Направени са много научни изследвания относно това, че децата могат да променят експресията на гените в мозъка благодарение на грижите към тях. Невралните

мрежи и пътища се осъществяват до шест годишна възраст. А точно този период дава най-високата възвръщаемост от инвестиция от обучението на детето.

Според Уницеф 200 милиона деца до петгодишна възраст не успяват да реализират своя потенциал. Ключов етап от развитието на мозъка е инвестирането в ранна интервенция, която може да подобри живота на децата.

Добре ли е да се използват мобилни информационни технологии в образованието на децата? Както много други въпроси, така и този има поддръжници с две противоположни гледни точки.

Статията е структурирана по следния начин: секция 2 описва влиянието на технологиите върху емоционалното здраве на децата, секция 3 дава определение за кортикални зрителни увреждания (CVI) и описва спецификате за работа с деца за всяка една от фазите, секция 4 анализира съществуващи мобилни приложения, предназначени за деца със CVI, в секция 5 се предлагат съвети за дизайн на приложения относно фазите на зрителните увреждания и необходимостта от създаване на мобилни образователни приложения на български език, в секция 6 се предлагат подходи за разработка на мобилни приложения и секция 7 обобщава изложеното в статията

2. Влиянието на технологиите върху емоционалното здраве на децата

Съществуват две различни гледни точки относно времето на използване на електронните мобилни технологии от децата.

2.1. Хората с първата гледна точка са на мнение, че използването на **технологиите в ранна детска възраст имат отрицателно въздействие върху емоциите и поведението на децата**. Уповават се основно на идеята, че децата:

- по-трудно обработват информация в училище, поради високата стимулация, която дават видеоигрите след използване;
- лесно губят търпение и концентрация;
- когато гледат в електронните устройства, е установено, че част от мозъка им спира да обработва информация за разлика от четенето, където се изисква активна мозъчна дейност за развитие на въображението.

Съществуват няколко източника, които потвърждават гореизброените причини:

В книгата си „Малките данни“ [2] Мартин Линдстрьом цитира думите на Стив Джобс, публикувани в статията „Steve Jobs was a low-tech parent“ във вестник New York Times през 10 Септември 2014 г:

„Вкъщи ограничаваме достъпа на децата си до технологията. Знам от първа ръка какви са вредите от технологията. Виждал съм го лично и не искам същото да стане и с моите деца.“

Според В. Тончева и В. Банова [9] от най-ранна детска възраст гледането на телевизия води до нарушение в общуването и любопитството към околните и околния свят, а това води до изоставяне в развитието на детето както и до проява на аутистично поведение. Но също така са на мнение, че съвременните технологии са неизменна част от живота на детето и не могат да бъдат елиминирани и детето не трябва да бъде оставяно дълго време безотговорно да борави с тях без надзор.

В друга статия [10] се споделя, че много от децата вече имат мобилни телефони и че пристрастяването им към игрите е, защото там има ясни правила. Както и че отговорността за тази създала се зависимост към мобилните телефони е не само на разработчиците на игри, но и на всички участващи във възпитанието на децата като законодателството, технологиите, образованието, семействата, които трябва да действат заедно.

Севджихан Еюбова [11], която се уповава на [8] казва, че при изследване с ЯМР се установява, че децата в ранно детско развитие, които стоят пред екрана повече от един час на

ден, без контрол от страна на родителя, имат по-ниски нива на развитие на бялото вещество на мозъка в ключова област, отговорна за развитието на езика, грамотността и когнитивните умения. В заключение тя казва, че технологиите трябва да бъдат в интерес на хората и най-вече да им служат, като бъдат използвани внимателно в ежедневието им. Както и че родителите са тези, които трябва да бъдат отговорни към съдържанието и времето, за което децата прекарват с технологиите.

2.2. Втората гледна точка е на хората, които смятат, че **технологиите могат да бъдат друг, подпомагащ обучителен метод за деца със специални образователни потребности.** От малкия си опит с незрящи деца мога да твърдя, че четенето със специален софтуер на компютър е за предпочитане пред четенето чрез Брайл.

Според мненията, представени в точка 2.1, става ясно, че технологиите, предоставени на децата за дълго време без надзор, водят до неадекватно поведение, стигащо до диагноза аутизъм. Не се отрича използването на технологиите, но това трябва да бъде извършено под наставничеството на родителите.

В „Емоционалната интелигетност“ [3] Даниъл Голман споделя, че зрението на човека се развива през първите шест години от живота му. А от това може да се направи извод, че:

- помощта и интервенцията в ранното детско развитие е от голямо значение за деца, които имат проблеми със зрението и слуха;
- технологиите могат да бъдат в подкрепа на деца със специални образователни потребности.

Деца, които са с диагноза кортикално зрително увреждане (CVI) или са с нарушено зрение или нарушен слух, не винаги могат да бъдат под прякото обгрижване на специалист, особено сега в условията на пандемия Covid 19. Те не могат да се възползват отдалечено от досегашните съществуващи методи за работа с тях и са поставени на онлайн терапия. В тази ситуация, когато специалистите не са в пряк контакт с детето, се налага родителите да поемат част от ролята на специалиста, но това не винаги се получава толкова успешно, а това води до загуба на ресурси и време за икономиката като цяло, защото прекъсването на терапевтичната работа води отново до започване на работа с детето от първоначалната стъпка. В този ред на мисли Норбеков казва: „Което се тренира, то се развива!“.

В книгата си „Да натиснем Refresh“ [1] Сатя Надела, сегашният главен изпълнителен директор на Microsoft и баща на дете с увреждане в резултат на вътреутробна асфикция, споделя: „При едно от посещенията ми в интензивното отделение...видях нещата под един съвсем различен ъгъл. Забелязах колко много от устройствата работеха с Windows и как все повече бяха свързани с облачната технология - ...която приемаме като нещо съвсем обикновено. “ Той осъзнава взаимодействието между съпричастност и технология.

В главата „Роботи в клас“ в книгата „Светът утре“ [5] Ричард ван Хойдонк споделя, че недостигът на учители е предизвикателството, което ще бъде разрешено чрез технологиите и че е назряла нуждата да бъдат внедредни дигитални учебни методи и роботи, подкрепени от изкуствен интелект. Той споделя и за децата, младежите и възрастните генерация Z, които са родени между 1992 г. и сега, че вече са свикнали с технологиите на лицево и гласово разпознаване и управляват с жестове най-различни апаратури. В този ред на мисли технологиите на лицево и гласово разпознаване, както и мобилните технологии са жизнено важни за работа, обучение и когнитивно развитие за деца със специфични образователни потребности.

3. Какво е кортикално зрително увреждане (CVI)?

Кортикално зрително увреждане (CVI) [6] е термин, използван за описание на зрителни увреждания, които възникват поради мозъчно увреждане.

За разлика от другите видове зрителни увреждания, които се дължат на физически проблеми с очите, CVI се причинява от увреждане на зрителните центрове на мозъка, което пречи на комуникацията между мозъка и очите. Очите могат да виждат, но мозъкът не интерпретира това, което се вижда.

Кортичното зрително увреждане (CVI) често се споменава с други термини, включително: церебрално зрително увреждане, неврологично зрително увреждане, зрително увреждане на мозъка и така нататък. Всички тези термини се отнасят до зрителна дисфункция в резултат на нараняване на зрителните центрове на мозъка.

Според Dr. Christine Roman в нейната книга "Cortical Visual Impairment – An Approach to Assessment and Intervention" [6] зрителни увреждания се делят на три фази:

- Фаза 1 – основно ниво;
- Фаза 2 – за междинно ниво;
- Фаза 3 – за по-напреднали.

Тези фази са предназначени с цел общи насоки, но нито един бенифициент не може да бъде строго определен точно със съответната категория. Фазите основно са, за да може да се подготвят материали и и да се модифицират уроците така, че да бъдат специално пригодени за съответното дете.

3.1. Стратегии за работа с деца във Фаза 1 [12] – основно ниво

Децата в тази фаза често фиксират и реагират на един източник на светлина, ярко оцветени предмети, бавно движещи се цели и ярки стени. Предметите трябва да бъдат разположени на една ръка разстояние, да контрастират с фона, а броят им трябва да бъде не повече от два, разположени на голямо разстояние един от друг.

Често успяват да забележат предмети, разположени на височината на очите, в крайно ляво или дясно и рядко в долната част на зрителното си поле. Визуална умора често е проблем и много от децата имат затруднение с използването на очите или ръцете.

Препоръки и методи за работа с децата във Фаза 1:

Да се използват предмети, които са познати от всекидневието на детето и с които детето е имало смислен опит.

Спрямо визуалната цел да се използват едноцветни предмети в жълто, червено или флуоресцентни цветове. Да се избягват мигащи светлини.

3.2. Стратегии за работа с деца във Фаза 2 – междинно ниво

В тази фаза децата използват зрението си по-последователно, но често неефективно, без определена цел. Могат да разпознават ниски нива на познат фонов шум. Контрастът между целта и фона продължава да е важен. Визуалната умора все още често е проблем и трябва да се наблюдава позиционирането на предмета.

Да се избягва използването на снимки. Децата се нуждаят от реални, осезаеми цели (3D). Преминаването към 2D цели или снимки често е трудно. Представянето на цветната снимка на iPad работи добре, поради подсветката, осигурена от iPad.

На това ниво децата могат да видят 2, 3 или 4 добре разположени визуални цели в най-доброто зрително поле за тях. Могат да се изберат приложения за iPad, които предлагат бавно движещи се цели, които се открояват на фона им.(приложение Tap-N-See Now.)

3.3. Стратегии за работа с деца във Фаза 3 – за по-напреднало ниво

В тази фаза децата често демонстрират по-типично визуално поведение, но е възможно да изпитват все още затруднения при гледането на целите, когато са представени в долното им зрително поле. На това ниво все още изпитват затруднения с представянето на твърде много цели, особено когато гледат предметите от разстояние. Може все още да имат слухови и / или тактилни разсейвания.

Тук може да се използва видеонаблюдение за гледане от разстояние. Тези видеонаблюдения позволяват на ученика да увеличи близката цел и да регулира цвета на фона, за да отговаря на неговите нужди. Когато детето се учи да чете, е препоръчително да се използва програма с големи интервали между думи и картинки, както и да се използват разнообразни цветове за фона и за текста.

4. Съществуващи мобилните приложения за деца със CVI

От проучвания относно съществуващи приложения, предназначени за работа с деца с кортикални зрителни увреждания, представени в таблицата 1, се вижда, че намерените приложения с помощта на функцията за търсене в Apple App Store и Google Play Store са идентифицирани за Android - 19, а за Apple - 25. Това показва, че приложенията, създадени за Apple, са много повече от тези за Android. Тъй като в България мобилните телефони с марка Apple са доста по-скъпи от Android, би следвало да се зададе въпросът: Колко от средностатистическите потребители в България могат да си позволят мобилен телефон Apple и дали приложенията, написани за тази операционна система, могат да се използват от нуждаещите се? Отговорът на тези въпрос може да се даде след проучване сред потребителите.

Таблица 1. Списък на съществуващи приложения за операционни системи Apple и Android, предназначени за деца с кортикални зрителни увреждания (CVI)

Apple	Android
• Onni & Ilona by Pintxo Creative • You Doodle by Digital Ruby • Sagomini Music Box • Infant Zoo Lite by treebetty • Big Bang Patterns • Fluidity HD by nebulus design • Liquid Touch by Freescapes Labs • Endless Reader by Originator • Sparkabilities by Sparkabilities • Tap-n-See Now by Little Bear Sees • Sensory Electra by Sensory Apps • Keeble accessible keyboard by AssistiveWare • Peekaboo Barn by Night and Day Studios • ilovefireworks by Fireworks Games • Dr. Seuss Treasury Kids Books by Oceanhouse Media • Cause and Effect Sensory Lightbox by Cognable • Sensory Electra by Sensory Apps • Big Bang Pictures by Inclusive Technology • EDA Play by EDA Play • Bloom HD by EDA Play • Art of Glow by Natenai Ariatrakool • Writing Wizard for Kids by L'Escapadou • Baby View Pocket Lite by Wayne Smith • Learn Shapes with Dave and Ava by Dave and Ava • Infant+ by Dynamic App Design	• Knee Bouncers Big Little Games • My Baby Drum • Kids Xylophone • Little Piano • BalloonMaker • Balloonimals • Pop Goes the Bubble • Sound Touch • Kids Doodle - Color & Draw • Magic Doodle • Magic Paint Kaleidoscope • Neon Drawing • Picasso: Magic Paint • Peekaboo Barn • Maze Ball • Itsy Bitsy Spider • Wheels on the Bus • Five Little Monkeys • Kids Place - Parental Control

От разговори с психолози и други специализирани лица се установи, че има остра необходмост от приложения като част от методиката в ежедневието за работа с деца с такава

проблематика, които да бъдат на български език и които да са друг удобен инструмент в подкрепа на родители и психолози. Това може да бъде подчертано като проблем, който в настоящата статия се дискутира и представлява начална отправна точка за бъдеща работа и разработки на подходящи приложения, пригодени за българските условия на работа.

5. Необходимост от създаването на приложения, съвети за дизайн на приложения спрямо трите фазите на зрителните увреждания и ползите от образователните игри

5.1 Какво е необходимо за създаването на един продукт или приложение?

Когато трябва да бъде разработен един продукт или приложение, в много от случаите се отделя малко време в мислене за съпричастност при проектиране на технологиите [1], а ние възприемаме света чрез комуникацията и сътрудничеството. В книгата си „Креативен отбор“ [4] Кен Косиенда, който е създадел на дигиталната клавиатура в мобилните телефони на Apple, споделя, че ако искаме да създадем продукт, който да намира място в живота на хората и да отговаря на нуждите им, трябва да се опитаме да виждаме света през очите на хората около нас. А съпричастността е решаваща за създаването на добре изглеждащ продукт.

За да бъде създадено едно приложение за деца с кортикални зрителни увреждания, освен съпричастността, която трябва да проявим, трябва да имаме предвид и някоко съвета за дизайн на приложения от специалисти.

5.2 Съвети за дизайн на приложения за кортикални зрителни увреждания [13]

- **За фаза 1** се предлага да се използват приложения с изключени сила на звука и глас, с цел намаляване на разсейването на зрителното внимание на детето.
- **За фаза 2** - да се използват ясни, висококачествени снимки (на реални обекти) за класифициране и сортиране.
- **За фаза 3** - в тази фаза могат да се използват звук и глас. Тук трябва да се използват графики/изображения, които имат значение за индивида, тъй като не може да се предполага, че щом едно избражение се гледа, то може да се интерпретира правилно. В тази фаза е много полезно също, ако буквите или думите се подчертават.

Други съвети за дизайн на приложения

- Да се намали зрителното объркване.
- Да се има предвид зрителната умора и свръх стимулацията

За да бъде изградено едно приложение, то трябва да се отличава както с функционалност, така и с иновативен дизайн [7], като основното умение тук да може да се балансира между логика и емоция. За да може да постигне този баланс, трябва да се използва много умело UX дизайнът, който е отговорен за дизайна на потребителския опит, съвместно с UI дизайна, отнасящ се до потребителския интерфейс. И двете области работят заедно с цел създаване на добър продукт.

Психологията е една от основните области, в които UX дизайнерът трябва да притежава умения. UX дизайнът стъпва върху психологията и това са две взаимосвързани направления.

„Дизайнер, който не разбира човешката психология, няма да бъде по-успешен от архитект, който не разбира физика.“ - Джо Лич

За направата на едно леснодостъпно приложение дизайнерите трябва да облекчат натоварването на хората, които се опитват да използват нещата, създадени от тях, както и че и дребните опростения са от значение [4].

