НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
НАВЧАЛЬНО-НАУКОВИЙ
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

# СИСТЕМНІ ДОСЛІДЖЕННЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

МІЖНАРОДНИЙ НАУКОВО–ТЕХНІЧНИЙ ЖУРНАЛ

№ 4 2025

ЗАСНОВАНО У ЛИПНІ 2001 р.



**АДРЕСА РЕДАКЦІЇ:**
03056, м. Київ,
просп. Берестейський, 37, корп. 35,
НН ІПСА КПІ ім. Ігоря Сікорського
Тел.: **204-81-44**; факс: **204-81-44**
E-mail: journal.iasa@gmail.com
http://journal.iasa.kpi.ua

## У номері:

NATIONAL ACADEMY OF SCIENCES OF UKRAINE
EDUCATIONAL AND RESEARCH
INSTITUTE FOR APPLIED SYSTEM ANALYSIS
OF THE NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
«IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE»

# SYSTEM RESEARCH AND INFORMATION TECHNOLOGIES

INTERNATIONAL SCIENTIFIC AND TECHNICAL JOURNAL

№ 4 2025

IT IS FOUNDED IN JULY 2001



**THE EDITION ADDRESS:**

03056, Kyiv,
av. Beresteiskyi, 37, building 35,
ER Institute for Applied System Analysis
at the Igor Sikorsky Kyiv Polytechnic Institute
Phone: **204-81-44**; Fax: **204-81-44**
E-mail: journal.iasa@gmail.com
http://journal.iasa.kpi.ua

## In the issue:

## Шановні читачі!

Навчально-науковий інститут прикладного системного аналізу Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» видає міжнародний науково-технічний журнал

**«СИСТЕМНІ ДОСЛІДЖЕННЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ».**

Журнал публікує праці теоретичного та прикладного характеру в широкому спектрі проблем, що стосуються системних досліджень та інформаційних технологій.

**Провідні тематичні розділи журналу:**

Теоретичні та прикладні проблеми і методи системного аналізу; теоретичні та прикладні проблеми інформатики; автоматизовані системи управління; прогресивні інформаційні технології, високопродуктивні комп'ютерні системи; проблеми прийняття рішень і управління в економічних, технічних, екологічних і соціальних системах; теоретичні та прикладні проблеми інтелектуальних систем підтримання прийняття рішень; проблемно і функціонально орієнтовані комп'ютерні системи та мережі; методи оптимізації, оптимальне управління і теорія ігор; математичні методи, моделі, проблеми і технології дослідження складних систем; методи аналізу та управління системами в умовах ризику і невизначеності; евристичні методи та алгоритми в системному аналізі та управлінні; нові методи в системному аналізі, інформатиці та теорії прийняття рішень; науково-методичні проблеми в освіті.

**Головний редактор журналу —** завідувач кафедри системного проєктування НН ІПСА КПІ ім. Ігоря Сікорського, професор Мухін Вадим Євгенович.

Журнал «Системні дослідження та інформаційні технології» включено до переліку наукових фахових видань України (категорія «А»).

Журнал «Системні дослідження та інформаційні технології» входить до таких наукометричних баз даних: Scopus, EBSCO, Google Scholar, DOAJ, Index Copernicus, реферативна база даних «Україніка наукова», український реферативний журнал «Джерело», наукова періодика України.

Статті публікуються українською та англійською мовами.

Якщо ви не встигли передплатити журнал, його можна придбати безпосередньо в редакції за адресою: 03056, м. Київ, просп. Берестейський, 37, корп. 35.

Завідувачка редакції **С.М. Шевченко**
Редакторка **Р.М. Шульженко**
Молодша редакторка **Л.О. Тарин**
Комп'ютерна верстка, дизайн **А.А. Патіохи**

Рішення Національної ради України з питань телебачення і радіомовлення
№1794 від 21.12.2023. Ідентифікатор медіа R30-02404

Підписано до друку 29.12.2025. Формат 70×108 1/16. Папір офс. Гарнітура Times.
Спосіб друку – цифровий. Ум. друк. арк. 14,411. Обл.-вид. арк. 28,56. Наклад 150 пр. Зам. № 11/04

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Свідоцтво про державну реєстрацію: ДК № 5354 від 25.05.2017 р.
просп. Берестейський, 37, м. Київ, 03056.
ФОП Пилипенко Н.М., вул. Мічуріна, б. 2/7, м. Київ, 01014. тел. (044) 361 78 68.
Виписка з Єдиного державного реєстру № 2 070 000 0000 0214697 від 17.05.2019 р.

## Dear Readers!

Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" is published of the international scientific and technical journal

## "SYSTEM RESEARCH AND INFORMATION TECHNOLOGIES".

The Journal is printing works of a theoretical and applied character on a wide spectrum of problems, connected with system researches and information technologies.

**The main thematic sections of the Journal are the following:**

Theoretical and applied problems and methods of system analysis; theoretical and applied problems of computer science; automated control systems; progressive information technologies, high-efficiency computer systems; decision making and control in economic, technical, ecological and social systems; theoretical and applied problems of intellectual systems for decision making support; problem- and function-oriented computer systems and networks; methods of optimization, optimum control and theory of games; mathematical methods, models, problems and technologies for complex systems research; methods of system analysis and control in conditions of risk and uncertainty; heuristic methods and algorithms in system analysis and control; new methods in system analysis, computer science and theory of decision making; scientific and methodical problems in education.

**The editor-in-chief of the Journal** is Head of the Department of Systems Design of ER IASA, Igor Sikorsky Kyiv Polytechnic Institute, professor Vadym Mukhin.

The articles to be published in the Journal in Ukrainian and English languages are accepted. Information printed in the Journal is included in the Catalogue of periodicals of Ukraine.

# СИСТЕМНІ ДОСЛІДЖЕННЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

4 • 2025

## ЗМІСТ

# SYSTEM RESEARCH AND INFORMATION TECHNOLOGIES

4 • 2025

## CONTENT

ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ ПРОБЛЕМИ ІНФОРМАТИКИ

UDC 621.535
DOI: 10.20535/SRIT.2308-8893.2025.4.01

# DEVELOPMENT OF ALGORITHMS FOR DETECTING DEFECTS IN THE CODE SEQUENCE STRUCTURE ON THE SURFACE OF MODULATION DISKS

**D.Yu. MANKO, IE.V. BELIAK, A.A. KRYUCHYN, R.M. ISHCHENKO, V.V. ZAVARZINA**

**Abstract.** This study investigates algorithms for detecting and localizing defects in code sequence structures on modulation disk surfaces. It targets small anomalies in lithographically patterned elements that can cause readout errors or reduced measurement accuracy. A multi-level image-processing model combines Gaussian smoothing, adaptive thresholding, morphological operations, and contour-based segmentation. Processing stages are formalized as mathematical operators for reproducible implementation. Defects are characterized using perimeter- and area-based metrics, and their area distribution is approximated by a normal law. A spatial model computes defect centroids, enabling comparative quality assessment of disk samples. The software provides an interface for tuning thresholds, visualizing contours and defect-area plots, and exporting results. Tests on real defective disks confirm the method's reliable detection of local structural violations and its suitability for diagnostic systems.

**Keywords:** modulation disks, automated inspection, code sequence, microstructural anomalies, image preprocessing, morphological analysis, contour segmentation.

## INTRODUCTION

The integration of automated surface inspection methods into the technological workflow of optical and micromechanical components, particularly modulation disks, plays a crucial role in ensuring the accuracy and reliability of photoelectric measurement systems [1–3]. Previous studies have reported that the formation of high-precision coded structures on transparent substrates using photolithographic techniques is often accompanied by the emergence of local defects. These defects may result from technological inaccuracies, residual stresses, or surface contamination [4–6]. In response to the growing demands placed on the metrological performance of encoding systems, the development of effective technical diagnostic procedures for the detection of defects within code sequences at submicron structural resolution has become increasingly relevant.

Traditional inspection methods based on visual assessment and manual surface marking of modulation disks are significantly outperformed by modern approaches utilizing computer vision systems (Fig. 1). These advanced systems enable automated processing of digital images, integration with production lines,

and real-time adaptation to new requirements through the implementation of neural network algorithms [7]. The development of corresponding algorithms for the structured analysis of modulation disks is considered a ***priority direction*** in advancing the technology of high-precision optomechanical component fabrication and verification.

