МИНОБРНАУКИ РОССИИ

Федеральный исследовательский центр «Информатика и управление» РАН
Национальный комитет при президиуме РАН по распознаванию образов и анализу изображений
Институт информационных технологий Национальной академии наук Азербайджана
Институт проблем передачи информации им. А.А. Харкевича РАН
Издательство «Наука и технологии»
Национальный исследовательский Томский государственный университет
Федеральный исследовательский центр «Карельский научный центр РАН»
Федеральное государственное бюджетное образовательное учреждение высшего образования «Юго-Западный государственный университет»

ОПТИКО-ЭЛЕКТРОННЫЕ ПРИБОРЫ И УСТРОЙСТВА В СИСТЕМАХ РАСПОЗНАВАНИЯ ОБРАЗОВ И ОБРАБОТКИ ИЗОБРАЖЕНИЙ

Распознавание – 2021

Сборник материалов XVI Международной научно-технической конференции

14–17 сентября 2021 года

Курск 2021

УДК 621.383.68.3: 681.785
ББК В 338.4
О 66

О 66 **Оптико-электронные приборы и устройства в системах распознавания образов и обработки изображений. Распознавание – 2021**: сб. материалов XVI Междунар. науч.-техн. конф. / ред. кол.: С. Г. Емельянов, В. С. Титов (отв. ред.) [и др.]; Юго-Зап. гос. ун-т. – Курск, 2021. – 284 с.

ISBN 978-5-7681-1520-3

Сборник содержит материалы XVI Международной научно-технической конференции «Оптико-электронные приборы и устройства в системах распознавания образов и обработки изображений» (Курск, 14–17 сентября 2021 г.), целью которой является ознакомление с имеющимися достижениями по созданию оптико-электронных приборов, систем и внедрение информационных технологий в научные исследования, учебный процесс и промышленность, а также координация по эффективному их применению в системах распознавания образов и обработки изображений.

Сборник предназначен для научных сотрудников, преподавателей, аспирантов и студентов вузов.

Издание осуществлено с авторских оригиналов. Редакция не несет ответственности за ошибки авторов.

Материалы для публикации одобрены программным комитетом Международной научно-технической конференции.

УДК 621.383.68.3: 681.785
ББК В 338.4

ISBN 978-5-7681-1520-3

СОДЕРЖАНИЕ

each convolution layer of the model: $U_i = \mathrm{Re}lu(S_i) \triangleq \max(0, S_i)$. Maxpooling was used in the model as a combination operation: $pool(U_i) := \max_{l=1}^{p} U_{l+(i-1)e}, \quad \forall i = 1, 2, \ldots, \frac{k-m}{d} + 1$, where p is the size of the puling, e is the stride size. Since the classification problem is solved in the article, softmax was used as an output activation function (sigmoid can be used in regression problem). Thus, for the classification problem, the softmax is expressed as follows:

$$\hat{y} = \frac{e^{O_n}}{\sum_{j=1}^{N} e^{O}}, \quad n = 1, 2, \ldots, N.$$

The cross-entropy function was used to minimize the difference between the predicted values and the actual values in the training data and is defined as follows:

$$L_{\text{crossentropy}} = -\frac{1}{q}\sum_{i=1}^{q}\sum_{n=1}^{N} 1\{y_i = n\} \log \hat{y}_i + \left(1 - 1\{y_i = n\}\right) \log\left(1 - \hat{y}_i\right),$$

where y_i is the output value of the i-th training sample and q is the total number of training samples. In the equation, $1\{y_i = n\}$ is a logical expression that always returns a value of 0 or 1. After selecting the loss function, the Adam Optimization function was used to train the parameters to update the weights. Here, CNN updates the weights until the model achieves the least loss at the predefined maximum iteration. The CSIC2010 and CSE-CIC-IDS2018 datasets were used for the experiments. Here, the effectiveness of the method was evaluated on the basis of Accuracy, Precision, recall and f1-score metrics, and the results are included in table.

Evaluation CNN efficiency on CSE-CIC-IDS2018 DDoS and CSIC 2010 datasets

	Class	Accuracy	Precision	Recall	F1-score
CSE-CIC-IDS2018 DDoS dataset	Benign (0)	0.9853	0.9923	0.9921	0.9921
	DDoS (1)	0.9999	0.9999	0.9999	0.9999
	DoS (2)	0.9931	0.9903	0.9901	0.9944
CSIC 2010 dataset	Anomalous Traffic (0)	0.6331	0.9201	0.6311	0.7531
	Normal Traffic (1)	0.9911	0.9121	0.9924	0.9521

As can be seen from Table 1, the model was able to detect Anomalous Traffic (0) class data with low efficiency when tested on CSIC 2010 dataset data. Thus, the model over the accuracy, precision, recall, and F1-score metrics achieved 0.63, 0.92, 0.63, 0.75 values respectively. In the recognition of Normal Traffic (1)

data, the model showed good results and achieved 0.99, 0.91, 0.99, 0.95 for these metrics, respectively. When the model was tested on the CSE-CIC-IDS2018 DDoS dataset, high results were obtained for almost all metrics.