За да бъде направено едно приложение за деца, то трябва да съдържа в себе си игрови елемент, чрез който да се представят стимули и да се поддържа мотивацията, за да бъдат постигнати образователните цели чрез него.

5.3 Образователни игри

Целта на образователните игри [14] е да помогне на хората да научат нещо или да придобият умения, докато играят. Това им дава интелектуално и личностно развитие. Образователните игри имат полезни свойства. Те:

- **Увеличават капацитета на мозъка** - стимулират дейността на човешкия мозък, спомагат за подобряване на функциите му, както и за увеличаване на неговия капацитет.
- **Подобряват комютърните умения**
- **Изграждат стратегическо мислене** - Образователните игри изграждат умение за вземане на решение в напрегнати ситуации. А това може да има множество ползи в бъдещи ситуации.
- **Развиват социални контакти** – едновременно чатят и играят с приятели.
- **Повишават фокуса** - повишават способността на децата за концентрация, фокусиране и насочват усилията към постигане на дадена цел. Това оказва голямо въздействие върху нетърпеливите деца, които лесно се отказват от по-нататъшни опити. Тези деца успяват да се фокусират за по-дълго време, като им се дава визуално представяне на напредък и преминаване на следващо ниво в играта.
- **Са полезни за увереността** - Образователните игри повишават самочувствието на децата и им дават увереност, насърчават креативното мислене. При постигане на цел, децата се чувстват удовлетворени от своите действия и преценки. Това чувство на удовлетвореност може дори да им послужи като стимул за поемане на рискове в други аспекти от техния живот.
- **Освобождават от стреса** - Образователните игри са начин за научаване на нови неща без страх от подигравки.
- **Оказват положителна роля при дефицит на вниманието.**

Образователните игри се делят на три вида:

- **Традиционни образователни игри** - стимулират стратегическото мислене и внимателното планиране на следващите ходове в играта, както и развиват търпение в процеса на преследване на целите. Пример за такава игра е шахматът.
- **Електронни образователни игри** - образователните игри се насочват към тестване на бързите реакции и рефлекси на децата, а други – към развиване на техните знания и умения в отделни области.
- **Поведенчески образователни игри** – оказват положително въздействие върху поведението, както и за оказване на помощ на детето във връзка с преодоляване на различни проблеми, трудности и колебания, които пречат на нормалното му развитие. Това са най-новите модерни образователни игри и като пример за такава е Fast ForWord.

От изложеното до момента относно позитивите, които дават електронните образователните игри и съветите за дизайн на приложения за кортикални зрителни увреждания следва да се обобщи, че създаването на приложения за деца със споменатата проблематика при това на разбираем майчин език ще бъде от огромно значение за тяхното личностностно развитие и прогрес.

6. Подходи за разработка на мобилни приложения

След направения преглед на съществуващи приложения в таблица 1 за операционни системи Apple и Android, могат ясно да се разграничат предимствата и недостатъците на три подхода за реализиране на приложенията:

Таблица 2. Характеристики на технологии за разработка на мобилни приложения

Native	Hybrid	Mobile Web
• Скорост за разработка: **Бавна** • Цена за рaработка и поддръжка: **Висока** • Най-добре за разработка на игри и за потребителски фокусирани приложения • Бързо време за реакция с нулева мрежова зависимост • Възползва се максимално от функциите на телефона • Може да се използва и актуализира със съхранени данни • По-богат потребителски опит	• Скорост за разработка: **Средна** • Цена за рaработка и поддръжка: **Средна** • Може да се свали от различни канали за дистрибуция • Изискват Интернет връзка • Голяма част от кода на приложенията трябва да бъде преизползван • Използва функцията на устройството • По-богат потребителски опит	• Скорост за разработка: **Бърза** • Цена за рaработка и поддръжка: **Ниска** • Изисква Интернет връзка • Комуникация чрез сървър • Не се изискват канали за дистрибуция • Ограничава разработването на богат потребителски опит • На практика няма достъп до функциите на устройството

Таблица 3. Списък на технологии за разработка на мобилни приложения

Native Platform	Hybrid Platform	Cross- Platform
Android **Java, Kotlin** IDE: Android Studio SDK: Android SDK **iOS** **Objective-C, Swift** IDE: Xcode SDK: iOS SDK	**PhoneGap**(HTML5, CSS, JavaScript) **Ionic** is an AngularJS-based framework	**Xamarin**, **React Native** **Titanium** **NativeScript**

Изводи

От разгледаните подходи, представени в таблица 2 и таблица 3 [15], може да се направи извод, че за разработката на мобилни приложения за деца със специфични образователни

потребности е най-добре да бъде избран Native или Crosse-Platform подход, тъй като тези подходи предлагат възможност за разработка на игри и достъп до пълната функционалност на телефона.

Кой от двата подхода ще бъде избран, ще зависи от броя на потребителите, в резултат от проучавания, използващи мобилно устройство с определения вид операционна система.

7. Заключение

За да бъде един продукт лесен за използване, забавен за употреба и удобен за потребителите, е необходимо винаги да има пресечна точка между технологиите и хуманитарните науки [4].

От направеното изложение по-горе може да се обобщи, че технологиите могат да са в помощ за работа с деца с образователни потребности и да бъдат част от методите за работа използвани от специалисти и от родители.

По мнение на специалисти съществуващите налични електронни приложения за работа с деца не винаги са достатъчно ефективни, тъй като вербализирането спомага за по-бързото усвояване и запомняне, а когато то не е на майчин език, това води до спънка в развитието на детето.

Наличието на адекватни електронни образователни приложения биха помогнали на специалистите и на родителите в развитието на детето.

Литература

[1]. Надела, Сатя. Да натиснем Refresh. Да открием духа на Microsoft и да си представим по-добро бъдеще за всички, София, Хермес, 2018, ISBN 978-954-26-1857-7

[2]. Линдстрьом, Мартин. Малките данни, София, Изток-Запад, 2017 ISBN 978-619-152-981-0

[3]. Голман, Даниел. Емоционалната интелигетност, София, Изток-Запад, 2011 ISBN 978-954-321-888-2

[4]. Косиенда, Кен. Креативен отбор, София, Сиела, 2019 ISBN 978-954-28-2866-2

[5]. Ван Хойдох, Ричард. Светът утре, София, Кръгозор, 2019 ISBN 978-954-771-415-1

[6]. D-r Roman-Lantzy, Christine, Cortical Visual Impairment – An Approach to Assessment and Intervention, APH Press,ISBN: 978-0-89128-829-9

[7]. Unger, Russ., Chandler, Carolyn, A Project Guide to UX Design, New Readers, 2012 ISBN 13: 978-0-321-81538-5

[8]. Hutton JS, Dudley J, Horowitz-Kraus T, DeWitt T, Holland SK. Associations between Screen-Based Media Use and Brain White Matter Integrity in Preschool-Aged Children. JAMA Pediatr. 2020; 174(1):e193869. doi:10.1001/jamapediatrics.2019.3869.

[9]. „Влиянието на виртуалната реалност върху психичното развитие на детето от 0 до 3 години“ [Online] Available: https://prakticheska-pediatria.net/2020/02/25/virtualnata-realnost-vlianie-vurhu-psihichnoto-razvitie-na-deteto-ot-0-do-3-god/?fbclid=IwAR1JSniQ1Ecz-d4BveEEjYkyXyn4eu-Uj_Sd-0MwmNkFVGa0zsdpnqYn-Fc [Accessed: 25.02.2020]

[10]. “Как да държим децата далеч от мобилните игри?” [Online] Availabel: http://bulgarian.cri.cn/961/2018/11/01/1s183362.htm [Accessed: 01.11.2018]

[11]. „Поведенчески и психосоматични промени, свързани с използването на технологии в детството“ [Online] Availabel:https://prakticheska-pediatria.net/2020/04/27/povedencheski-i-psihosomatichni-problemi-pri-decata-vuv-vruzka-s-digitalnite-tehnologii/ [Accessed: 27.04.2020]

[12]. CVI Range Assessment [Online] Availabel: https://seeingsadieandcvi.com/cvi-range-assessment/
[13]. Apps for CVI â€“ Give Me Simplicity, Contrast and Movement! [Online] Availabel: https://www.bridgingapps.org/2019/05/apps-for-cvi-give-me-simplicity-contrast-and-movement/
[14]. Образователни игри [Online] Availabel: https://neuro-english.eu/educational-games/
[15]. Three Major Approaches of Mobile App Development [Online] Availabel: https://openxcell.wordpress.com/2017/01/23/three-major-approaches-of-mobile-app-development/

За контакти:
Ass. Prof. Eng. Galina T. Naydenova
Department of Computer Science and Technologies
Technical University of Varna
E-mail: galina.naydenova@tu-varna.bg

Bulgaria Communications Chapter

TU-Varna CSE Computer Science & Engineering Department

LIGHTING SYSTEM CONTROL FOR EVERYDAY AND THERAPEUTIC PURPOSES

Viktor Mashkov

Abstract: The goal of this article is to provide a method for controlling an LED-based, scalable lighting system, where the spectral characteristics of the luminous flux mimic those of daily solar radiation. Into account are also taken users' age group, their momentary desire for the magnitude of the luminous flux and possibilities for therapeutic applications aiming to reduce some hormonal imbalances.
Keywords: LED Lighting, Artificial Light Control, Circadian Rhythm

INTRODUCTION

Through recent years, LED lights have rapidly progressed, offering massive energy savings, reliability and management possibilities. Lately, a growing interest in the latter has occurred. In 2018, the Scientific Committee on Emerging and Newly Identified Health Risks published a paper on "Health Effects of Artificial Light" where possible health risks of different artificial light sources are analyzed [1]. Proposed is regulating different spectral characteristics of light in order to achieve greater cognitive, mood and general health outcomes. Since then, increasing attention has been given to managing the correlated color temperature of the luminous flux of an artificial light source to mimic the diurnal cycle of the summer sunshine [1, 2]. Our aim is to offer a solution which provides control over the correlated color temperature and luminous flux of an LED-based artificial light system with personalization in mind without the use of complex and expensive hardware.

Theoretical investigations

Taking into account numerous research [1-7 and many others], we gather the relevant parameters for this paper. It is shown that human circadian rhythm has a strong dependency on spectral characteristics of ambient light [1, 2]. In addition, with age, the eyes' crystalline lens acquires a yellow tint, affecting perception of color [3]; hence adjustments based on user age are needed. In this paper users are divided into three separate age brackets: young children, teens to middle aged and elderly. Different age brackets affect the peak Correlated Color Temperature (CCT) of the simulated solar cycle. The first age bracket is categorized with little to no yellow tint of the crystalline lens (peak CCT does not exceed 4500K), the second - with little to moderate (peak CCT does not exceed 6500K) and the third - with moderate to high (peak CCT can reach 7500K) [3]. An additional regime of function is added for therapeutic purposes - users are to use it for a limited amount of time - anywhere from 30 minutes to 2 hours for Seasonal Affective Disorder (SAD) management [4].

The system's block diagram is shown in Fig. 1. Firstly, users choose normal or therapeutic function (block 2). The system then calculates appropriate CCT - current CCT for normal mode, as a function of time and the user's age (blocks 3, 4, 8) or as a function of age for therapy mode (blocks 5, 6). Next, calculations are initialized with the lowest achievable by the system CCT - that of the warm white lighting channel (block 9). With respect to the characteristics of the used LEDs, current CCT is calculated (block 11). Then, the algorithm "increments" blue color until a satisfying CCT is reached (blocks 10, 11, 12). The resulting color parameters are converted to PWM signal and sent to the driver circuit depending on the exact hardware used (blocks 13, 15). At this stage we

can increase or decrease the overall luminous flux of the lights without changing their spectral composition (block 14).

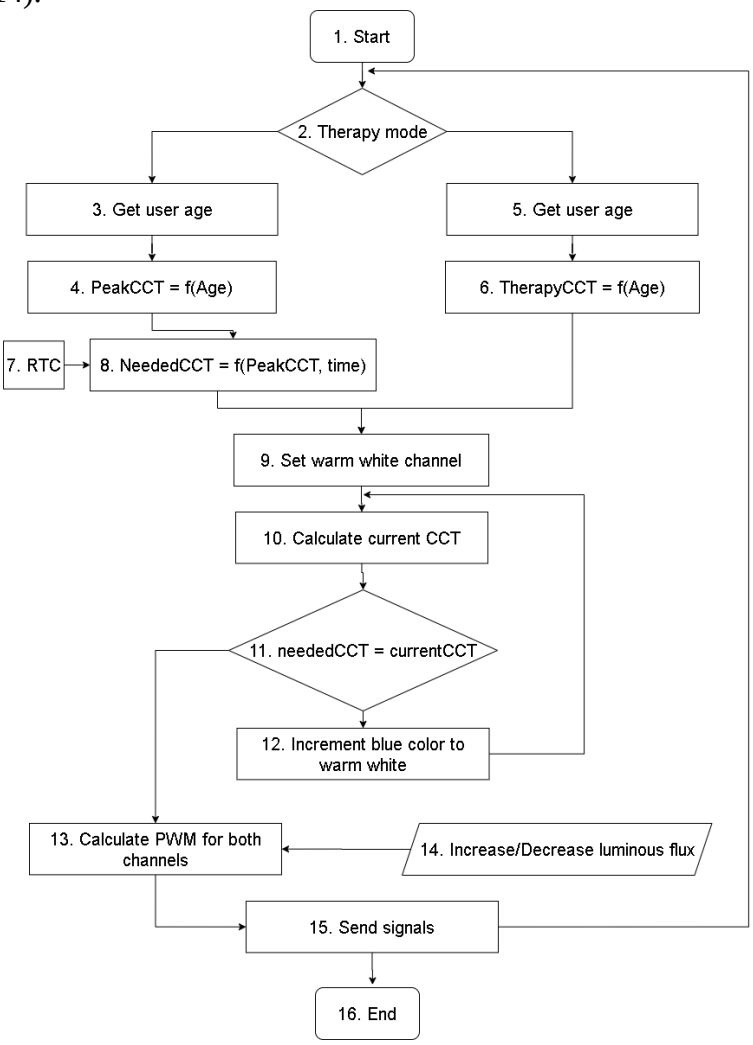


Fig. 1. Block diagram of the used algorithm

Obtaining desired CCT is heavily reliant on the use of the CIE 1931 XYZ color and xy chromaticity space and the McCamy xy chromaticity to CCT cubic approximation. When mixing colors we deploy the vector properties of the aforementioned color space - addition of colors and determination of resulting characteristics are reduced to the corresponding vector operations - addition and multiplication. Assuming X, Y, Z color coordinates, the coordinates at which its directrix pierces the chromaticity plane (the xy chromaticity coordinates) are calculated as follows:

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad z = \frac{Z}{X + Y + Z}$$

LED manufactures provide datasheets [6], containing chromaticity coordinates (x, y) and luminous flux (Y), depending on operating current, voltage and temperature of the p-n junction. With such data in the system we can calculate the resulting color using the formulas:

$$X_{\sum} = m * x_W * \left(\frac{Y_W}{y_W} \right) + n * x_B * \left(\frac{Y_B}{y_B} \right)$$

$$Y_{\Sigma} = m * Y_W + n * Y_B$$

$$Z_{\Sigma} = m * \left(\frac{Y_W}{y_W} \right) * (1 - x_W - y_W) + n * \left(\frac{Y_B}{y_B} \right) * (1 - x_B - y_B)$$

where X_{Σ}, Y_{Σ} and Z_{Σ} are the resulting color coordinates. We can then calculate the xy chromaticity coordinates for the new color.

When a system has achieved the appropriate CCT, but a user wishes to change the luminous flux we can, while maintaining the chromaticity coordinates, adjust both lighting channels by a coefficient, leveraging the vector nature of how our system processes colors.