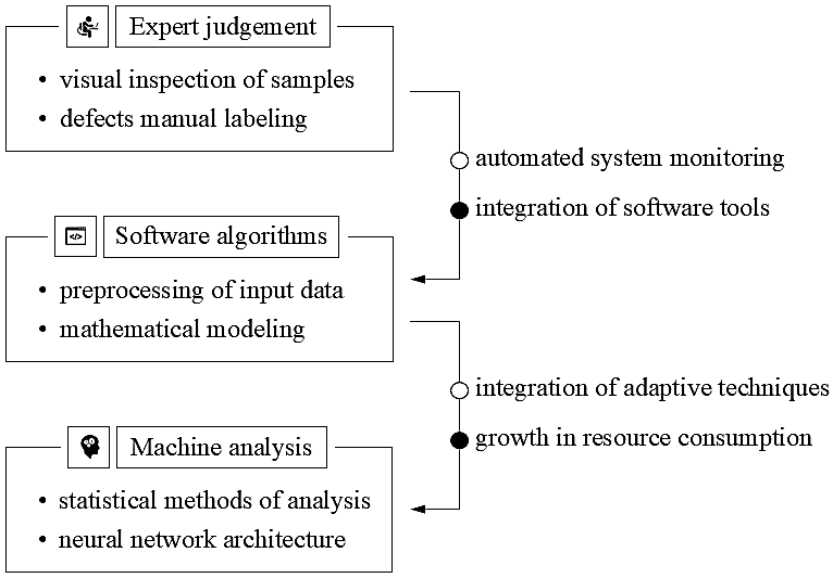



*Fig. 1*. Evolution of automated inspection tools for code sequences on modulation disks

**An analysis of scientific studies** devoted to the automation of defect detection in binary structures formed during the photolithographic deposition of code sequences reveals the active development of two principal approaches: classical algorithmic solutions [7–10] and machine learning-based methods [7; 11–14]. The first category focuses primarily on traditional image preprocessing techniques, including filtering, adaptive thresholding, segmentation, and morphological transformations [8–10]. These methods allow for both the restoration of the digital image matrix and the basic detection of structural anomalies. However, such approaches exhibit limited adaptability to changes in illumination, local distortions, and micro-scale defects, which are common in coded patterns produced by photolithographic processes. The second category of research emphasizes the use of neural network architectures, particularly convolutional neural networks (CNNs), autoencoders, and transformer-based models [11–14], which offer superior classification accuracy and enhanced generalization in the presence of incomplete input data and high noise levels. Nevertheless, the implementation of these solutions imposes substantial computational demands, often requiring graphics processing units (GPUs) or tensor accelerators, which complicates their deployment in software systems operating in real-time environments [11–14]. Furthermore, training neural network models necessitates the preparation of large datasets of annotated digital images with labeled defects, which may be infeasible in production settings with a limited number of representative samples. These considerations highlight the need for a comprehensive methodology that combines the efficiency of classical image processing algorithms, the flexibility of machine learning techniques, and the optimization of computational resources. Such an approach should aim to strike a balance between defect detection accuracy, processing speed, and adaptability to real-world industrial operating conditions.

**The aim of this study** is to develop a mathematically grounded approach for detecting defects in the structure of code sequences on the surface of modulation disks by integrating image preprocessing techniques, morphological analysis, and statistical interpretation of the results. The primary focus is placed on constructing a comprehensive methodology based on thresholding and contour analysis, employing adaptive filters, shape moments, and area distribution approximation of the detected defects. Given the constraints of computational resources and the need for integration with embedded control systems, the study does not explore the broad application of resource-intensive neural network models. Instead, it proposes an efficient software-based algorithmic solution that prioritizes detection accuracy, processing speed, and feasibility for deployment in industrial environments. The proposed model is designed to ensure the identification of local structural anomalies within code sequences, with the potential for future enhancements tailored to the specific characteristics of high-precision optomechanical components.

## PROBLEM STATEMENT: DEFECT DETECTION IN THE BINARY STRUCTURE OF A CODE SEQUENCE

The present study addresses the task of automatic defect detection in a binary structure formed on the surface of a modulation disk as a result of photolithographic reproduction of a code sequence. The corresponding structure is composed of a periodic or quasi-periodic set of elements, which are read by optoelectronic sensors with high spatial resolution [4–6]. The occurrence of defects in such structures—such as geometric distortions, fragmented damage, local darkening, or bright artifacts—can lead to positioning errors, signal readout failures, and degradation of the specified level of metrological accuracy.

The defect detection task is formalized as a process of digital image analysis, where the code sequence is represented as a binary mask corresponding to a two-dimensional matrix $\mathbf{BM}(x, y) \in \{0;1\}$, which contains pixel values obtained after thresholding the input data. The input dataset, in turn, is defined as a grayscale image matrix $\mathbf{GI}(x, y) \in [0;255]$. The objective of the software algorithm is to localize and classify regions that potentially deviate from the expected geometry of the binary structure. To achieve this, a sequence of filtering and morphological operations is applied, resulting in a set of contours $\{C_n\}$, where each $n \in [1; N]$ denotes a distinct object in the input image matrix, indicating a possible defect in the binary sequence structure. For each contour $C_n$, the corresponding area $S_n$ and perimeter $P_n$ are calculated based on the number of points forming the contour. A contour $C_n$ is classified as defective if its geometric parameters fall outside the empirically or calibration-defined thresholds: $S_{\min}$, $S_{\max}$, $P_{\min}$ and $P_{\max}$, which are set according to the objectives of the inspection system (see Fig. 2). Thus, the problem of defect detection in a code sequence structure is reduced to the construction of a computational procedure capable of reliably localizing anomalous regions based on geometric features of contours formed through morphological image processing

The selected approach avoids the use of complex machine learning models by implementing a software algorithm with controllable parameters, which can be

adapted to the limited computational resources of the hardware platform within the automated inspection system.

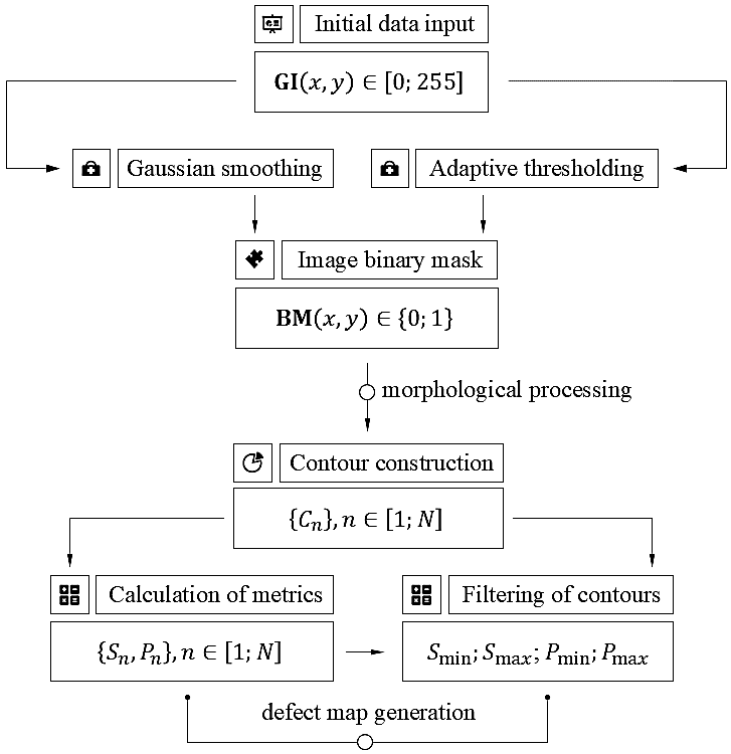



*Fig. 2*. Algorithmic flowchart for processing the digital image matrix for defect detection

## MATHEMATICAL MODEL FOR DEFECT RECOGNITION IN THE CODE SEQUENCE STRUCTURE ON THE SURFACE OF MODULATION DISKS

The formalization of the defect recognition procedure within the code sequence structure on the surface of a modulation disk is based on the development of a mathematical model comprising the stages of digital image preprocessing, morphological filtering, geometric contour analysis, and statistical evaluation of the parameters of the detected objects. The proposed model describes image transformations as a sequence of operations applied to the input image matrix and the resulting binary mask, thereby ensuring algorithmic modularity, reproducibility of results, and adaptability to specific application requirements.