This work was supported by the Science Development Foundation under the President of the Republic of Azerbaijan – Grant No. EIF-BGM-4-RFTF-1/2017-21/08/1.

1. Singh K. J., De T. Analysis of Application Layer DDoS Attack Detection Parameters Using Statistical Classifiers // Internetworking Indonesia. 2017. N 9(2). P. 23–31.

2. Liao Q., Li H., Kang S., Liu C. Feature extraction and construction of application layer DDoS attack based on user behavior / 33rd Chinese Control Conference (CCC), 2014. P. 5492–5497.

UDC 004.052.42

T. H. Kazimov[1], T. A. Bayramova[1]
e-mail: toma_b66@mail.ru
[1] Institute of Information Technology of ANAS, Baku, Azerbaijan

ABOUT A METHOD FOR EVALUATING THE DEGREE OF SOFTWARE COMPLEXITY INTRODUCTION

Application of program code metrics allows professionals who work on the project to evaluate various feauters of existing or to be created software, to predict the scope of work, quantitatively characterize these or other project solutions, to evaluate quality of prepared systems, complexity and realiability of software.

Problem setting. Different metrics (especially quantities) that are used when evaluating the efforts and efforts of the program employee are of a recommendation nature. Because some workers deliberately reduce or inflate these indicators. Therefore, an assessment of the complexity of the task facing an employee can play a significant role in solving this problem. For example, it may take days, weeks, and sometimes months, to find a bug in any complex program, but its correction results in a change of one line of program code [1, 2].

At work, a different metric is used to evaluate the complexity of software. This metric allows you to more effectively evaluate complexity while maintaining the basic features of Halstead, Mccabe and Jilb metrics [3, 4].

To evaluate complexity coefficient - CD of software, let's take following labels:

- AC – the total number of operators in the software module;
- CC – a parameter based on the number of cycle operators contained in the program module;
- FC– a parameter based on the number of conditional operators contained in the program module;
- CP– number of applications in standard and ready-to-use programs contained in the software module.

Solution of the problem. Complexity coefficient of software – CD is calculated as follow:

$$CD=(CC+FC+CP)/AC. \quad (1)$$

$CD\in[0;1]$ –and its greater values is considered to be better.

The CCparameter, based on the number of cycle operators in the program module, is calculated by the following formula:

$$CC=\sum_{i=1}^{n}(i+1)CC_i,\ (i=0,1,...,n), \quad (2)$$

whereiCC_i is the number of cycle operators with the number i cycle operator inside. And $(n=0,1,2,3,...)$.

The FCparameter based on the number of conventional operators in the program module is calculated using the following formula:

$$FC=\sum_{i=1}^{n}(i+1)FC_i,\ (i=0,1,...,n), \quad (3)$$

where FC_i is the number of conventional operators with the i number conventional operator. And $(n=1,2,3,...)$.

Conclusion

Numerous studies in the field of software complexity metrics allow us to say that there is no universal metric to evaluate the complexity of any program code. The use of any metric or hybrid metric, or several metrics, depends on a particular issue. But the proposed metric allows for a more accurate (at the expense of additional indicators) complexity of software modules of a large class, separately from known metrics. This, in turn, will improve the quality, reliability of existing systems and so on. to evaluate.

REFERENCES

1. Alenezi M., Zarour M. On the relationship between software complexity and security // International Journal of Software Engineering & Applications (IJSEA). 2020. Vol.11, No.1. P. 51–60.

2. Mohd. Kamran Khan et al. Literature review on software complexity, software Usability and software deliverability International Journal of Advanced Research in Computer Science. 2018. 9 (2). P. 438–441.

3. Hariprasad T., Vidhyagaran G., Seenu K., Thirumalai C. Software complexity analysis using halstead metrics // International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli. 2017. P. 1109–1113.

4. Ali Athar Khan, Amjad M., Sajeda M. Amralla, Tahera H. Mirza Comparison of Software Complexity Metrics // International Journal of Computing and Network Technology. 2016. No. 1. P. 19–26.

UDC 519.711

***Sh. J. Mahmudova*[1]**
e-mail: shafagat_57@mail.ru
[1] *Institute of Information Technology of ANAS, Baku, Azerbaijan*

THE APPLICATION AREAS OF INTELLIGENT SYSTEMS

The paper explores the field of intelligent systems. The structure, types, application and classification of the intelligent system are studied. The intelligent system is a technical or software system that is capable of solving the problems which are specific and creative in a particular subject area, so that knowledge is stored in the memory of such systems.

Intelligent Information System (IIS) is a set of software, language and logic tools to support the human activities for the key task implementation and to search the information in the natural language in a dialogue mode.

Intelligent systems in decision-making technologies are intelligent information-computing systems that are capable to solve the problems without human involvement [1].

The structure of the intelligent system includes three main units:

- Knowledge base.
- Solution acquisition mechanism.
- Intelligent interface.

The following aspects are used to ensure the functioning of the intelligent system:

- Mathematical;
- Linguistic;