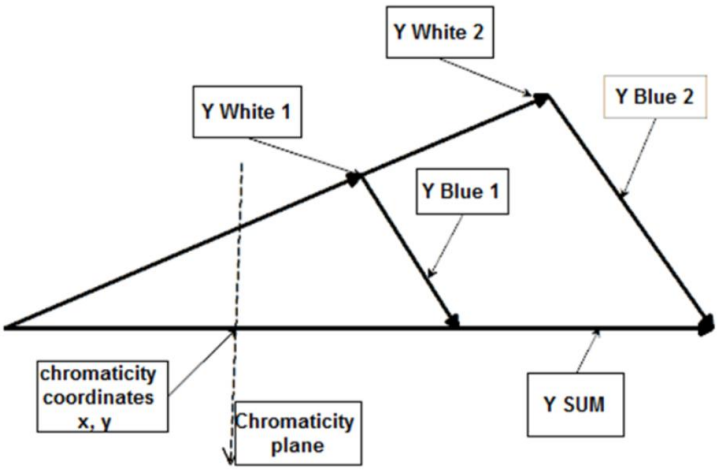


Fig. 2. Y White1, Y White2 – luminous flux of white LEDs before and after change, lm; Y Blue1, Y Blue2 - luminous flux of blue LEDs before and after change, lm; Y SUM - resulting luminous flux, lm

Controlling the LEDs' luminous flux can be achieved with the usage of a true constant current driver. Such drivers allow control over a dedicated PWM channel, therefore operating current can be accurately set (RSE < 1%).

To achieve accurate relative luminous flux control, further exploration is needed. Manufacturers provide a graph, containing relative luminous flux as a function of operating current (Fig.3) [6].

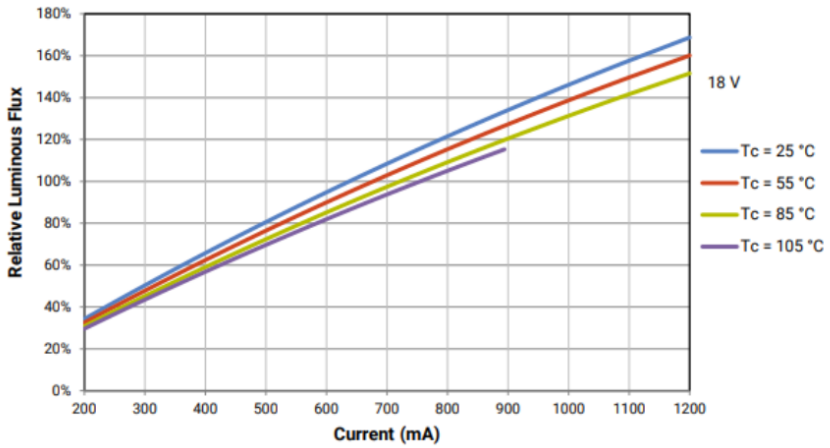


Fig. 3. Relative luminous flux as a function of operating current for CREE CXB1512 2700K, 80CRI LED modules [6]

After the appropriate LEDs and their operating temperatures are chosen we can approximate the current needed to achieve a certain relative luminous flux on a per-case basis. It should be noted that relations between relative luminous flux and current do not present a perfect linear relation, but after further experimental investigation of our linearization method we found minimal deviation ($R^2 > 0.99$), which makes the approximation perfectly acceptable for this paper's use case.
The software part of this project must provide the following capabilities:

- Normal lighting mode – used for everyday purposes with the goal of imitating summer solar radiation. Can be customized based on users' age as follows:
    - Mode 1 – Users are children. The system limits radiation in the blue spectral range (under 500nm). Peak CCT does not exceed 4500K.
    - Mode 2 – Users are teens and adults. Peak CCT can reach 6500K.
    - Mode 3 – Users are elderly. Peak CCT can reach 7500K.
- Multiple therapeutic lighting modes – used for specific situations (battling Seasonal Affective Disorder, depressive states, dementia etc.).
- User and touch-friendly interface that allows users to manipulate the system accordingly.
- This paper proposes a solution based on the Raspberry Pi computer. In addition, to achieve the set goals we used:
- 3 CREE CXB1512 LED modules. These modules provide adequate luminous flux (3000lm at nominal current) and 2700K CCT.
- 8 CREE XQE Blue LEDs. Combined with the aforementioned warm white modules, the system can achieve a wide range of correlated color temperatures at a variety of luminous fluxes.
- 2 MEANWELL LCM-60 LED drivers. The drivers allow control of the current through the LEDs via PWM signals, generated by the Raspberry Pi.

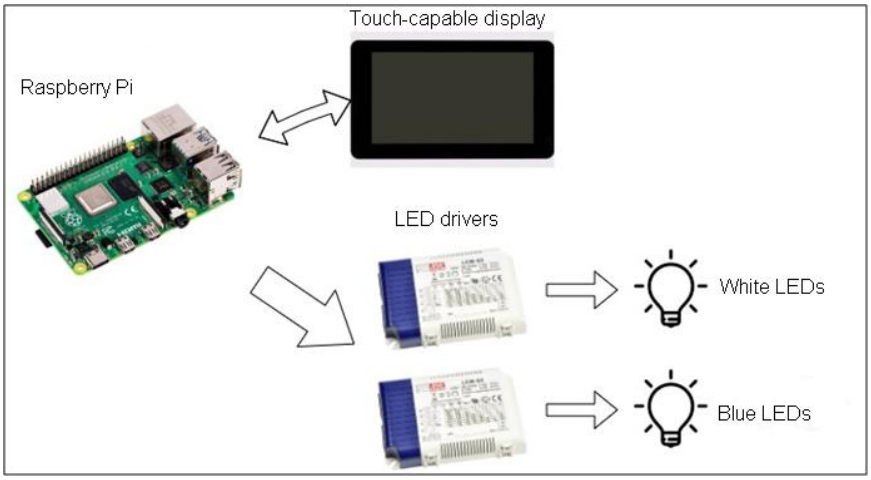


Fig. 4. Block diagram of the main system modules interacting with each other.

Experimental investigations

A number of experiments were performed, concerning the accuracy of the PWM control signals and the overall spectral characteristics the system provides.

The first set of experiments consisted of examining the control signals for both the warm white and blue channels via the use of an oscilloscope. Different working modes were selected and the relation between the desired control signals, the actual control signals and the desired CCT values was examined.

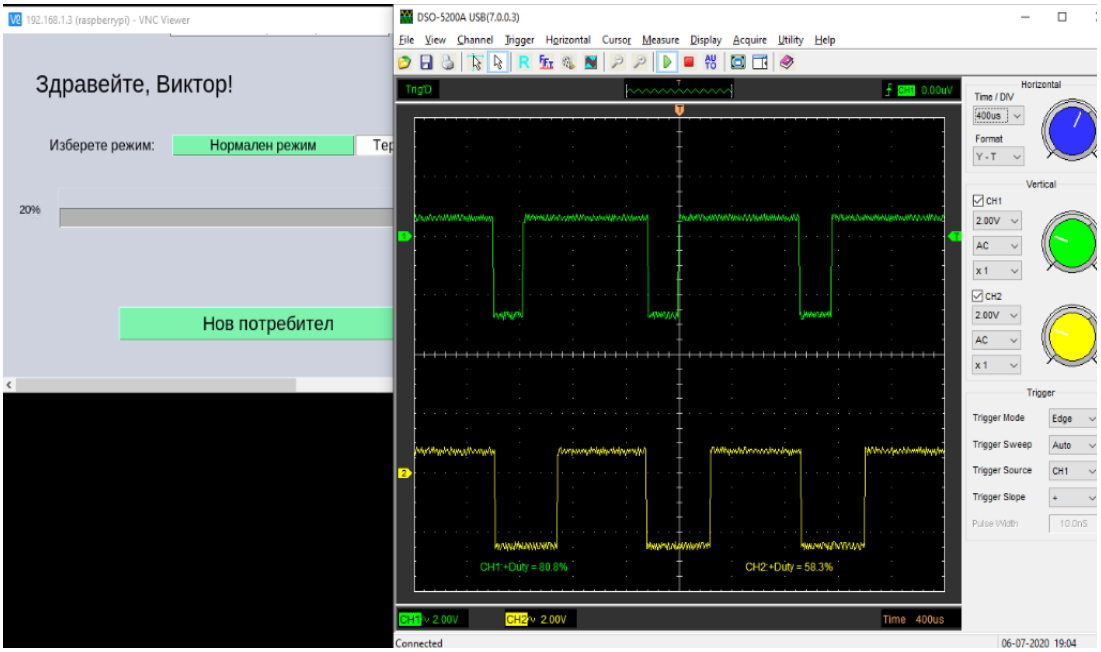


Fig. 5. Exemplary result for therapeutic mode selected (CCT – 7500K, Relative luminous flux – 90%)

The second set of experiments was associated with the spectral characteristics of our illuminator. An integrating sphere and a spectroradiometer were used to investigate changes in photometric characteristics in relation to changing user parameters.

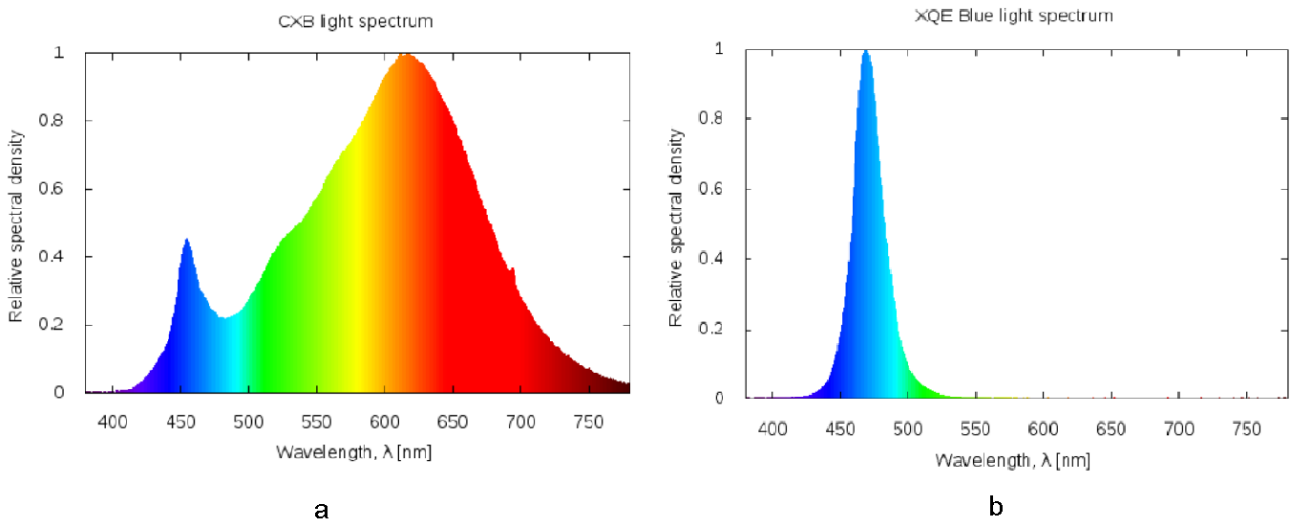


Fig. 6. Experimentally obtained spectral characteristics for used LEDs (5a – CXB1512 warm white, 5b – XQE Blue)

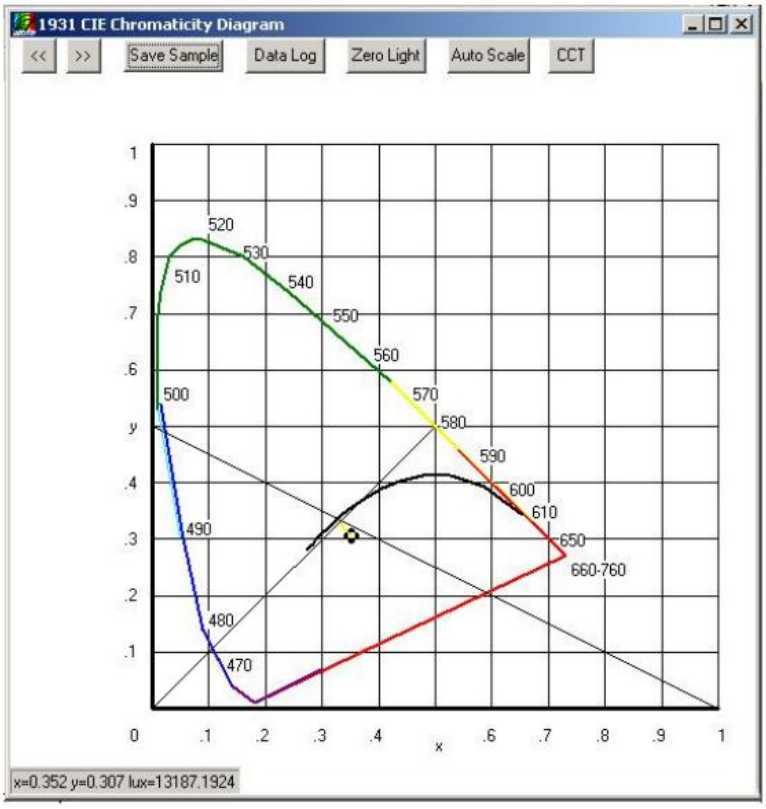


Fig. 7. Exemplary result showing spectral and photometric characteristics of our luminaire for desired CCT – 4500K, measured CCT = 4450K, I_{white} = 350mA, I_{blue} = 185mA, luminous flux Y_{Σ} = 1404lm

Results

The following table shows results for two set CCTs (4400K and 6500K). The relative luminous flux is changed through the system's user interface. The actual luminous flux, correlated color temperature, current and Duv are measured:

CXB1512 I, [mA]	XQE Blue I, [mA]	Flux, [lm]	x	y	CCT, Desired, [K]	CCT, Measured, [K]	Duv
350	185	1404	0,352	0,307	4400	4450	0,027
400	205	1577	0,352	0,307	4400	4315	0,026
500	255	1904	0,353	0,308	4400	4403	0,027
600	310	2256	0,353	0,307	4400	4402	0,027
700	360	2666	0,353	0,307	4400	4369	0,028
350	285	1432	0,320	0,276	6500	6490	0,032
400	328	1632	0,320	0,275	6500	6484	0,032
500	420	2014	0,321	0,275	6500	6480	0,032
600	520	2384	0,321	0,275	6500	6426	0,032
700	632	2709	0,321	0,275	6500	6435	0,032

Conclusion

A lighting management system for controlling the luminous flux characteristics of an LED based luminaire was developed and presented. This paper offers a relatively simple, reliable and scalable solution to adaptive lights. Numerous experiments showed sufficient accuracy without the need for a feedback system, keeping our solution low-cost. The presented system allows CCT control of the luminous flux as well as constant CCT while changing luminous flux. Taken into account are relevant user parameters.

References

[1] Scientific Committee on Health, Environmental and Emerging Risks - SCHEER Opinion on Potential risks to human health of LEDs; 5-6 June 2018.

[2] Gabel V., M. Maire, C. F. Reichert, S. L. Chellappa, C. Schmidt,V. Hommes, A. U. Viola and C. Cajochen, Effects of Artificial Dawn and Morning Blue Light on Daytime Cognitive Performance, Well-being, Cortisol and Melatonin Levels, Chronobiology International, Informa Healthcare USA, Inc. (2013), pp.1–10.