At the first stage, the digital image matrix is converted into grayscale format, which reduces computational costs and enables processing based on the brightness values of each element $GI(x,y) \in [0;255]$. To reduce the negative impact of high-frequency noise and eliminate digital artifacts that may be mistakenly identified as defects, a Gaussian smoothing procedure is applied. The Gaussian smoothing method is based on the convolution of the image matrix with a two-dimensional kernel $G$, which is mathematically formalized as:

$$GI_G(x,y) = (GI * G)(x,y), \quad \text{де} \quad G(i,j) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{i^2+j^2}{2\sigma^2}\right).$$

The parameters of the two-dimensional Gaussian kernel were selected to ensure a balance between background smoothing and the preservation of image components. Following the smoothing stage, adaptive thresholding is applied to convert the image into binary form while accounting for local variations in illumination. Each pixel $GI_G(x,y)$ is mapped to a corresponding value in the binary image $BM(x,y)$ based on the average brightness $\mu_{GI}(x,y)$ within a local neighborhood of size $\Delta x \times \Delta y$, which was set to $11\times11$ pixels in this study, while the threshold offset parameter $\Delta\mu$ was determined empirically and adjusted using an interactive control element:

$$\begin{cases} BM(x,y)=1 \text{ при } GI_G(x,y)>\mu_{GI}(x,y)-\Delta\mu, \\ BM(x,y)=0 \text{ при } GI_G(x,y)\le\mu_{GI}(x,y)-\Delta\mu. \end{cases}$$

In the software implementation, inverse thresholding was applied, meaning that the binary mask is interpreted with reversed polarity and is formalized as $BM(x,y)=1$ when $GI_G(x,y)<\mu_{GI}(x,y)-\Delta\mu$. As a result of the aforementioned transformations, a binary mask $BM(x,y)\in\{0;1\}$ is obtained, in which potentially defective regions are highlighted as connected components with high contrast relative to the background. This stage is critically important for ensuring the clear formation of contours in the subsequent steps of morphological analysis of the image matrix.

After adaptive thresholding is applied, the binary mask matrix may contain residual noise, small-size artifacts, and structural distortions in the components of visual objects. To improve the quality of defect detection, classical morphological operations are used, allowing for the restoration of object shapes within the binary image matrix and the stabilization of the subsequent contour analysis stage. The fundamental morphological operation in this context is the morphological closing operation (MCO), which is implemented by sequentially performing dilation and erosion procedures on the binary mask matrix. The application of the closing operation to the binary mask $BM(x,y)$ is mathematically formalized as:

$$BM_{MCO}(x,y)=(BM\oplus MK)\ominus MK.$$

where $MK$ is the structural element that defines the shape and size of the morphological window (Morphological Kernel, MK). In the software implementation used in this study, a $5\times5$ pixel kernel was applied, with all elements set to $MK(x,y)=1$. The closing operation enables the suppression of digital artifacts that cause internal holes, contour breaks, and distortions in the overall shape of visual objects. This is followed by the application of the morphological opening operation, which is performed in reverse sequence:

$$BM_{MOO}(x,y)=(BM\ominus MK)\oplus MK.$$

The opening operation, in turn, is intended to remove small-size artifacts from the image that do not correspond to actual visual objects, eliminate isolated noise, and preserve the core geometry of larger objects. Thus, the sequential application of closing and opening operations enables the formation of a refined binary mask in which local defects have clearly defined boundaries without internal breaks or extraneous artifacts. This is critically important for the accurate extraction of contours in the subsequent stage. The structural element of the kernel $MK$ plays a key role in the quality of the restored image matrix. The selected rectangular kernel of $5\times5$ pixels ensures symmetric filtering of digital artifacts and thus

enables proper processing of both horizontal and vertical components. If necessary, the shape and size of the structural element can be adapted according to the specific characteristics of the defects.

After the morphological processing of the image, a refined binary mask is formed, reflecting potential regions with structural anomalies. The next step is the contour detection procedure (CDP), which identifies closed sequences of pixels that define the boundaries of connected components. Each contour is treated as a separate object that may correspond to a local defect. For each detected contour $\{C_n\}$, containing $n \in [1; N]$ points, the perimeter $P_n$ and area $S_n$ are calculated, serving as the fundamental geometric features:

$$\begin{cases} S_n = \frac{1}{2}\left|\sum_{m=1}^{M_n}(x_m\, y_{m+1} - x_{m+1}\, y_m)\right|, \\ P_n = \sum_{m=1}^{M_n}\sqrt{(x_m - x_{m+1})^2 + (y_m - y_{m+1})^2}, \end{cases}$$

where $M_n$ is the number of points in contour $C_n$. The contour $C_n$ is classified as containing a defect based on the threshold value pairs $\{S_{\min}; S_{\max}\}$ and $\{P_{\min}; P_{\max}\}$, if at least one of the following conditions is satisfied:

$$\begin{bmatrix} S_n < S_{\min}, \\ S_n > S_{\max}, \end{bmatrix} \quad \begin{bmatrix} P_n < P_{\min}, \\ P_n > P_{\max}. \end{bmatrix}$$

Thus, a controllable feature set is formed for each detected defect in the following form:

$$D_n = \{P_n, S_n, X_n, Y_n\},$$

where $\{X_n, Y_n\}$ are the coordinates of the centroid of the corresponding contour $C_n$. The resulting set $\{D_n\}$ serves as an analytical basis for subsequent visual and statistical analysis. After classifying contours as defective based on geometric criteria, a set of the areas of the detected objects $\{S_n\}$ is formed. To analyze the statistical characteristics of the distribution, the mean area $\overline{S}_n$ and the standard deviation $\sigma_S$ are estimated as follows:

$$\begin{cases} \overline{S}_n = \frac{1}{N}\sum_{n=1}^{N} S_n, \\ \sigma_S = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(S_n - \overline{S}_n)^2}. \end{cases}$$

The corresponding parameters make it possible to quantitatively characterize the variability of the geometric properties of the defects and to identify the presence of anomalous objects whose areas significantly deviate from the mean level. To visualize the statistical distribution, a histogram of defect areas is constructed and supplemented by a normal distribution approximation. In this case, the probability density is modeled by the function:

$$f_S = \frac{1}{\sqrt{2\pi}\,\sigma_S} e^{-\frac{(S_n - \overline{S}_n)}{2\sigma_S^2}}.$$

The parameters $\overline{S}_n$ and $\sigma_S$ are considered maximum likelihood estimates (MLE) for the normal distribution. The proximity of the empirical distribution toa normal profile serves as an indicator of the homogeneity of the detected defect class. Deviations from the normal distribution may indicate the presence of foreign objects or structural inhomogeneities.

## SOFTWARE-BASED DEFECT RECOGNITION IN THE CODE SEQUENCE STRUCTURE ON THE SURFACE OF MODULATION DISKS

To validate the functionality and effectiveness of the proposed approach, a software implementation of the defect identification algorithm for the code sequence structure of a modulation disk was developed. The corresponding software module integrates stages of preprocessing, morphological filtering, contour analysis, defect classification, and statistical evaluation of defect parameters. The user interface provides interactive tools for adjusting thresholding and classification parameters and enables visualization of processing results, including graphical representation of detected defects, histogram construction of defect areas, and tabular output of coordinates and numerical characteristics.

The defect identification algorithm was implemented as a modular Python application with a graphical user interface. The system architecture is based on the principles of separating image processing logic, parameter control, result visualization, and data export to external formats. This structure ensures flexibility, scalability, and ease of modification for individual stages of the algorithm. The software algorithm consists of three key components:

1. The graphical data processing module is responsible for the step-by-step transformation of the input image matrix, including Gaussian smoothing, adaptive thresholding, morphological filtering, contour detection, and the calculation of geometric and statistical parameters of the detected objects. The core element is the “ImageProcessor” class, which implements the main logic for binary mask analysis and the formation of the defect feature set.

2. The Graphical User Interface (GUI) is implemented using the “Tkinter” library. This component enables image loading, interactive adjustment of the adaptive thresholding value, visualization mode switching, display of analysis results, and result saving. The interface is divided into functional panels: the control panel, the visualization area, and the text fields for statistics and coordinates.