[3] Pescosolido N, Barbato A, Giannotti R, Komaiha C, Lenarduzzi F. Age-related changes in the kinetics of human lenses: prevention of the cataract. Int J Ophthalmol. 2016;9(10):1506-1517. Published 2016 Oct 18. doi:10.18240/ijo.2016.10.23

[4] Virk G, Reeves G, Rosenthal NE, Sher L, Postolache TT. Short exposure to light treatment improves depression scores in patients with seasonal affective disorder: A brief report. Int J Disabil Hum Dev. 2009;8(3):283-286. doi:10.1901/jaba.2009.8-283

[5] McCamy, Calvin S. (April 1992). "Correlated color temperature as an explicit function of chromaticity coordinates". Color Research & Application. 17 (2): 142–144. doi:10.1002/col.5080170211. plus erratum doi:10.1002/col.5080180222

[6] https://www.cree.com/

[7] J.L. Xian, W.X. Cai, H.X. Pan, N.Z. Chen & X.Y. Chen, Y.W. Sun, D. Yan; DNN-HMM-based automatic speech recognition system for intelligent LED lighting control; Automatic Control, Mechatronics and Industrial Engineering – He & Qing (Eds) 2019 Taylor & Francis Group, London, ISBN 978-1-138-60427-8; pp. 73-78.

[8] Covatti, Luís Fernando Peláez, Test-bed for feedback control of color of light of LED lamps; DAS5511: Projeto de Fim de Curso; https://core.ac.uk/download/pdf/78549974.pdf, 2013.

[9] Razvan Cirjan, Sergiu Gheorghe Microchip Technology Inc.; Offline Intelligent RGB LED Lighting System Solution; AN2691; http://ww1.microchip.com/downloads/en/AppNotes/Offline-Intelligent-RGB-LED-Light-System-Solution-00002691A.pdf

For contacts:

Eng. Viktor Mashkov, BSc

Computer Science and Engineering Department

Technical University of Varna

email: mashkoff98@gmail.com

„УМНО ОГЛЕДАЛО“ С RASPBERRY PI И PYTHON

Тодор М. Манолов

Резюме: Статията представя система от тип информационен дисплей „Умно огледало“, която доставя до потребителя метеорологична информация, най-популярните новинарски заглавия и времеви данни, като час, дата и ден от седмицата. Разработката може да действа също така и като огледало. Тя е конкурентна на предлаганите на пазара подобни системи, показва стабилна работа, има опростен дизайн, лесна е за поддръжка.
Ключови думи: Умно огледало, Raspberry Pi, Python, IoT Project

Design and Development of “Smart mirror” using Raspberry Pi and Python

Todor M. Manolov

Abstract: The aim of this article is to design “Smart Mirror” system which delivers to its users weather forecast, the most popular news headlines, time, date and day of the week.It can also be used as a mirror by itself. It competes with other similair systems available on the market, it shows stable operation, has a simplistic design and is easy to maintain.
Keywords: Smart mirror, Raspberry Pi, Python, IoT Project

1.Увод

В наши дни ежедневието е много забързано, има все по-малко и по-малко свободно време и се гледа колкото се може повече да се автоматизират и улеснят задачите, които трябва да се свършат във всекидневието. Извеждането на информация, като час, дата, ден от седмицата, информация за текущата метеорологична обстановка, метеорологична прогноза за следващите няколко дни и най-популярните новинарски заглавия върху обект, който почти всеки има и ползва, като огледало, би подобрило качеството на живот.

Целевите групи са хората, които доста често пътуват по работа,тези които работят във връзки с обществеността и разбира се всеки на когото му допада продукта, смята, че ще му улесни живота и би му бил от полза.

В момента на пазара няма голям избор от информационни дислпеи тип „Умно огледало“. Единственият модел, предлаган на българския пазар, е показан на фигура 1. Съществуват само професионални модели със:

+ Сертифицирана влагоустойчивост по IP65 стандарта
+ Bluetooth свързаност
+ Капацитивен сензорен екран
- Доста висока цена

Извод:

На базата на тези характеристики може да се заключи, че тези модели са предназначени към групата от хора, които са с високи доходи.

А целта на обекта на тази разработка е по-опростен и лесен за поддръжка продукт.

Фиг. 1. Единственият модел, предлаган на българския пазар

Останалите решения, които ще бъдат разгледани, са от типа „направи си сам“ и не могат да бъдат закупени готови.

Решение 1 (фигура 2):

Фиг. 2. Решение 1

Това решение е доста опростено откъм дизайн и изведена информация и не разчита на свързване към електрическа мрежа поради факта, че предназначеното му място е в банята, където влагата може да доведе до инцидент. За захранване ползва външна батерия и консумираният ток е толкова малък, че влажната среда не представлява проблем. Също така притежава възможност за възпроизвеждане на видеоклипове от телефон посредством Google Chromecast, което представлява устройство за поточно предаване към дисплей. Решението има също система против изпотяване, която представлява специално запечатано отделение, което трябва да се напълни с топла вода преди експлоатация, предотвратявайки кондензацията.

Резюме на характеристиките:

+ Влагоустойчивост
+ Енергонезависимост
+ Възможност за поточно предаване
+ Система против конденз
- Твърде малко информация, достигаща потребителя
- През определен период от време трябва да се зарежда батерията, която захранва устройствата

- Доста трудоемко е да се конструира
- Изисква допълнителни части, които не са леснодостъпни

Извод:

Предлага се едно устойчиво на влага и конденз решение, но това и функцията за гледане на видеа не може да компенсира крайно недостатъчната информация, която достига до потребителя, трудното конструиране и нужните допълнителни части.

Решение 2 (фигура 3):

Фиг. 3. Решение 2

Представеното решение е елегантно и по-лесно конструируемо от решение 1, показва приближаващите празници и колко време остава до тяхното настъпване, но отново има липсваща информация за метеорологичната обстановка. Не показва достатъчно новинарски заглавия. Реализирано е с Raspberry Pi.

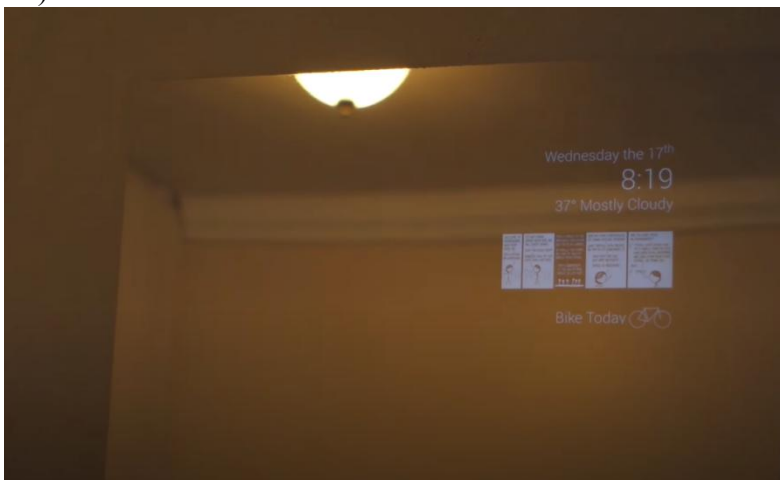
Резюме на характеристиките:

+ Елегантен и приятен за очите дизайн
+ Показване на приближаващи събития и празници
- Недостатъчна информация
- Изисква повече време да се настрои

Извод:

Неподходящо решение с добър дизайн, но с липсваща метеорологична информация.

Решение 3 (фигура 4):



Фиг. 4. Решение 3

Това решение е най-бързо конструируемо и не изисква време за настройка, има приличен дизайн, който може да се променя спрямо желанията на потребителя. Няма възможност за показване на метеорологична прогноза напред във времето. Реализирано е с таблет с операционна система Android, върху който се тегли готово приложение от Google Play Store.

Резюме на характеристиките:

+ Лесно за направа
+ Гъвкав и добър дизайн
- Липса на бъдещи метеорологични данни
- Изисква таблет, който няма да се ползва за други цели
- Неустойчиво на времето

Извод:

По-добро решение от предходно представените. Липсата на по-детайлна персонализация и нуждата от таблет, който няма да бъде ползван за други цели, отсъжда това решение за неподходящо.

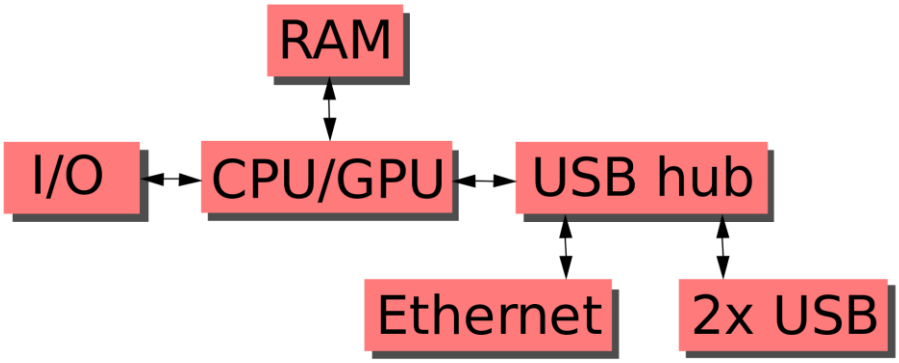
Предложеното тук устройство има по-опростен и лесен за поддръжка информационен дисплей тип „Умно огледало“, което се извършва с помощта на Raspberry Pi, Python, изходно устройство (монитор) и двупосочна отразяваща повърхност.

Целевите групи, за които е предназначена разработката, са хората, които доста често пътуват по работа, тези, които работят във връзки с обществеността и, разбира се, всеки, на когото му допада продукта, смята, че ще му улесни живота и би му бил от полза.

2.Изложение

2.1.Обзор на съществуващи решения

В този проект средствата, които са използвани, са Python и Raspberry Pi. Raspberry Pi е серия от едноплаткови компютри с размер на кредитна карта, създадени са с цел популяризиране на обучението по компютърни науки в учебните заведения [1][2][3]. Представлява едночипова система на Broadcom с процесор ARM и графичен процесор VideoCore IV, с вариращa RAM памет в зависимост от модела и няколко I/O интерфейси, които отново зависят от модела [4][5][6]. Общата схема на свързване на вътрешните компоненти на Raspberry Pi е показана на фигура 5.



Фиг. 5. Обща схема на свързване на вътрешните компоненти на Raspberry Pi

Други подобни едноплаткови компютри, с които би могло да се осъществи подобно решение, са примерно Arduino и BeagleBone. Arduino е създадено в Италия с цел проектиране и производство на платформа за достъпни, свободни и лесни за ползване хардуер и софтуер за хора, които не са специалисти[7]. Електронните платки може да се закупят готови сглобени или като комплект „направи си сам“ поради причината, че схемите

им са публично достъпни за всеки, който иска да си ги сглоби сам. В основата на Arduino стоят 8-битов Atmel AVR микроконтролер с допълващи се компоненти или 32-битови ARM процесори Atmel [8]. С характеристиките на Arduino не би могло да се осъществи този проект по предварително предвидения начин заради изключително ограничените му компютърни ресурси (фигура 6).

	Arduino Uno
Price	$30
Size	7.6 x 1.9 x 6.4 cm
Memory	0.002MB
Clock Speed	16 MHz
On Board Network	None
Multitasking	No
Input voltage	7 to 12 V
Flash	32KB
USB	One, input only
Operating System	None
Integrated Development Environment	Arduino

Фиг. 6. Спецификации на Arduino Uno

От таблицата можем да заключим, че ресурсите, които предлага Arduino, са крайно недостатъчни. По-подходящо би било за проекти, в които се управлява хардуер, било то датчици или други изпълнителни устройства.

BeagleBoard е едноплатков компютър с отворен код с ниска мощност, произведен от Texas Instruments в сдружение с Digi-Key и Newark element14. Проектиран е също така със софтуер с отворен код отново за учебни цели [9].

Спецификации(BeagleBone Black):

- 512MB DDR3 RAM
- 4GB 8-bit eMMC on-board flash storage
- 3D graphics accelerator
- NEON floating-point accelerator
- 2x PRU 32-bit microcontrollers
- USB client for power & communications
- USB host
- Ethernet
- HDMI
- 2x 46 pin headers

Откъм характеристики и системни ресурси виждаме, че е напълно достатъчно. Откъм цена Raspberry Pi определено превъзхожда BeagleBone с почти два пъти по-ниска цена. Raspberry Pie е изключително универсален компютър и може да се ползва почти за всичко. BeagleBone има двойни 46-пинови заглавки, които го правят доста по-подходящ за проекти по роботика и хардуерни такива. След оглед на възможните ресурси за реализация може да се заключи, че Raspberry Pie е най-подходящо.

На Raspberry Pi е най-удобно и лесно да се пише на C или Python. Python е интерпретаторен, интерактивен, обектно ориентиран език за програмиране, създаден е от Гуидо ван Росум в началото на 90-те години. Предлага добра структура и поддръжка за разработка на големи приложения. Той притежава вградени сложни типове данни като гъвкави масиви и речници, за които биха били необходими дни, за да се напишат ефективно на C [10][11]. Език от високо ниво е, което значи, че е по-бавен от други езици от по-ниско ниво като C, но това не го прави по-лошия избор. За приложения, за които скоростта не е от особено голямо значение, е по-добре да се разработят на Python, заради:

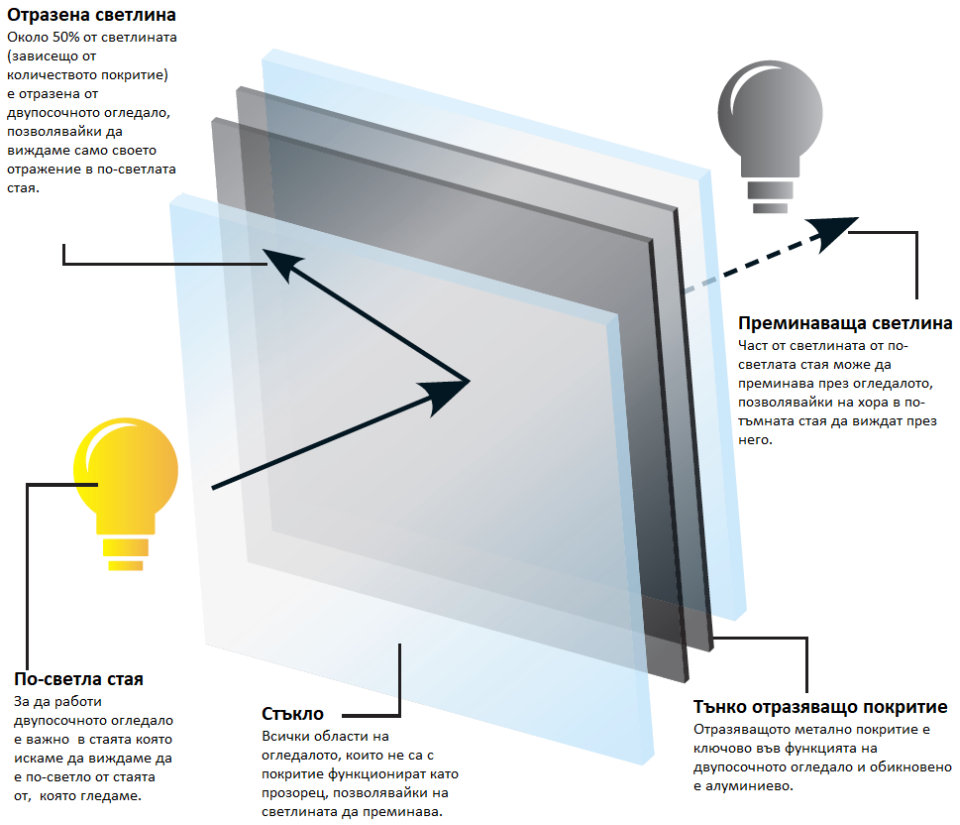
+ Python е по-продуктивен
+ Богат избор от библиотеки
+ Голяма общност
+ Компаниите могат да оптимизират най-ценния ресурс - работниците
+ Кодът изглежда по-чист и подреден
+ Кодът е по-лесен за подръжка и с по-малък обем [12]

Изтъкнатите предимства определят избора на Python като най-подходящ.

За да се направи обоснован избор на огледалната повърхност, която ще се използва, първо трябва да се разбере как точно работят огледалата.

Обикновено огледалата се изработват от стъкло, покрито с метален слой, най-често алуминий. Когато светлината премине през стъклото и стигне до метала, тя се отразява, което е причината да виждаме себе си, когато погледнем в огледалото.

Двупосочните огледала се изработват от същите материали, но металният слой е доста по-малко. Както се вижда на фигура 7, ако само половината повърхност на стъклото е покрита с отразяващи молекули, двупосочното огледало отразява само половината светлина, която е достигнала до него, което означава, че останалата светлина достига до другия край. Докато стаята от другата страна на огледалото е тъмна, ще бъде възможно да се вижда през огледалото в по-светлата стая от по-тъмната стая [13].



Фиг. 7. Визуално представяне на принципа, по-който работят двупосочните огледала [13]

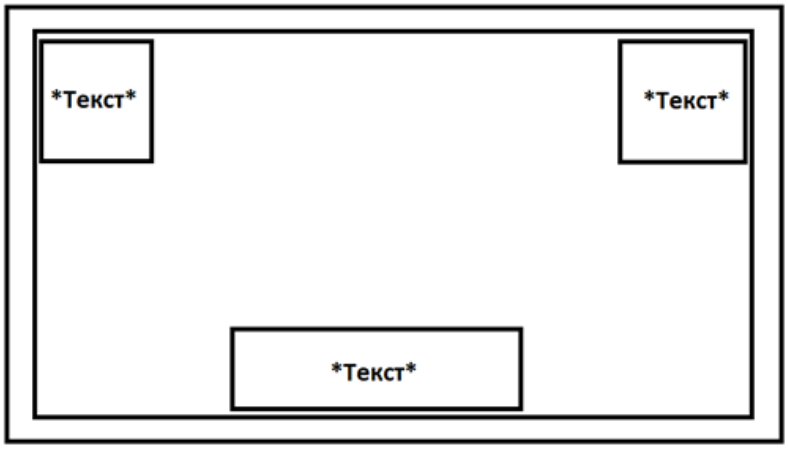
2.3. Проектиране на приложението

За да се постигне най-голяма отражаемост, трябва фонът да е с цвят, който поглъща най-много светлина, поради тази причина най-удачно е да се използва черен цвят за приложението. За шрифта се използва бял цвят заради големия контраст, който се получава между бялото и черното, което от своя страна осигурява четливост.

Важен е и въпросът с избора на шрифт за извежданите надписи.

Фиг. 8. Разликата между серифен и несерифен шрифт

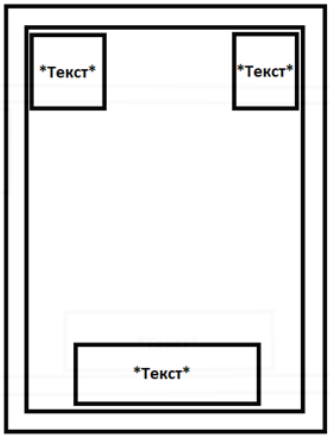
И двата типа шрифтове, показани на фигура 8, имат своите ползи и недостатъци и предават различни съобщения. Например серифните шрифтове датират още от 18-ти век, когато са били издълбавани в камък. Днес се виждат серифни шрифтове в традиционни медии като вестници, списания и книги. Затова този тип шрифтове се разпознават като класически и вдъхват чувство за традиция, затвърденост и доверие. Несерифните шрифтове от друга страна са опростени и са символ на модерното. Основните характеристики на този тип шрифтове са липсата на серифи и изчистените и опростени линии с еднаква дебелина. Тези характеристики са основната причина много уеб дизайнери да предпочитат този тип шрифтове. Изчистените линии и ясно изразените върхове се изобразяват по-ясно на екрана и увеличават четливостта [14]. Това прави несерифните шрифтове по-добрият избор. Популярни такива са: Helvetica, Open Sans, Proxima Nova, Arial и др.В този проект се използва Helvetica.



Фиг. 9. Скица на разположение на информацията

За да се постигне маскимално голяма повърхност, в която може ясно да се види отражението, е хубаво текстът да се разположи в краищата на монитора - така се постига

определена подреденост и централната част на огледалото е освободена от нарушение на черния фон (фигура 9).

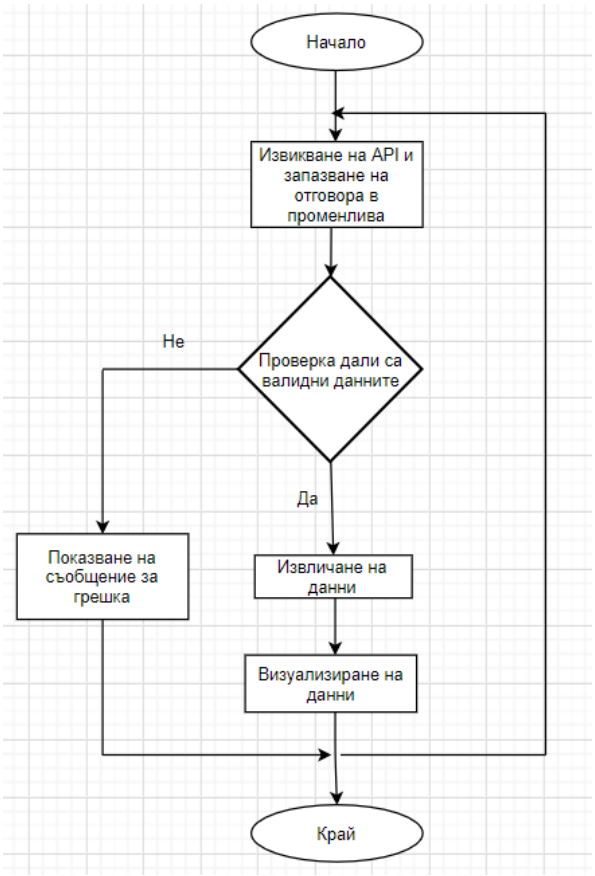


Фиг. 10. Разположение на информацията в приложението във вертикална ориентация

Завъртането на монитора във вертикална ориентация би предоставило възможност огледалната повърхност да се употреби по-оптимизирано, защото позволява да използваме повече от нея (фигура 10).

2.4. Програмна реализация

Реализацията е осъществена с помощта на библиотеките на Python Tkinter, time и requests. Обобщената блокова схема на алгоритъма за функциите извличащи данни е представена на фигура 11.



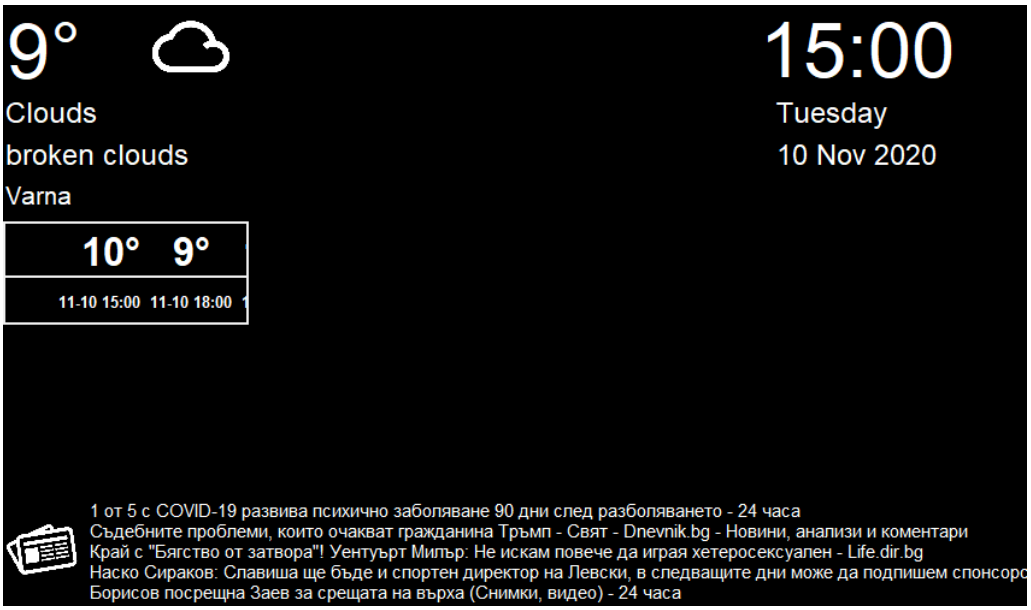
Фиг. 11. Обща блок схема на алгоритъма за функциите извличащи данни

Приложно програмният интерфейс (API), най-общо казано, предоставя един по-абстрактен и опростен план за разработчика на приложения, който би му спестил изучаването на няколко различни слоя от Операционната или софтуерната система зад интерфейса. По този начин се достига ефективност и бързина при адаптирането на нови софтуерни технологии. В този случай ни позволява да се установи връзка с големи бази от данни, които съдържат информация и прогнози за метеорологичното време.

За получаване на данните се използват различни API:

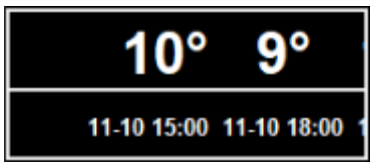
- API от OpenWeatherMap за текущата метеорологична обстановка.
- API от OpenWeatherMap за прогнозирана метеорологична обстановка.
- API от NewsApi за най-популярните новинарски заглавия.

3. Резултати



Фиг. 12. Как изглежда приложението

В горен десен ъгъл се вижда час в 24 часов формат, под него - денят от седмицата, а под него - днешната дата във формат „дата/месец/година“. В горния ляв ъгъл се изписва температурата в момента навън в градуси по целзий, вдясно от нея има икона, която илюстрира метеорологичните условия, под тях - описание на текущото атмосферно време с думи и малко по-подробно описание на същото под него, а по-отдолу – градът, за който се отнасят тези данни. Всички метеорологични данни се обновяват на всеки 10 минути, а новинарските заглавия - на всеки 1 час.



Фиг. 13. Метеорологична прогноза по дни и по часове

Това е метеорологичната прогноза за следващите 5 дни във фиксирани часове, които са: 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00, 21:00. Температурите се движат заедно с датите под тях, от дясно наляво. Всяка температурна стойност отговаря на датата под нея, която е във формат „месец-ден час“ (фигура 13).

Заключение

Представената система от тип информационен дисплей „Умно огледало“ доставя до потребителя метеорологична информация, най-популярните новинарски заглавия и времеви данни, като час, дата и ден от седмицата. Разработката може да действа също така и като огледало. Проектираният софтуер дава следните възможности:

- Да се следи температурата навън в реално време с актуализация през 10 минути.
- Да се следи температурата за следващите 5 дни за фиксирани часове с актуализация през 10 минути.
- Да се следи часът, датата и денят от седмицата в реално време.
- Да се следят последните най-популярни новинарски заглавия с актуализация през един час.

Тестовете показват стабилна работа на системата и навременна актуализация във времето на данните в нея.

Системата е конкурентна на предлаганите решения, превъзхождайки ги в следните направления:

- Опростен и интуитивен дизайн;
- Точни метеорологични и времеви данни;
- По-подробна информация достигаща потребителя.

Предложената система би могла да бъде усъвършенствана, интегрирайки в нея изкуствен интелект с разпознаване на гласови команди и жестов контрол.

Литература

[1]. Cellan-Jones, Rory (5 May 2011), "A£15 computer to inspire young programmers", BBC News, 5 май 2011.

[2]. Price, Peter. Can a £15 computer solve the programming gap?. // BBC Click, 3 юни 2011.

[3]. Bush, Steve. Dongle computer lets kids discover programming on a TV. // Electronics Weekly, 25 май 2011.

[4]. BCM2835 Media Processor; Broadcom. // Broadcom.com, 1 септември 2011.

[5]. Brose, Moses. Broadcom BCM2835 SoC has the most powerful mobile GPU in the world?. // Grand MAX. 30 януари 2012.

[6]. Model B now ships with 512 MB of RAM. // Raspberrypi.org.

[7]. Arduino Project introduction

[8]. Arduino.cc

[9]. Coley, Gerald (August 20, 2009). "Take advantage of open-source hardware"

[10]. Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. 23 June 2012.

[11]. van Rossum, Guido (20 януари 2009). "A Brief Timeline of Python". *The History of Python*

[12]. https://medium.com/@trungluongquang/why-python-is-popular-despite-being-super-slow-83a8320412a9

[13]. https://www.howitworksdaily.com/how-do-two-way-mirrors-work/

[14]. https://www.impactbnd.com/blog/sans-serif-vs-serif-font-which-should-you-use-when#:~:text=A%20serif%20is%20a%20decorative,hence%20the%20%E2%80%9Csans%E2%80%9D).

За контакти:
инж.Тодор М. Манолов
катедра „Компютърни науки и технологии“
Технически университет - Варна
E-mail: todormanolov97@gmail.com

ПРОЕКТИРАНЕ И РЕАЛИЗАЦИЯ НА СИСТЕМА ЗА ПОДРЕЖДАНЕ НА РУБИК КУБ

Павлин Д. Щерев

Резюме: Настоящата статия представя проектиране и реализация на система за подреждане на Рубик куб чрез прилагане на начинаещия метод за подреждане. Решаването на куба се изпълнява от робот, който разпознава различните страни и цветовете им чрез алгоритъм за обработка на изображения. Стъпките за подреждането се изпълняват софтуерно, след което решението се изпълнява от робота. Оптималността на алгоритъма се постига чрез използване на нишки, намиращи най-краткото решение.
Ключови думи: алгоритъм, изображения, робот, нишки.

Design and Implementation of a System for Solving a Rubik's Cube

Pavlin D. Shterev

Abstract: The current article presents the design and implementation of a system for solving Rubik's Cube, by applying the beginner's method for solving. The solving of the cube is performed by a robot that recognizes the different sides and their colors using an image processing algorithm. The steps for the arrangement are performed by a software, after which the solution is executed by the robot. The optimality of the algorithm is achieved by using threads that find the shortest solution.
Keywords: algorithm, images, robot, threads.

1. Увод

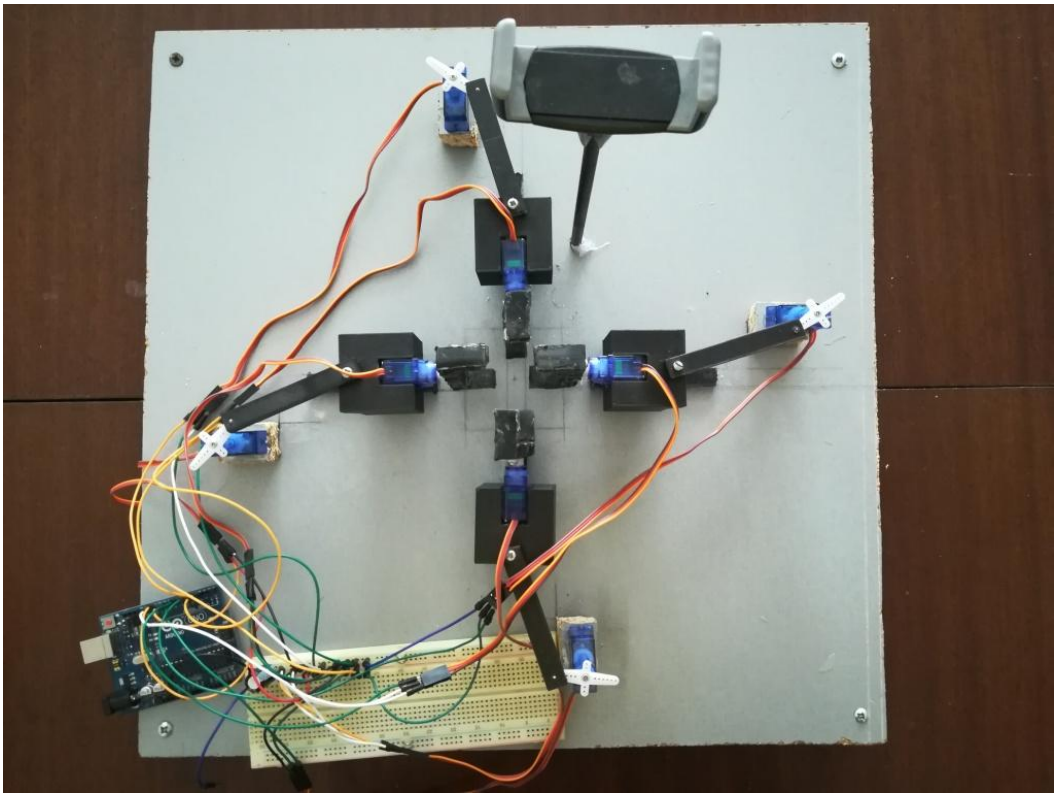
В наши дни иновациите бързо навлизат във всички сфери на индустрията, включително в сектори, в които работата с десетилетия се е извършвала по един и същ начин. С разрастването и прогреса в почти всяка сфера от човешката дейност, компютърните технологии навлизат все по-осезаемо навсякъде около нас. Технологиите се развиват в бърз темп и се стремят към бързина и надеждност.