3. The result-saving mechanism enables the export of detected defects in graphical PNG and tabular CSV formats. The defect mask, annotated image with highlighted objects, and centroid coordinates can be saved as separate files for further use in technical inspection systems or external analysis.

To implement the aforementioned functions, the following external libraries were used: “OpenCV” for image loading, preprocessing, morphological operations, contour detection, and calculation of geometric parameters; “NumPy” for vectorized data processing and basic statistical computations; “Matplotlib” for generating histograms and visualizing the area distribution of detected defects; “Pandas” for constructing tabular structures and exporting results in CSV format; and “Scipy.stats” for approximating the area distribution using a normal distribution curve. The architectural design is based on a clearly structured separation of

component functions, enabling both local testing of individual modules and integration of the system into a broader software environment for technical inspection and machine-based analysis.

The defect detection algorithm is implemented using the methods of the "ImageProcessor" class and automated through interaction with the graphical user interface elements. The system's operational logic involves the sequential execution of the following stages:

- loading the grayscale image matrix, which reduces computational complexity by excluding color components;
- Gaussian smoothing with a fixed kernel to reduce noise levels and prepare the image for thresholding;
- adaptive thresholding using a local mean, with an adjustable offset parameter controlled via a slider;
- morphological filtering, including a closing operation to eliminate internal breaks and an opening operation to remove noise;
- contour analysis to determine the geometric characteristics of connected components (perimeter and area) and evaluate their compliance with predefined threshold criteria;
- classification of contours as defects based on whether their area or perimeter exceeds or falls below the specified threshold values.

The interface allows the user to modify key processing parameters in real time, such as the threshold offset for adaptive image binarization, the minimum perimeter value for classifying an object as defective, and a visualization mode toggle that enables or disables the overlay of circles on the centroids of detected defects. These parameters make it possible to adapt the algorithm's sensitivity to various lighting conditions, image scales, and defect types.

To evaluate the accuracy of the core functionalities performed by the modular Python application, verification was carried out using real microphotographs of modulation disk surfaces. The processing results demonstrate the system's ability to effectively detect defects originating from the photolithographic process by isolating anomalous regions based on geometric and statistical criteria. Figs. 3–5 present the processing outcomes for microimages of code sequence samples "1", "2", and "3", respectively, showing the original grayscale microimage, the binary mask with overlaid contours (green contours indicate objects without defect features, while white circular markers denote objects classified as defective), as well as the histogram of detected defect areas with an overlaid normal distribution curve and corresponding statistical data:

- total number of detected defects;
- average defect area;
- average defect perimeter;
- range of defect areas;
- range of defect perimeters.

The histograms constructed based on defect areas characterize the structure of the sample and visualize its variability. To approximate the empirical distribution, a normal distribution model was applied using the parameters of mean defect area and standard deviation. The obtained parameters represent maximum likelihood estimates and reflect a distribution skewed toward lower values, which is typical for defects associated with microcracks, scratches, and

contamination particles. It should be noted that statistical indicators provide insight not only into the number but also the nature of the defects. For instance, a high standard deviation indicates significant variability in defect sizes, which may suggest inconsistency in the technological process. Additionally, the coordinates of defect centroids can be used for targeted adjustment of the photolithography system and for initiating subsequent stages of detailed inspection.

Fig. 3 presents the results of applying the algorithm to code sequence sample "1", demonstrating the system's capability to effectively localize both isolated anomalies and small-scale digital artifacts, thereby enabling a comprehensive assessment of the processed surface condition.

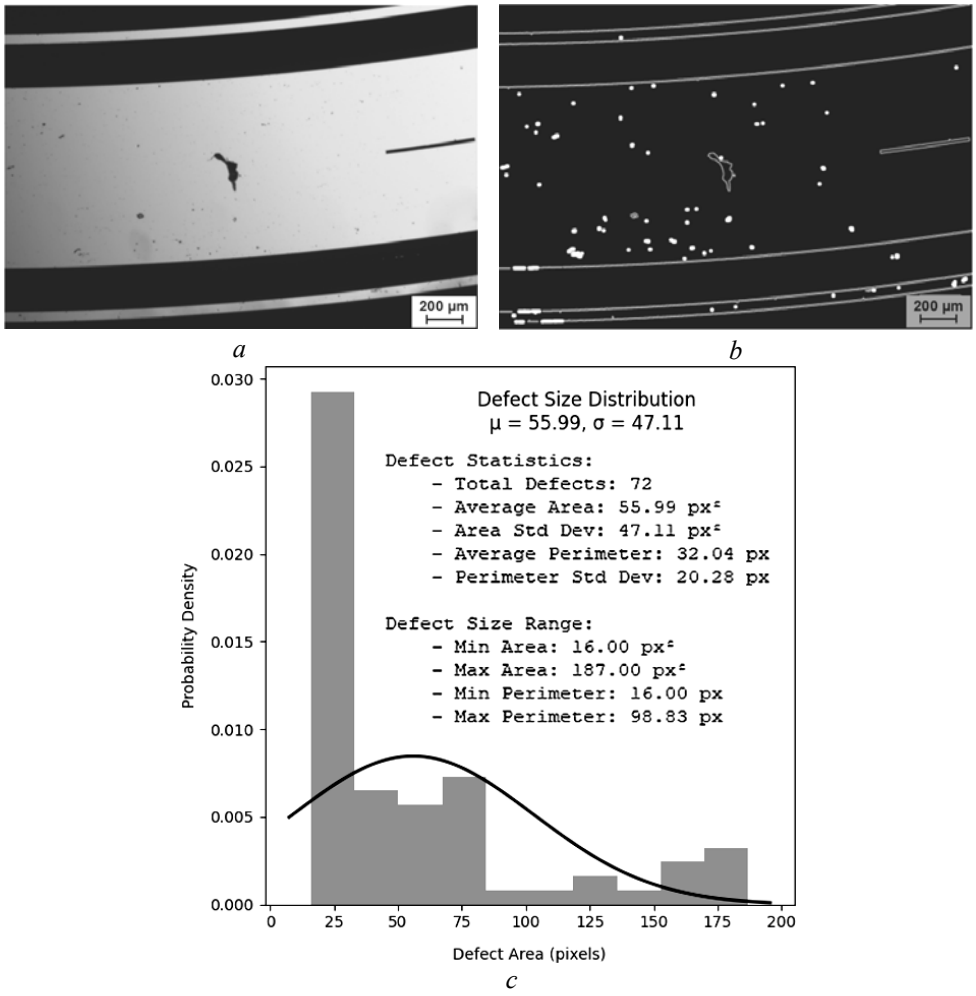



*Fig. 3*. Processing results for code sequence sample "1": *a* — original grayscale microimage; *b* — binary mask with overlaid contours; *c* — histogram of detected defect area distribution

Fig. 4 shows the analysis results for code sequence sample "2", indicating a higher total number of defects but with a lower maximum area and less pronounced dominance of a single large defect. This suggests a different nature of structural disturbance in the binary code sequence compared to sample "1", potentially associated with dust or contamination deposition processes or exposure instability in certain regions.