Поради точно тези качества роботите бързо намериха място в живота ни. С помощта на тях се разви бързодействието и точността в промишлените дейности, а всички прогнози сочат, че и за в бъдеще тяхното разпространение ще нараства. Все по-често започваме да виждаме роботите и в домашна обстановка, свързани с ежедневни домакински дейности – типичен пример е прахосмукачка-робот или дори развлекателни активности. Възможно е и човек да се учи от робот, като наблюдава действията му, а това добавя и учебна сред многото ползи на роботизацията.

Обект на разработката е създаването на апаратно-програмна система-робот за подреждане на Рубик куб.

В основата на системата за подреждане на Рубик куб стои начинаещият алгоритъм. Избраният алгоритъм се имплементира, действието му се оптимизира и се комбинира с алгоритъм за намиране на най-краткия път за подреждане на кубчето. Следва изработване на графичното оформление на куба, което да онагледи прилагането на алгоритъма и да улесни проследяването на стъпките и това дали са успешни. Разработен е дизайнът на робота, а също така е създаден и визуален 3D модел на частите на робота. Частите са принтирани чрез 3D принтер. Комуникацията между софтуера и хардуера се осъществява чрез серийна връзка. Изпълнението на програмата започва с направата на снимки, четенето и анализирането им,

изчисляване на решението и намирането на най-краткия път за нареждането на куба. След тази последователност от действия, към робота се изпращат команди за изпълнение на движенията, нужни за нареждането на кубчето.



Фиг. 1. Апаратно-програмна система-робот за подреждане на Рубик куб

В основата на начинаещия алгоритъм [1] стои направата на бял кръст върху бялата страна, чието оформяне няма определен алгоритъм, а става интуитивно. Белият кръст трябва да е правилен – цветовете, тръгващи от страните на кръста, трябва да съвпадат с центъра на страната на кубчето. Следва втора стъпка – белите ъгли, с която завършва с направата на цялата бяла страна. Трета стъпка е направата на втория слой, който вече е започнал да се образува след правилното нареждане на бялата страна. След като вторият слой е готов, следва подреждането на противоположната на бялата страна, а именно жълтата, като отново се придържаме към фигурата кръст. Свързването на цветовете на страните с ръбовете на жълтия кръст е стъпка номер пет, като по този начин вече всяка от страните е с нареден централен ред. Предпоследната стъпка се достига, когато са останали ненаредени само последните ъгли на слоя под жълтата страна. Жълтите ъгли трябва да бъдат поставени на правилното място, което се получва при следването на подходящ алгоритъм. Последна стъпка е завъртането на жълтите ъгли, за да се нареди кубчето, като накрая се получава един готов Рубик куб [2].

2. Алгоритъм за обработка на изображения за определяне цветовете на плочките

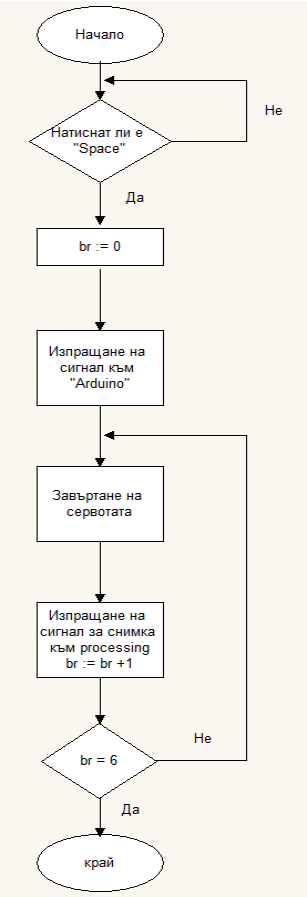
При стартиране на системата първата стъпка е да се направи връзка между компютъра и камерата. След това започва самото взимане на снимките от куба. Щом „space“ (начало) бива натиснат, към Arduino платката се изпраща променлива от тип „char“ със стойност “C”. Ако има нещо записано върху серийния порт на Arduino, в частта „loop“ на Arduino има switch, който според стойността на променливата „char“ изпълнява различни функции.

При стойност “C” се изпълнява функцията Capture, като тя завърта кубчето във всички посоки. Между всяко от завъртанията на различните страни към средата Processing се изпраща стойност “P” (Photo) на променливата тип „char“. Програмата чака за получаване на стойност “P”, като щом я получи, се прави снимка единствено на позиционираното в квадрата изображение. Това изображение се запазва в PImage променлива, която е част от вграден клас в средата Processing. Получават се шест подобни PImage променливи, които са запазени в отделен масив (img).

Програмата съдържа брояч, който брои колко пъти е изпратена променливата P към Arduino. Щом този брояч достигне стойност шест, приложението преминава към picturesProcessing – функция, в която за всяка от снимките се взимат пикселите, чрез които за всяка плочка от страната на куба се определя нейният цвят. Пикселите за всяка снимка се записват в ArrayList. Тези пиксели имат числова стойност. След записването на всички пиксели на дадена снимка, те се изпращат на функцията hueToString. Тази функция взима стойностите на всеки пиксел и връща като резултат цвета под формата на string. Функцията сама по себе си съдържа switch, чрез който се определя кой цвят да бъде преобразуван. Всеки от цветовете се съдържа в даден числов диапазон, спрямо стойностите на hue.

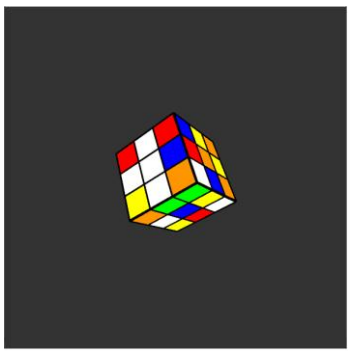
След взимането на цветовете трябва да се определи кой цвят е най-често срещаният в дадена плочка. Това се случва чрез функцията mostCommon, която връща като резултат най-повтарящия се цвят. Тези действия се извършват за всяка една страна и за всяка една плочка, а резултатът от тези действия се записва в двумерен масив от тип string. След това се създава виртуално кубче, на базата на което се създават други шест невизуализирани кубчета.

Алгоритъмът е представен на фигура 2.



Фиг. 2. Взимане на страните на куба

3. Алгоритъм за подреждане, базиран на начинаещ метод



Фиг. 3. Начална инициализация на Рубик куб

За реализацията на алгоритъма е необходимо някои от стъпките да бъдат декомпозирани на по-малки такива.

Първата стъпка, която се предприема за направата на белия кръст, е следната: проверява се дали на бялата страна има бял ръб, който да съвпада с белия център (фигура 3). Ако няма такъв съвпадащ бял ръб, се преминава към следващата стъпка. Ако обаче има съвпадащи бели ръбове, се проверява най-добрата възможна ситуация от тях. Под най-добра възможна ситуация се разбира максималната бройка от съвпадащия друг цвят на тези ръбове със съответния цвят на център. Това се проверява чрез завъртане на бялата страна четири пъти – докато бъде преминато през всяка възможна комбинация на бяло завъртане. Всяко завъртане на дадена страна четири пъти завършва с началното състояние на страната.

Втората стъпка започва с проверка дали на жълтата страна – противоположната на бялата страна – има бели ръбове. Ако има такива, за всеки бял ръб се изпълнява следната последователност от стъпки: той се завърта толкова пъти, колкото е нужно, за да съвпадне другият му цвят със съответния си център. Щом това се случи, дадената страна се завърта на 180 градуса, което означава, че белият ръб е отишъл на бялата страна. Това се изпълнява за всеки от белите ръбове върху жълтата страна.

Третата стъпка проверява дали има бели ръбове от бялата страна, чийто друг цвят не съвпада със своя център. Ако има такъв бял ръб, то той се завърта на 180 градуса – отива на жълтата страна, и се извиква функцията, обработваща втората стъпка. Това се изпълнява за всеки от тези бели ръбове, които не са на правилната си позиция. Накрая на изпълнението на стъпката не трябва да има останали бели ръбове върху жълтата страна.

Четвъртата стъпка проверява дали другият цвят на ръбовете на жълтата страна е бял. Ако този друг цвят е бял, се изпълнява стъпка, подобна на втората, а именно жълтата страна се завърта, докато другият цвят на белия ръб не съвпадне със съответния си център. Тази ситуация означава, че този ръб е на правилното си място, но не е правилно ориентиран. Този ръб трябва да бъде поставен върху бялата страна, като се изпълни общоприет алгоритъм за завъртане. Стъпката се изпълнява за всеки такъв ръб.

При петата стъпка се извършва подобна последователност от действия като тези от предишната стъпка, но за белия цвят се използва завъртащият алгоритъм и неправилно ориентираните бели ръбове се качват на жълтата страна. След това се извиква функцията, която обработва втората стъпка.

Шестата стъпка разглежда средния слой на кубчето и се търси дали в този слой се съдържа ръб с бял цвят. Ако има такъв бял ръб, се проверява дали другият цвят на ръба съвпада със съответния си център.

Ако това е изпълнено, страната с другия цвят се завърта към бялата страна. Ако това не е изпълнено, тази страна се завърта към жълтата страна. След това се извикват двете функции, които обработват втора и четвърта стъпка, защото не се знае дали при завъртането на страната този бял ръб е ориентиран правилно или неправилно. Независимо от

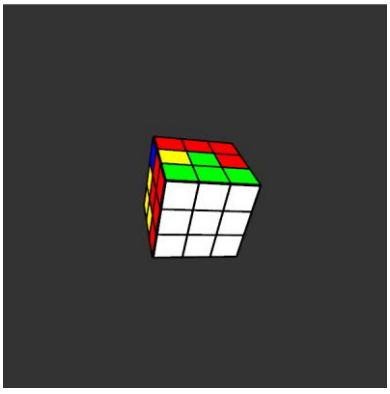
ориентирането на ръба, функцията е имплементирана по такъв начин, че да провери дали има ръб от дадения вид – ориентиран или не, а ако няма, изпълнението на функцията се преустановява. Стъпката се изпълнява за всяка наличност на бял ръб в средния слой. Тя завършва с образуването на готов бял кръст.

Седма стъпка започва нареждането на белите ъгли. Първоначално се разглеждат белите ъгли на жълтия слой, тъй като той е противоположен на белия и така със сигурност се гарантира, че даденият ъгъл ще отиде на правилното си място и няма да бъде разместен след това. Ако има бял ъгъл на жълтия слой, той трябва да бъде завъртян по такъв начин, че да бъде между страните на другите си два цвята.

След това този ъгъл отива на белия слой. Зависи обаче дали белият цвят на ъгъла съвпада с жълтата страна. Ако това е така, ъгълът ще бъде неправилно ориентиран, което означава, че отново трябва да отиде на жълтия слой, а след това да се върне обратно на белия. В противен случай тази последователност от действия не е необходима. Описаните стъпки трябва да бъдат повторени за всеки от белите ъгли на жълтия слой.

Осма стъпка проверява дали в белия слой има бял ъгъл. Ако това е така, трябва да се провери дали този бял ъгъл е на правилното си място и дали е правилно ориентиран.

Ако не е, неориентираният бял ъгъл отива на жълтия слой, след което се изпълнява функцията от предишната стъпка. След изпълнението на тази стъпка трябва да има готов бял слой – всички ръбове и ъгли да са на правилните си места и да бъдат правилно ориентирани (фигура 4).

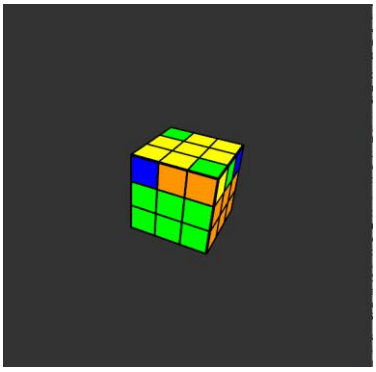


Фиг. 4. Завършен първи слой

Девета стъпка служи за нареждане на средния слой. По време на тази стъпка вече се използват общоприетите алгоритми за пермутация на слоя. Преди използването на съответните алгоритми обаче, се изисква подготовка на кубчето. Трябва да бъде проверено дали на жълтия слой има ръбове, които нямат жълт цвят. Жълтата страна трябва да бъде завъртяна толкова пъти, че този цвят на ръба, който е отстрани на жълтата страна, да съвпадне с центъра си.

След необходимия брой завъртания има два варианта на алгоритъма – ляв или десен. Изборът на вариант зависи от цвета на ръба, който е на жълтата страна. Целта е съответният ръб да бъде вмъкнат между страните на другите си два цвята. Левият алгоритъм се изпълнява чрез последователността U'L'ULUFU'F', докато десният чрез последователността URU'R'U'F'UF.

Има вариант, в който някой от ръбовете да е на правилното си място, но да не е правилно ориентиран. Тогава за този ръб трябва да се използва един от двата алгоритъма, без значение кой от тях, за да може този ръб да бъде преместен на жълтия слой. След това отново се използва левият или десен алгоритъм, по такъв начин, че ръбът да бъде вмъкнат между страните на другите си два цвята – да бъде правилно ориентиран. Действията се извършват за всеки от ръбовете, като в края на стъпката се получава завършен втори слой (фигура 5).



Фиг. 5. Завършен среден слой

Десетата стъпка започва нареждането на третия слой. Първо се проверява колко от ръбовете на жълтата страна съвпадат с жълтия център. Спрямо този брой се изчислява колко пъти да бъде използван следният алгоритъм: FRUR'U'F'.

Ако нито един от ръбовете на жълтата страна не съвпада със своя център, то алгоритъмът се изпълнява три пъти. След изпълнението на втория път е необходимо кубчето да бъде завъртяно на 90 градуса. След това алгоритъмът се изпълнява още веднъж. Няма как да има или един-единствен, или три съвпадащи жълти ръба с жълтия център, като, ако все пак се случи такава ситуация, то или първите два слоя на куба не са подредени, или кубчето е било разглобено и сглобено неправилно. Ако има два съвпадащи ръба, то те могат да бъдат разположени в два варианта. Първият е под формата на „Г“, а вторият - под формата на „-“. Ако вариантът е първият, горният алгоритъм се изпълнява веднъж, разположението на цветовете става под формата на „-“, след което алгоритъмът се изпълнява още веднъж. Ако още в началото разположението е по втория вариант, е нужно само едно изпълняване на алгоритъма. Така се подрежда жълтият кръст.