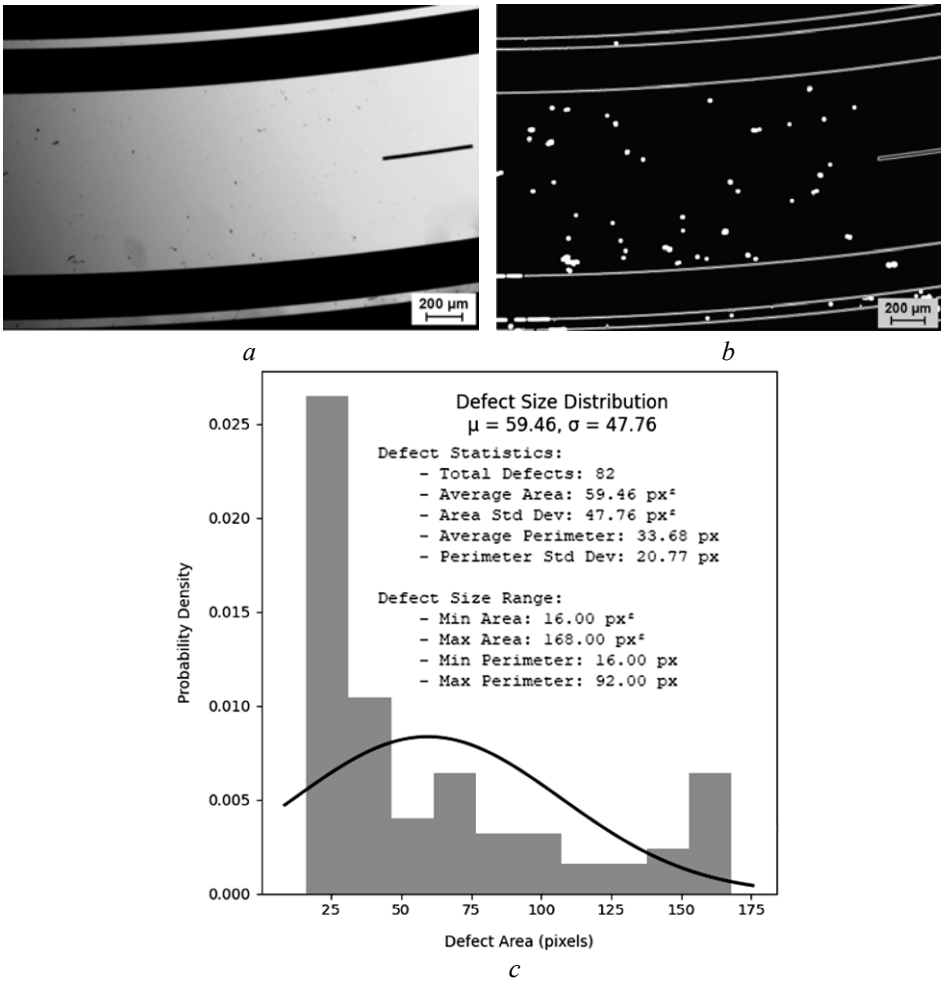



*Fig. 4*. Processing results for code sequence sample "2": *a* — original grayscale microimage; *b* — binary mask with overlaid contours; *c* — histogram of detected defect area distribution

Fig. 5 presents the processing results for code sequence sample "3", which contains high-contrast geometric structures and noticeable foreign inclusions. Sample "3" is characterized by greater area dispersion and the presence of pronounced macro-scale defects. This is confirmed by both the numerical characteristics and the shape of the histogram, where the normal distribution curve exhibits strong asymmetry. Such results indicate localized disruptions during fabrication or damage incurred during operation.

The presented results confirm the stability and consistency of the algorithm's performance under varying input conditions, such as defect geometry, image contrast, and noise variability. Thus, the proposed approach demonstrates high sensitivity to local structural anomalies while maintaining robustness against background artifacts and digital noise. The analysis of defect area distribution histograms shows that the system can adapt to changes in the nature of damage and maintain the reliability of statistical evaluation even in cases of asymmetric or anomalous distributions. As part of future improvements, it is planned to extend the algorithm by integrating machine learning classifiers for automatic defect type identification, incorporating spatial context in the analysis of centroid distribution,

and optimizing processing procedures for implementation on computational modules of embedded machine analysis systems operating in real time.

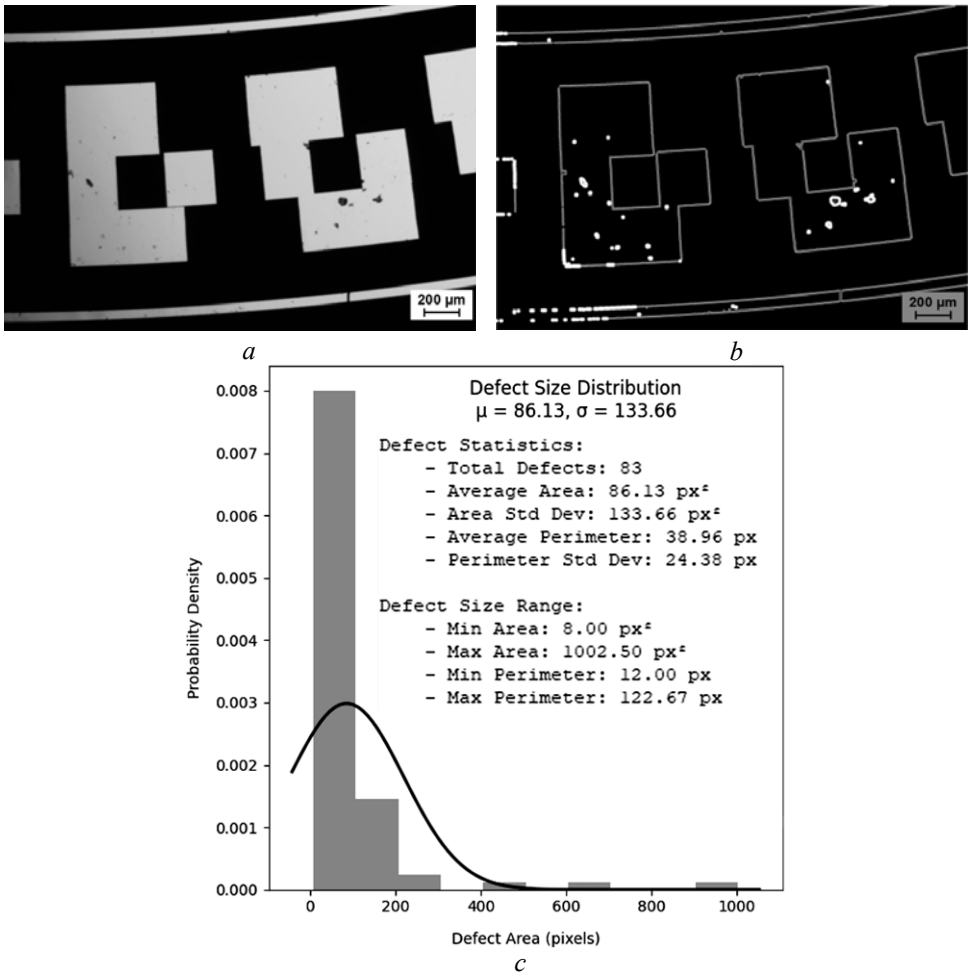



*Fig. 5*. Processing results for code sequence sample "3": *a* — original grayscale microimage; *b* — binary mask with overlaid contours; *c* — histogram of detected defect area distribution

## CONCLUSIONS

The article presents a comprehensive methodology for the automatic detection of defects in the binary structure of a code sequence on the surface of modulation disks, combining image preprocessing methods, morphological analysis, and statistical evaluation of the geometric characteristics of objects. A mathematical model is proposed that describes the stages of smoothing, adaptive thresholding, filtering, and contour detection, followed by classification based on area and perimeter. The developed software module provides a complete processing cycle of the input image: from conversion into a binary mask to the visualization of detected defects and construction of histograms with normal distribution approximation. The modular system architecture and the presence of a user interface that allows adjustment of key parameters enable the adaptation of the program to variations in image quality, scale, and the nature of defects. Experimental verification on samples of binary code sequences of modulation disks demonstrated the algo-

rithm's ability to detect both microdefects and local structural anomalies of considerable area. The stability of results under varying processing parameters confirms the algorithm's adaptability and its suitability for implementation in technical diagnostic systems under constrained computational resources. Further extension of the software functionality is possible through the use of machine learning classifiers, application of spatial contextual analysis, and integration with real-time hardware platforms to enable autonomous monitoring.

**Conflicts of interest.** There are no conflicts to declare.

**Acknowledgments.** The authors express their deep gratitude to the National Research Foundation of Ukraine for financial support under the project No. 2023.04/0004.

## REFERENCES

1. Y. Huang, Y. Yang, J. Liang, Z. Miao, M. Zhao, Y. Zheng, "An optical glass plane angle measuring system with photoelectric autocollimator," *Nanotechnology and Precision Engineering*, 2(2), pp. 71–76, 2019. doi: https://doi.org/10.1016/j.npe.2019.06.001
2. L. Lei et al., "A study on length traceability and diffraction efficiency of chromium gratings," *Photonics*, 11(3), 233, 2024. doi: https://doi.org/10.3390/photonics11030233
3. T. Wavrunek, S. Ball, Z. Gotto, B. White, "An Adhesion-based Alternative to Solvent Processing in Microfabrication," *Proceedings of The National Conference On Undergraduate Research (NCUR) 2020 Montana State University*, 2020. Available: https://libjournals.unca.edu/ncur/wp-content/uploads/2021/01/3238-Trevor-Wavrunek-FINAL.pdf
4. I.V. Kosyak, D.Yu. Manko, Ie.V. Belyak, A.A. Kryuchyn, "Methodology for transforming code sequences in accordance with the modulation disk coordinate system," (in Ukrainian), *Electronic Modeling*, 46(5), pp. 35–49, 2024. doi: https://doi.org/10.15407/emodel.46.05.035
5. V.V. Petrov, A.A. Kryuchyn, Ie.V. Beliak, D.Yu. Manko, I.V. Kosyak, O.G. Melnik, "Advantages of Direct Laser Writing for Enhancing the Resolution of Diffractive Optical Element Fabrication Processes," *Physics and Chemistry of Solid State*, 5(3), pp. 587–594, 2024. doi: https://doi.org/10.15330/pcss.25.3.587-594
6. A.A. Kryuchyn et al., "Prospects for the creation of the technology of maskless photolithography based on direct laser recording," *Semiconductor Physics, Quantum Electronics & Optoelectronics*, 28(1), pp. 93–101, 2025. doi: https://doi.org/10.15407/spqeo28.01.093
7. E.R. Davies, *Computer vision: Principles, algorithms, applications, learning.* Academic Press, 2018. doi: https://doi.org/10.1016/C2015-0-05563-0
8. X. Chen, "Optimization of image processing methods based on wavelet transform and adaptive thresholding," *Applied Mathematics and Nonlinear Sciences*, 9(1), 2023. doi: https://doi.org/10.2478/amns.2023.2.00665
9. E. Turajlic, E. Buza, A. Akagic, "Honey Badger algorithm and chef-based optimization algorithm for Multilevel Thresholding Image segmentation," *2022 30th Telecommunications Forum (TELFOR).* doi: https://doi.org/10.1109/telfor56187.2022.9983775
10. D. Sundararajan, "Morphological image processing," *Digital Image Processing*, pp. 217–256, 2017. doi: https://doi.org/10.1007/978-981-10-6113-4_8
11. Z. Lyu, C. Zhang, M. Han, "A nonsubsampled countourlet transform based CNN for real image denoising," *Signal Processing: Image Communication*, 82, 115727, 2020. doi: https://doi.org/10.1016/j.image.2019.115727
12. Z. Huang, H. Lu, X. Yu, H. Xiao, "Multi-Scale Feature Guided Transformer for Image inpainting," *IET Image Processing*, 19(1), 2025. doi: https://doi.org/10.1049/ipr2.70105

13. P. Li, J. Chen, C. Cai, “Reinforced Res-Unet transformer for underwater image enhancement,” *Signal Processing: Image Communication*, 127, 117154, 2024. doi: https://doi.org/10.1016/j.image.2024.117154
14. H. Wang, X. Lu, Z. Wu, R. Li, J. Wang, “Infrared and visible image fusion based on Autoencoder Network,” *IET Image Processing*, 19(1), 2025. doi: https://doi.org/10.1049/ipr2.70086

*Received 26.06.2025*

**INFORMATION ON THE ARTICLE**

**Dmytro Yu. Manko,** ORCID: 0000-0003-1848-2952, Institute for Information Recording of NAS of Ukraine, Ukraine, e-mail: dmitriy.manko@gmail.com

**Ievgen V. Beliak,** ORCID: 0000-0001-9045-0782, Institute for Information Recording of NAS of Ukraine, Ukraine, e-mail: beliak1312@gmail.com

**Andriy A. Kryuchyn,** ORCID: 0000-0002-5063-4146, Institute for Information Recording of NAS of Ukraine, Ukraine, e-mail: kryuchyn@gmail.com

**Ruslan M. Ishchenko,** ORCID: 0000-0003-0158-4020, National Transport University, Ukraine, e-mail: rm_ischenko@ukr.net

**Valentyna V. Zavarzina,** ORCID: 0009-0005-1666-3620, National Transport University, Ukraine, e-mail: valazavarzina48@gmail.com

**РОЗРОБЛЕННЯ АЛГОРИТМІВ РОЗПІЗНАВАННЯ ДЕФЕКТІВ У СТРУКТУРІ КОДОВОЇ ПОСЛІДОВНОСТІ НА ПОВЕРХНІ МОДУЛЯЦІЙНИХ ДИСКІВ** / Д.Ю. Манько, Є.В. Беляк, А.А. Крючин, Р.М. Іщенко, В.В. Заварзіна

**Анотація.** Дослідження присвячено алгоритмам виявлення та локалізації дефектів у структурах кодової послідовності на поверхнях модуляційних дисків. Воно спрямоване на невеликі аномалії в літографічно структурованих елементах, які можуть спричинити помилки зчитування або зниження точності вимірювання. Багаторівнева модель оброблення зображень поєднує гауссове згладжування, адаптивне порогове визначення, морфологічні операції та сегментацію на основі контурів. Етапи оброблення формалізовано як математичні оператори для відтворюваної реалізації. Дефекти характеризуються за допомогою метрик на основі периметра та площі, а їх розподіл за площею апроксимується нормальним законом. Просторова модель обчислює центроїди дефектів, що дає змогу виконувати порівняльне оцінювання якості зразків дисків. Програмне забезпечення надає інтерфейс для налаштування порогів, візуалізації контурів та графіків площ дефектів, а також експорту результатів. Тести на реальних дефектних дисках підтверджують надійне виявлення локальних структурних порушень та придатність методу для діагностичних систем.

**Ключові слова:** модуляційні диски, автоматизований контроль, кодова послідовність, порушення мікроструктури, попереднє оброблення зображень, морфологічний аналіз, контурна сегментація.

UDC 623.4/519.2+004.75
DOI: 10.20535/SRIT.2308-8893.2025.4.02

# THE RESULTS OF THE MULTI-POSITION SURVEILLANCE SYSTEM'S EFFICIENCY, DEPENDING ON THE LOCATIONS OF ITS SENSORS, USING ADDITIONAL DATA PROCESSING

**V.Yu. TYMCHUK, O.O. MEDIAKOV, O.O. POPOV,
T.V. TRYSNYUK, S.A. TSYBULIA**

**Abstract**. The efficiency of a multi-position system depends on the realization of its structure—how many elements it includes, where they are located, and how the environment and terrain influence its operation. The paper is dedicated to data processing in a multi-position surveillance system as an additional option, leveraging the in-between big data from the system's elements. A sufficient number of numerical data generated by the multi-position system and its elements–sensors–allows the use of statistical methods and models from machine learning or deep learning. The ontology for quality estimation of the multi-position system, depending on its configurations, is proposed. The results of the distributions of detected events are presented in graphical forms that allow statistical evaluation of the distributed data. Our findings allow us to ensure the efficiency of a multi-position system in an unpredictable, variable environment by reconfiguring it when it offers better capabilities.

**Keywords**: data processing, surveillance, detection, multi-position system, system-of-systems, efficiency.

## INTRODUCTION

Being multidisciplinary, science is able to cover different areas simultaneously. Our research corresponds to this feature — there are several independent areas from System Engineering, Estimation theory, Geospatial Intelligence, Big Data Processing, Machine Learning (ML), Deep Learning (DL). For example, there is a well-known problem of accuracy in Detection theory that has limits when using its methods and algorithms, but due to collaborating with methods and algorithms from other scientific fields such limits would be overcome. Another example is a problem of detection system's design to be optimized for unpredictable and variable environment. The mentioned problem takes place for multi-positional detection system. It is a very difficult mission to define the effective structure of multi-positional detection system especially during warfare. This research investigates a method for evaluating a detection system's structure based on its operational results, — the results for in-between data (not the results of direct detection) from elements of multi-positional detection system are presented. As usual, any system

produces a great part of in-between data that could be additional information when it is necessary to improve the efficiency of the system. The volume of in-between data is enormous, so it makes possible to use Big Data Processing, ML or DL. So, the data preprocessing and statistical analysis were used to present the detection results on additional grid coordinate system and to make a statistical evaluation for the data.