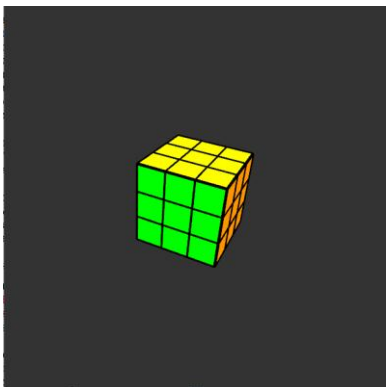
Единадесетата стъпка се състои в правилното ориентиране на жълтия кръст. Първо е нужно жълтият слой да се завърти по такъв начин, че два от другите цветове на ръбовете на слоя да съвпадат със своя център.

След това има два варианта на разположение – съвпадащите ръбове да бъдат един срещу друг или един до друг. При първия вариант е необходимо даденият алгоритъм – RUUR'U'RU'R'U' – да бъде изпълнен два пъти, като между двете изпълнения е нужно кубът да бъде завъртян на 90 градуса. При втория вариант е необходимо едно реализиране на алгоритъма. След изпълнението на алгоритъма се получава правилно ориентиран жълт кръст.

Стъпка дванадесет е правилно позициониране на ръбовете на жълтия слой. Правилното позициониране е изпълнено, ако цветовете на дадения ъгъл съвпадат с цветовете на трите страни, върху които той лежи. Първо се проверява дали ръбовете не са вече подредени, като, ако са, то тази стъпка се пропуска. Ако нито един от тях не е правилен, е нужно изпълнението на следния алгоритъм – L'URU'LUR'U'. След изпълнението на алгоритъма поне един от ъглите ще бъде правилно позициониран. След намирането на този ъгъл, кубът трябва да се завърти по такъв начин, че ъгълът да бъде поставен в предния горен десен ъгъл на кубчето. Алгоритъмът се повтаря, докато всички ъгли не бъдат поставени на правилните си места.

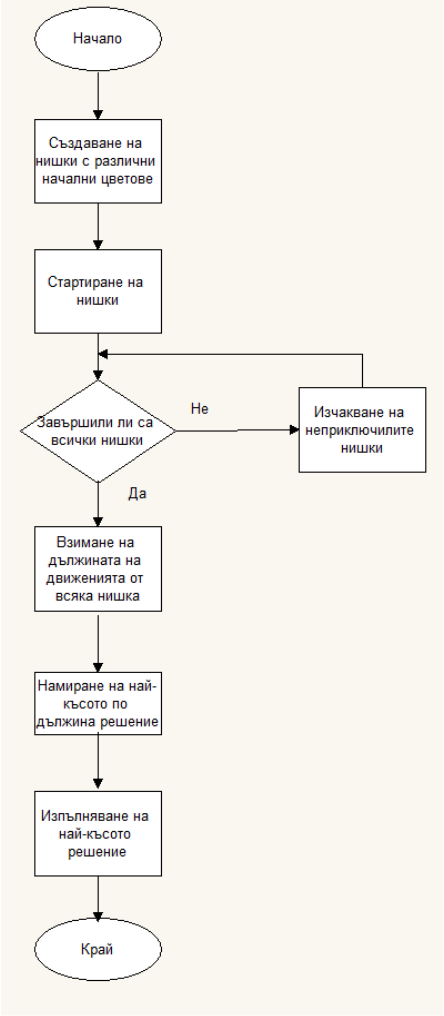
Последната стъпка е правилното ориентиране на жълтите ъгли. Възможно е да има нула, един, два или четири правилно ориентирани ъгъла. При вариант с четири кубът е вече нареден. Не може да имаме вариант с три правилно ориентирани ъгъла – това би означавало, че няма наредени предишни слоеве или кубът е бил сглобен неправилно. Правилното ориентиране на ъглите става чрез следния алгоритъм – RU'U'R'U'RU'R'L'UULUL'UL. Ако има нула ориентирани ъгъла, то алгоритъмът се изпълнява веднъж. След това се получава поне един ориентиран ъгъл. Без значение е дали има един или два ориентирани ъгъла, като поне един от тях трябва да е в горния ляв ъгъл срещу нас – това става чрез завъртане на кубчето.

Продължава се с изпълняването на алгоритъма, докато кубчето не бъде подредено (фигура 6).



Фиг. 6. Подреден Рубик Куб

За всеки цвят на кубчето се използва нишка, която започва да решава куба от различен начален цвят (фигура 7). Така се получава паралелно изпълнение на алгоритъма за подреждането, което от своя страна води и до понижаване на времето на намиране на най-краткото решение. След това решението се подава към Arduino микроконтролера и серията от движения се изпълнява върху физическия Рубик куб.



Фиг. 7. Работа на нишките

4. Заключение

Реализирането на системата се доближава до нареждането на куб от човек, като тя успешно разпознава страните на кубчето чрез заснемането на всяка от тях, изпраща ги към програмата и обработва този резултат. Средното време за решаване на задачата е 1.5 s, което го отличава сериозно от останалите системи за решаване на Куб на Рубик, повечето от които имат време за решаване от 3s до 9s [3]. Другите системи използват метод за подреждане на кубчето, който е различен от начинаещия и води до забавяне в решението, като още веднъж утвърждава предимството на реализирания в текущия алгоритъм начинаещ метод. Начинаещият метод има по-бързо време за решаване на задачата, но последователността от движения, нужни за подреждането на куба, е по-дълга.

Реализираният алгоритъм за подреждане на Рубик куб съчетава в себе си множество технологии като обработка на изображения, използване на нишки и роботизация. Алгоритъмът е оригинален, тъй като по-важната част от него, а именно началото му и направата на белия кръст, се постига интуитивно при нареждането на куба от човек, което прави прилагането ѝ в софтуерен алгоритъм по-трудно.

Литература

[1]. Ruwix, Beginner’s Method, https://ruwix.com/the-rubiks-cube/how-to-solve-the-rubiks-cube-beginners-method/, 2017
[2]. Wikipedia, Rubik’s Cube, https://en.wikipedia.org/wiki/Rubik%27s_Cube, 2007
[3]. Rubik’s Cube Solver, https://rubiks-cube-solver.com/, 2016

За контакти:
инж. Павлин Д. Щерев
катедра „Компютърни науки и технологии”
Технически университет - Варна
E-mail: pavlinshterev97@gmail.com

ПРОЕКТИРАНЕ НА РОБОТ ЗА ДВИЖЕНИЕ В СРЕДА С ПРЕПЯТСТВИЯ

Ивайло Ст. Георгиев

Резюме: Статията представя робот, който да може свободно да се движи в среда с препятствя. Роботът е ефикасен и евтин вариант, който да е достъпен до по-голям брой хора. Разработката е в опростен вид, но към нея могат да се добавят модули, които допълнително да я специализират в дадена насока. Конкурентна е с предлаганите на пазара подобни роботи за развлечение, показва стабилна работа и е лесна за поддръжка.
Ключови думи: Избягване на препятствия; робот; Arduino.

Design of a Robot Moving in an Environment with Obstacles

Ivaylo St. Georgiev

Abstract: The article presents a robot that can move freely in an obstacle environment. The robot is an efficient and inexpensive option that is accessible to a larger number of people. The development is in a simplified form, but modules can be added to it to further specialize it in a given direction. It is competitive with similar entertainment robots on the market, shows stable operation and is easy to maintain.
Keywords: Obstacle avoidance; robot; Arduino.

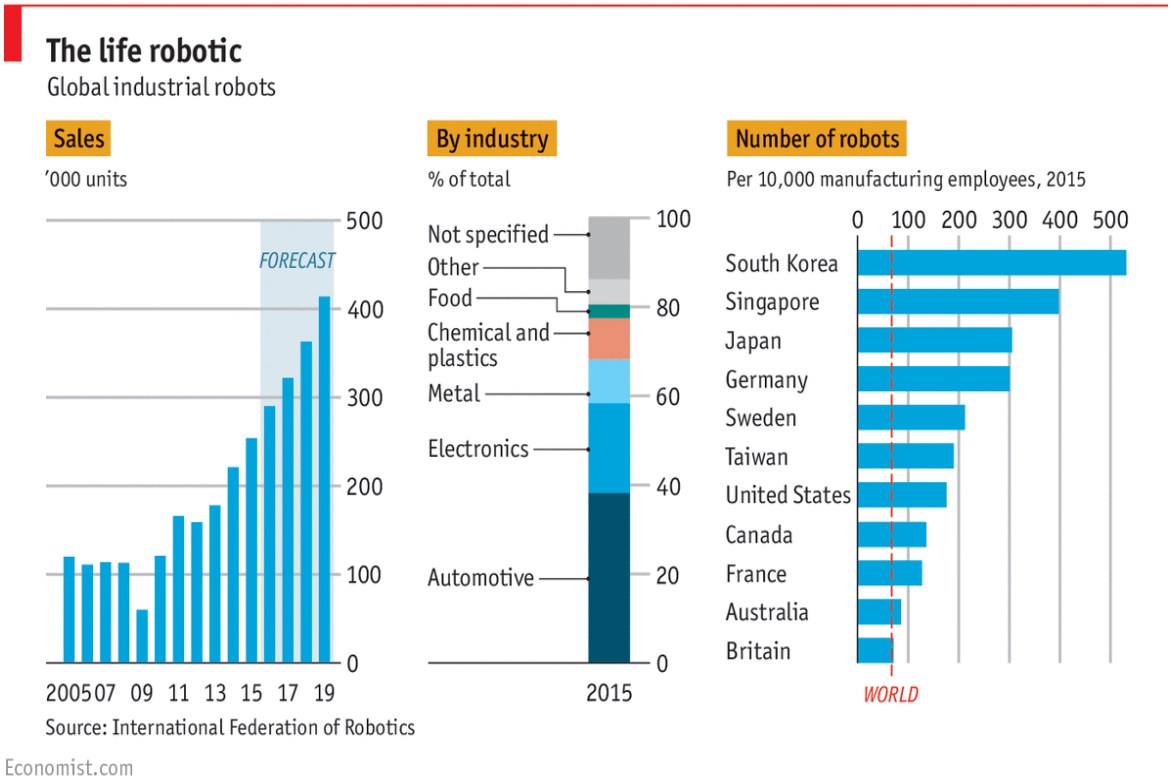
1. Въведение

С бързото развитие на роботиката и автоматиката във всички сфери на индустрията, включително в сектори, в които работата се извършва по един и същи начин от десетилетия, се създават много иновации. Те предоставят алтернативни методи, чрез които тази работа може да се ускори и по този начин да се увеличи продуктивността на предприятието. Автоматизирането на тази процоси се състои в разработването на роботи, които да изпълняват дадени задачи без външна намеса. По този начин се елеминира необходимостта от наемането на допълнителен персонал, като по този начин стойността на тези апаратни устройства се вдига многократно. Точността, производителността и техническите характеристика на хардуерните системи растат всеки ден. С всеки изминал ден технологиите се развиват все по-бързо и по-бързо. С техния бърз темп и стремежа им към бързина и надеждност, роботите, базирани на тези технологии, бързо намират място в живота ни.

В промишлената дейност те бързо намират своето място като незаменими за много предприятия, в които се стреми да се достигне максимална производителност и изпълнение на брой поръчки. Интернационалната федерация по роботика (IRF) прогнозира, че в идните години все повече и повече устройства ще биват предлагани на пазара.

Фигура 1 илюстрира графики, които са били публикувани в уебсайта за икономика - Икономист (Economist) през 2017 година. На тях са показани в какво количество и къде намират приложение роботите.

С увеличаването на достъпността на роботите до общото население се създава и нужда за роботи, които да изпълняват някои докамински задачи или да бъдат използваниь за просто развлечение. Все по-популярни стават и състезанията между роботи, които се конкурират по техника, прецизност или създаване и използване на по-добър алгоритъм. В домашната сфера един такъв добър пример, който увеличава комфорта на живот, се явява роботът прахосмукачка Румба - "Roomba" (фигура 2).



Фиг. 1. Глобално разпространените роботи в различните индустрии [7]

Фиг. 2. - Робот прахосмукачка Румба (Roomba)

Роботът, представен на фигура 2, изпълнява алгоритъм, който го оповестява, ако се блъсне в стена или в някакво друго препятствие, което е може би единственият му минус.

2. Изложение

2.1 Преглед на съществуващи решения

В този проект използваните средства са Arduino Uno, HC-SR04 Ultrasonic Sensor, L298N Motor Driver Module. Много хора биха избрали да използват продукти като Raspberry Pi, но за този проект, в който целим по-висока ефикасност на по-ниска цена, Arduino се явява по-добрият избор. Той има своя среда за програмиране, която е C базирана и удобна за използване дори и от любители.[1][2]

Arduino се състои от 8-битов Atmel AVR микроконтролер с допълващи се компоненти, които улесняват програмирането и включването в други вериги. Важен аспект на Arduino платформата е наличието на стандартни съединители, които позволяват на потребителите да свързват CPU платката към голям набор от различни, взаимозаменяеми модули, наречени разширения. Някои комуникират с Arduino директно, посредством различни съединители. Благодарение на I^2C шина, няколко разширения могат да бъдат прикачени и използвани паралелно.[3][4]

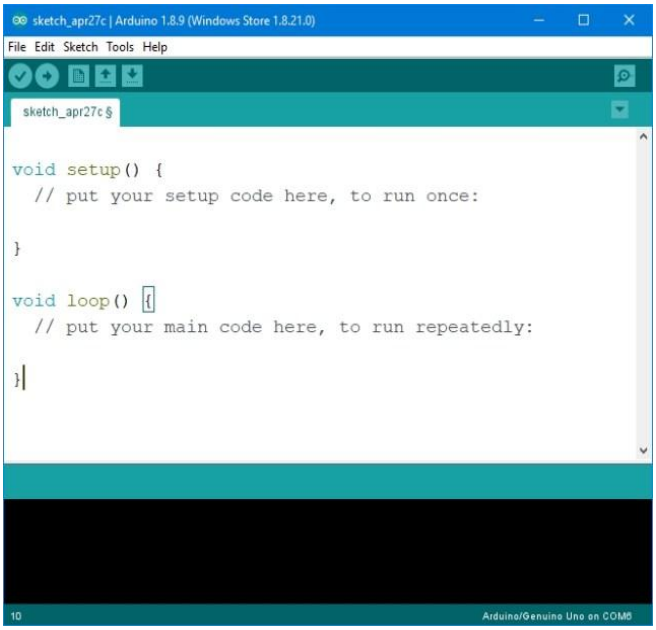
Таблица 1. Технически спецификации на Arduino Uno R3

Microcontroller	Microchip Atmega328P
Operating Voltage	5 Volts
Input Voltage (recommended)	7-12 Volts
Input Voltage (limit)	6-20 Volts
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by the bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

От таблицата можем да заключим, че ресурсите, които предлага Arduino, са пределно достатъчни. Подходящи са за проекти като този, в който се управлява хардуер, като датчици или други изпълнителни устройства [5]

За разработване на проекти, които използват микроконтролера на Arudino, се използва интегрираната среда Arduino IDE (фигура 3), която е платформа с отворен код.

На нея се създават различни проекти, които са свързани с електроника и библиотеките им използват функции, които са написани на езиците за програмиране C и C++.