## PROBLEM STATEMENT

The theory and practice of developing multi-position systems have been well-studied for a long time [1]. The problem to be solved is formulated as follows. Having some specific uncertain environment, the scientific and technical task is to register certain changes in this environment using technical solutions implemented in the systems. There are many areas of application for such a problem — studying nature and space, medicine, monitoring the security situation at infrastructure facilities, military monitoring systems [2]. Obviously, there are different well-known kinds of passive multi-position surveillance system (MPSS) for acoustic location, GPS tracking, imagery intelligence, seismic reflection, signals intelligence, thermal imaging, underwater acoustics, video surveillance, wireless tracking etc. The physical nature for each method is unique, but there are some common similarities: 1) waves (signals); 2) a set of distributed synchronous sensors; 3) a great volume of measurement results — the datasets.

Therefore, if a certain system has already been developed and is functioning, then the main next task is to ensure the best (optimal) way to process information (data, signals).

In modern systems, whether specialized functional systems [3; 4], complex multi-profile systems [5], or systems of systems [6] — signal and information processing occurs across multiple stages. These processes are diverse and often complementary, but the specific way used to identify them is less critical than their integration. Regardless of the system's architecture, an "integrated" processing pipeline inevitably incorporates Big Data analytics or the processing of large volumes of homogeneous data alongside other methods, algorithms, and software solutions [7].

**The purpose of the research.** Our goal is to make a qualitive analysis of the efficiency of multi-position system using data preprocessing and statistical data analysis to change its configuration improving the detection possibilities.

### The general aspects of a configuration of a multi-position surveillance system

A passive MPSS with a set of typical sensor posts (SPs) for terrain monitoring and/or control is done. The configuration of MPSS is the combination of all SPs on the terrain in some grid-like projection. The example of configuration is on the Fig. 1, where from 1 to 9 are the typical SPs being synchronous in a passive MPSS. The SPs are distributed on the terrain in some way, on the picture it is used a distance of 1.5 km between two sensors in the first (upper) line of the configuration and a distance of 1 km between first and second (below) lines. The MPSS base line is the distance between two utmost SPs (here are SP1 and SP6). The sensitivity for a MPSS means the possibility to detect some signals (on the potentially attainable distance for the signal with some defined level (power)). Here the range area is a circle of 25 km in radius (for the optimal conditions).

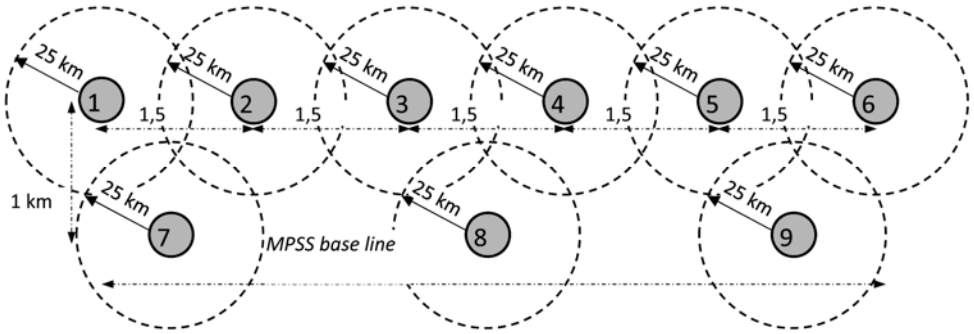



*Fig. 1.* The example of possible configuration for a typical MPSS (the sketch is not at scale)

As it is clear from the Fig. 1, the potential area of MPSS's sensitivity may have some form with 25 km in depth for any side direction from the utmost sensors. The two lines in MPSS's configuration allow for the determination of the Area of Interest (AoI) by excluding, for example, signals from back (useless) directions. Such a task is typical in security or military applications where both friendly (allied) and adversary forces are present. The optimal AoI for some general configuration and conditions should have some right form (Fig. 2). The orthogonal line (OL) to base line shows a main direction of AoI (main surveillance angle (MSA) is an angle between OL and main danger direction (MDD)).

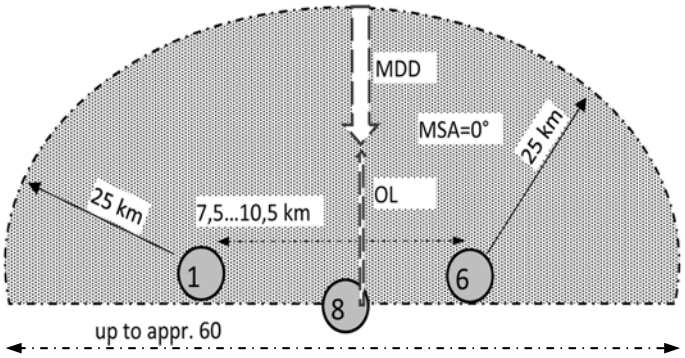



*Fig. 2.* The form of the MPSS's sensitivity (dotted area)

*A note*: 1 — The typical sensors 2–5 from the upper line (Fig. 1), sensors 7 and 9 from the below line are missing here to have a simple picture. MSA here is 0 degree that is the best for AoI

In real-world conditions, an ideal AoI does not exist. The first reason for it is the terrain features where the MPSS's configuration is set — there are no possibilities to establish typical sensors on the straight line (or two lines) with even distance between sensors. Further, the real terrain is characterizing with relief and natural or artificial coverings — the such reality effects on the MPSS's sensitivity. The second reason is some secure restrictions which are inevitable for warfare. The technical challenges (such as the problems with a sensor's operating or with some kind of destroying in warfare) are the next factor that makes an effect on the MPSS. So, the real MPSS's configuration has a non-optimal decision with a smaller number of sensors in a usage (Fig. 3) and it results in a less achievable AoI (Fig. 4).

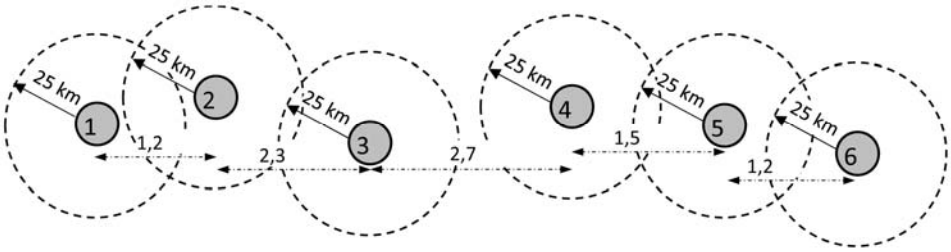



*Fig. 3.* The example of a real MPSS's configuration for some terrain and secure conditions (the sketch is not at scale)

So, for such occasional conditions of systems' utilization there is not just a technical optimization problem (which defends only from specifications of a system), but a procedure one too, especially how to get the best approaches for next processing of ongoing datasets during the time of MPSS's functionating.

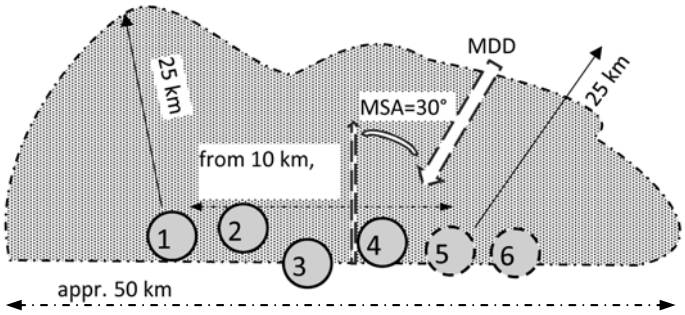



*Fig. 4.* The variant of the real AoI for s MPSS's (MPSS's sensitivity)

**The acoustic location multi-position surveillance system**

The stages of data preprocessing and data processing are presented on the example of an acoustic location MPSS.