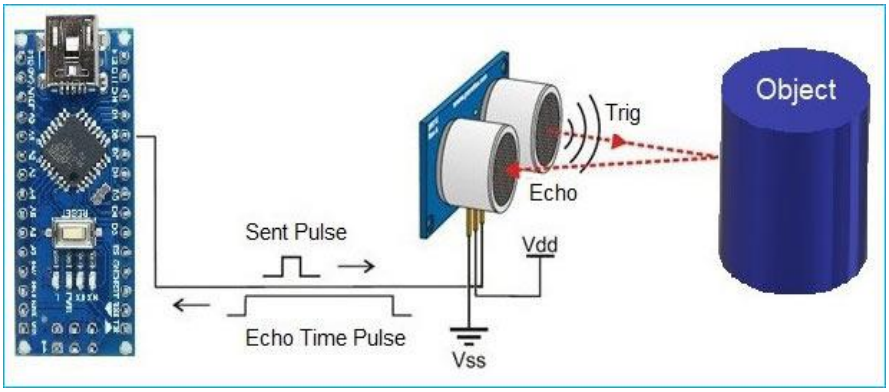
Фиг. 3. Arduino IDE – празен проект

За да се конвертира изпълнимият код в текстов файл в шестнадесетично кодиране, се използва програмата avrdude. Тя се зарежда в дъската Arduino от програмата за зареждане на фърмуера на платката. По подразбиране, тази програма се използва като инструмент за качване на потребителския код върху официалните платки на Arduino.

Едно от положителните качества на платката Arduino е това, че не се нуждае от допълнителен хардуер, за разлика от предхождащите я програмируеми платки, за да зарежда нов код в платката, тъй като свързането става с най-обикновен USB кабел.

За управлението на мотора се използва L298N Motor Driver Module [6], който получава командите си от дъската на Arduino и управлява в необходимите посоки стъпковите мотори, които се използват за задвижване на робота.

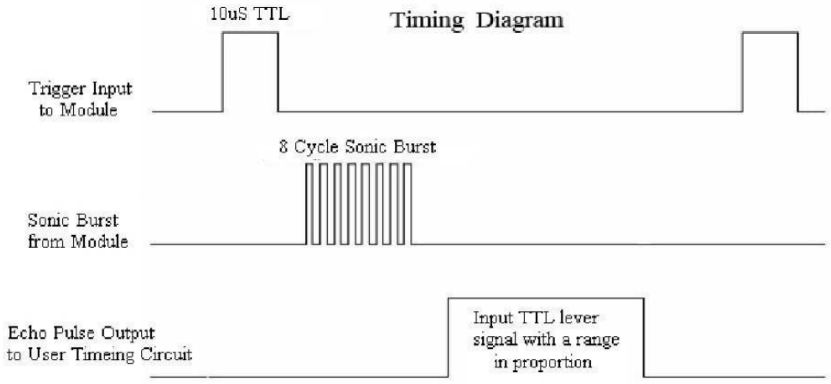
Един от най-подходящите датчици, който можем да използваме за навигиране в среда с препятствия, е HC-SR04 Ultrasonic Sensor [7], който е ултразвуков сензор. Той дава възможността на робота да предвижда препятствията и да предприеме нужните маневри, за да ги избегне.



Фиг. 4. Принцип на работа на ултразвуковия сензор

Необходимо е само да се подаде кратък импулс от 10uS към входа на спусъка, за да се стартира обхватът, и след това модулът ще изпрати 8-циклов взрив на ултразвук при 40 kHz и ще повиши своето ехо. Ехото е обект от разстояние, който е широчина на импулса и обхватът пропорционално. Може да се изчислите обхватът през интервала от време между изпращания задействащ сигнал и приемания ехо сигнал. (фигура 4)

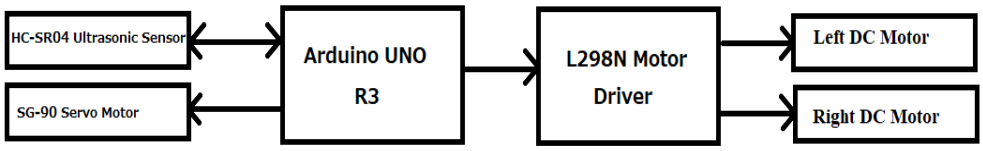
Формула: uS / 58 = сантиметри или uS / 148 = инч; или: диапазон = време на високо ниво * скорост (340M / S) / 2; препоръчително е да се използва цикъл на измерване над 60 ms, за да се предотврати задействащ сигнал към exo сигнала. Това се вижда по-ясно на времедиаграмата на фигура 5.



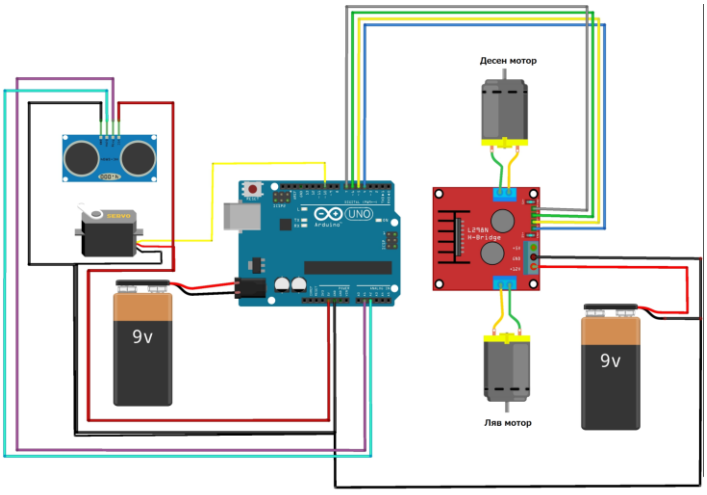
Фиг. 5. Времедиаграма на ултразвуковия сензор

2.2. Проектиране на системата

За да са реализира правилно прототипът, са необходими платка Arduino Uno, ултразвуков сензор HC-SR04, сервомотор SG-90, два постояннотокови мотора и L298N Motor Driver Module, който да управлява моторите. На фигура 6 е представена блок диаграма на прототипа и е показана каква е връзката между всеки елемент, който участва в неговата направа.



Фиг. 6. Блок-схема на прототипа



Фиг. 7. Цялостна схематична диаграма на прототипа

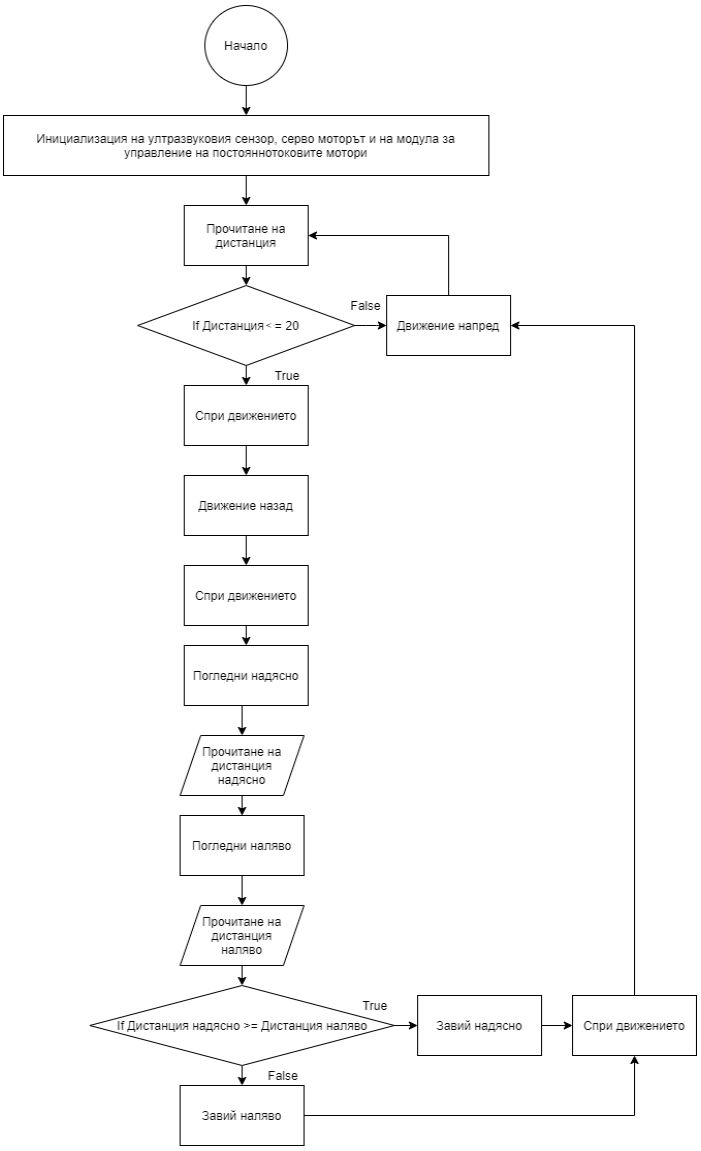
За да се създаде подходящ алгоритъм за прототипа, трябва да се изпълнят няколко последователни задачи:

- Трябва да се създаде функция за управление на моторите от драйвера.
- Необходимо е да се създаде функция, която да управлява сервомотора, който ще завърта сензора при необходимост.

- Да се създаде функция, която да управлява ултразвуковият сензор и да записва данните, които той изпраща.
- Да се създаде в циклична функция, която да използва данните от предходната функция и да управлява посоката на прототипа спрямо тях.

Последователността от действия, които роботът трябва да приеме, са (фигура 8):

1) Ултразвуковият сензор трябва да проверява постоянно дали в радиуса на обсега му има някакво препятствие, като получените резултати трябва да се изпращат и записват в ардуиното.
2) В случай че няма нищо пред него, роботът трябва да продължи да се движи напред, но ако има препятствиен трябва да спре и да се върне назад.
3) Следващата стъпка е да се задейства сервомоторът, който трябва да завърти ултразвуковия сензор, за да може той да се огледа какво има от неговата дясна и после лява страна, като се сравняват дистанциите, стига да има препятствия пред някоя от тях.
4) След като се сравнят резултатите от предходната стъпка, роботът трябва да предприеме действие и да завие в подходяща посока, след което всичко това се повтаря.



Фиг. 8. Блок-схема на последователността от действия, които предприема роботът

След нареждане на прототипа се пуска интеграционен тест, като в периода на няколко минути роботът се оставя да се движи. Той се счита за успешен, ако след протичането на предварително засеченото време нито веднъж не се е блъснал никъде, а е избегнал всяко възможно препятствие, което му се е изпречило по време на теста.

3. Заключение

В статия е предложено и практически реализирано решение на задачата да се създаде прототип на робот, който да може сам да избягва препятствия без външна намеса.

Проведените тестова с устройството еднозначно показват, че реализацията е успешна и че то може да се използва по предназначение.

Има няколко начина, по които може да се направи подобрение на робота:

Единият от тях е да се замени ултразвуковият сензор с инфрачервен сензор или със сензор за измерване на дълбочина (depth sensor), чрез които биха могли да се сравняват по по-различен начин дистанциите между препятствията. Интересното при инфрачервният сензор е, че той може да приема топлинни сигнали, което би могло да се използва, за да може прототип, направен с такъв сензор, да избягва по-горещи повърхности, които биха могли да разтопят колелетата му.

Друго евентуално подобрение е да се запази ултразвуковият сензор, но да се добави още един, за да се достигне по-широк обозр над терена, на който може да се използва такъв робот. По този начин може да се избегне използването на сервомотор и да се съкрати кодът, който се използва.

В заключение, създаденият прототип работи без никакви проблеми и извършва изискваните от него задачи.

Литература

[1]. Arduino for beginners - *"Arduino UNO for beginners - Projects, Programming and Parts"*.
[2]. “Arduino Software Release Notes". Arduino Project. Retrieved September 25, 2019.
[3]. “Arduino - FAQ". www.arduino.cc
[4]. https://www.oreilly.com/library/view/arduino-a-technical/9781491934319/ch04.html
[5]. Coley, Gerald (August 20, 2009). "Take advantage of open-source hardware"
[6]. Datasheet на L298N Driver Motor Module - https://components101.com/modules/l293n-motor-driver-module
[7]. Datasheet на HC-SR04 Ultrasonic Sensor - https://components101.com/ultrasonic-sensor-working-pinout-datasheet
[8]. https://www.economist.com/graphic-detail/2017/03/27/the-growth-of-industrial-robots

За контакти:
инж. Ивайло Ст. Георгиев
катедра „Компютърни науки и технологии”
Технически университет - Варна
E-mail: ivo.georgiev1997@gmail.com

ИЗИСКВАНИЯ ЗА ОФОРМЯНЕ НА СТАТИИТЕ ЗА СПИСАНИЕ "КОМПЮТЪРНИ НАУКИ И ТЕХНОЛОГИИ”

I. Статиите се представят разпечатани в два екземпляра (оригинал и копие) в размер до 6 страници, формат А4 на адрес: Технически университет – Варна, ФИТА, ул. „Студентска” 1, 9010 Варна, както и в електронен вид на имейл адреси: ned.nikolov@tu-varna.bg или yulka.petkova@tu-varna.bg.

II. Текстът на статията трябва да включва: УВОД (поставяне на задачата), ИЗЛОЖЕНИЕ (изпълнение на задачата), ЗАКЛЮЧЕНИЕ (получени резултати), БЛАГОДАРНОСТИ към сътрудниците, които не са съавтори на ръкописа (ако има такива), ЛИТЕРАТУРА и информация за контакти, включваща: научно звание и степен, име, организация, поделение (катедра), e-mail адрес.

III. Всички математически формули трябва да са написани ясно и четливо (препоръчва се използуване на Microsoft Equation).

IV. Текстът трябва да бъде въведен във файл във формат WinWord 2000/2003 с шрифт Times New Roman. Форматирането трябва да бъде както следва:

1. Размер на листа - А4, полета: ляво - 20мм, дясно - 20мм, горно - 15мм, долно - 35мм, Header 12.5мм, Footer 12.5мм (1.25см).
2. Заглавие на български език - размер на шрифта 16, удебелен, главни букви.
3. Един празен ред - размер на шрифта 14, нормален.
4. Имена на авторите - име, инициали на презиме, фамилия, без звания и научни степени - размер на шрифта 14, нормален.
5. Два празни реда - размер на шрифта 14, нормален.
6. Резюме и ключови думи на български език, до 8 реда - размер на шрифта 11, нормален.
7. Заглавие на английски език - размер на шрифта 12, удебелен.
8. Един празен ред - размер на шрифта 11, нормален.
9. Имена на авторите на английски език - размер на шрифта 11, нормален.
10. Един празен ред - размер на шрифта 11, нормален.
11. Резюме и ключови думи на английски език, до 8 реда - размер на шрифта 11, нормален.
12. Основните раздели на статията (Увод, Изложение, Заключение, Благодарности, Литература) се форматират в едноколонен текст както следва:
    a. Наименование на раздел или на подраздел - размер на шрифта 12, удебелен, центриран, един празен ред преди наименованието и един празен ред след него - размер на шрифта 12, нормален;
    b. Текст - размер на шрифта 12, нормален, отстъп на първи ред на параграф – 10 мм; разстояние от параграф до съседните (Before и After) за целия текст – 0.
    c. Цитиране на литературен източник - номер на източника от списъка в квадратни скоби;
    d. Текстът на формулите се позиционира в средата на реда. Номерация на формулите - дясно подравнена, в кръгли скоби.
    e. Фигури - центрирани, разположение спрямо текста: “Layout: In line with text”. Номер и наименование на фигурата - размер на шрифта 11, нормален, центриран. Отстояние от съседните параграфи – 6 pt.
    f. Литература – всеки литературен източник се представя с: номер в квадратни скоби и точка, списък на авторите (първият автор започва с фамилия, останалите – с име), заглавие, издателство, град, година на издаване, страници.
    g. За контакти: научно звание и степен, име, презиме (инициали), фамилия, организация, поделение (катедра), e-mail адрес, с шрифт 11, дясно подравнено.

Образец за форматиране можете да изтеглите от адрес http://cs.tu-varna.bg/ - Списание КНТ, Spisanie_Obrazec.zip.

Този брой от списанието се отпечатва със средства от научно-изследователски проект, финансиран целево от държавния бюджет.