The principles of acoustic MPSS are well-known: due to automatized detection and classification procedures, sensitive synchronous sensors and wireless communications the determination of the sound source from the muzzle wave Times of Arrival is a simple task for such kind of MPSS [8]. But the problems of the sound source location accuracy and even the recognition in typical circumstances when the various waves emitted by and during a shot (from many sound sources) are still actual. The reasons have the nature character — the initial projectile characteristics (the whole variety of its aerodynamic and ballistic coefficients), the range (Fig. 5), the atmosphere (with atmospheric wind (Fig. 6) and

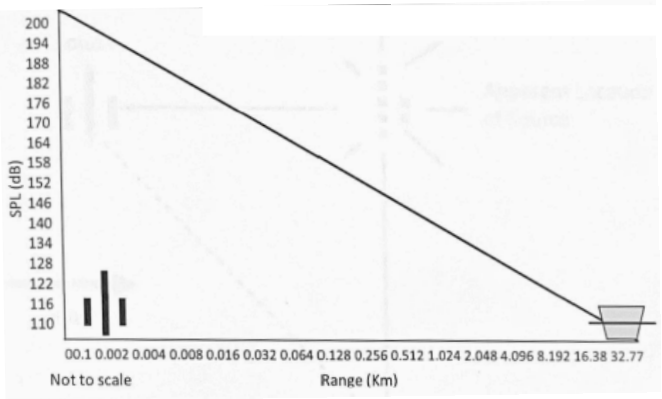



*Fig. 5.* The attenuation of the sound wave during its propagation in the atmosphere [11]

sound speed gradients), the ground, possible obstacles (woods, buildings, hills…, (Fig. 7)), refraction or air absorption, wave alterations, multipath arrival of the ballistic shock wave and so on. So, the outcome that the localization performance is affected, sometimes critically [9], is firm. Even more, the practice with highly intensive shot conditions that took place in Ukraine since 2022-24-02 [10], approved this statement.

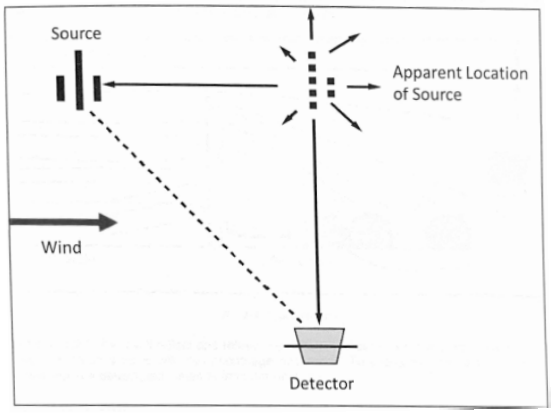



*Fig. 6.* The wind influence on the sound wave and source's localization [11]

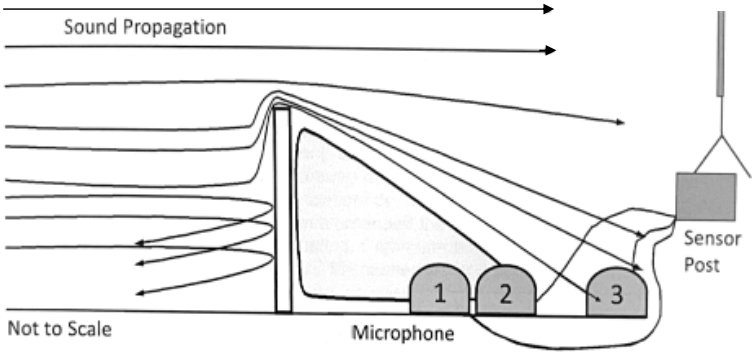



*Fig. 7.* The probable role of obstacles on the sound wave path in poor operation of a MPSS' sensor [11]

Therefore, many modern researchers are dedicated to improving the accuracy of acoustic MPSS in sound source localization. There are two main directions of the efforts: 1) to improve the established approaches (due to technical and different tools' progress, and processing method with some more efficiency [8; 9; 12; 13]); 2) to search new approaches for problem solving. The example of the second direction is an application of artificial intelligence — convolutional recurrent neural networks in [14] or other neural networks on the proposed software-mathematical models [15], depending on the shape of the location of the sensors (MPSS's configuration), the distance between SP, their number, the parameters of the neural network (the number of hidden layers and neurons), and the volume of the dataset for training. The authors proclaimed that their results for the neural network training algorithm ensure the average value of the absolute error in determining the grid coordinate are not exceed about 1 m and maximum absolute error value are not exceed 16 m for the Y grid coordinate (it corresponds the determination in depth) and 4.5 m for the X grid coordinate (it corresponds the determination in front) for the range 1800 m between sensors and a source of acoustic signals. Even if it is a fact for testing conditions (optimal

terrain specification and for an ideal configuration (Fig. 1) it is not useful for real circumstances, because the practical range between SP and a sound source is in 5…15 times much more, that means the average or maximum absolute error values are increasing properly — up to 50…300 m.

To approve the mentioned statement the field data are presented below.

**Some field data from the acoustic MPSS and their explanation**

The type of acoustic MPSS for the gathering data in field conditions is HALO [11]. According to the specifications of the HALO system the average value of the absolute error should be 100 m on the range up to 100 m and 0.7% of the range when the distance to the sound source exceeds 15 km.

The configuration for the HALO was occasional (Fig. 3) and included or 4, or 5, or 6 sensors with distance between them from 1.5 km till 4 km.

The field results for initial utilization of HALO are shown in the Table 1.

**Table 1.** The initial data about HALO's accuracy

| **Range between MPSS and sound source** | **Number of sensors with the base line distance** | | |
|---|---|---|---|
| | 5 SPs for 13.8 km *or* 6 SPs for 16.5 km | | 4 SPs for 11.2 km |
| | MSA (*see Fig. 4*) | | |
| | 0…±30 º | ±30…±60 º | 0…±30 º |
| | The average value of the absolute error | | |
| 7…10 km | 200…300 m | 500…700 m | 500…800 m |
| 10…15 km | 400…500 m | 1000 m | 1000…1500 m |
| 15…20 km | 600…800 m | 1500…2000 m | - |
| 20…30 km | 1000 m | - | - |

*A note*: 1 — The time period of observation is a month (since 2023-05-04 till 2023-08-05).

To estimate the value of the error it was possible to use the radar system for the same sound sources. The type of the radar is AN/TPQ-36 with accuracy up to 50 m [16].

**Table 2.** The comparison data between Radar's and HALO's target grid co-ordinates

| **Nr.** | **Detection time** | | **Location difference, m** | **Range** (between SPs of MPSS and sound source)**, m** | |
|---|---|---|---|---|---|
| | HALO | Radar | | | |
| 1 | 23:24 | 23:30 | 818 | 22240 | 26920 |
| 2 | 23:25 | 23:30 | 877 | - // - | - // - |
| 3 | 10:15 | 10:20 | 391 | 15310 | 18830 |
| 4 | 10:18 | 10:20 | 151 | - // - | - // - |
| 5 | 10:42 | 10:55 | 108 | 15480 | 19020 |
| 6 | 10:46 | 10:55 | 46 | - // - | - // - |
| 7 | 10:38 | 10:30 | 98 | 13320 | 16720 |
| 8 | 10:39 | 10:30 | 198 | - // - | - // - |
| 9 | 16:21 | 16:20 | 513 | 21790 | 26400 |
| 10 | 08:36 | 08:38 | 145 | 15480 | 18990 |
| 11 | 08:38 | 08:38 | 419 | - // - | - // - |
| 12 | 08:19 | 08:20 | 153 | 15390 | 18940 |
| 13 | 08:21 | 08:20 | 226 | - // - | - // - |
| 14 | 08:23 | 08:20 | 446 | - // - | - // - |

It is obviously that the statistical data for reliable conclusions are too little. But the main value of the data is their real (utilization) nature, not experimental.

It should be noted that the influence of meteorological data (which is required for HALO) was not considered. The average data are presented in the Table 3.

**Table 3.** The average data of HALO's accuracy during the field utilization

| Range between SP and sound source | Manual's meaning of HALO's error | The deviation diapason of sound source's coordinates from actual ones |
|---|---|---|
| 15…20 km | 105…140 m | up to 100 m |
| 20…30 km | 140…210 m | 500…900 m |

*A note*: 1 — The MPSS consists of six SPs, the MPSS's base line is 16.5 km, the MSA is 0…±30º.

So, the average error values may correspond to manual's ones (according to HALO's specifications) on the ranges up to 20 km. But it is not enough for direct application the surveillance data on next stages of decision-making. So, it is necessary to find approaches how to decrease average and maximum absolute error values.

**The input conditions and restrictions**

The objective of monitoring by means of MPSS is to determine the location of the targets after at least two systems' sensors produced information about detection of the target.

The MPSS consists of 6 sensors [10].

The fact of generating some kind of signal that was received/detected by a sensor(s) will be called an event.

The detected signal is the information about event transformed in some digital form including some specific features about the event, like time and potential coordinates.

The digital event set (DES) is the ordered combination of numbers that is generated by a sensor detecting the signal. Normally the DES includes time of detection, coordinates of a sensor that detects a signal and coordinates of an event, derived from some signal and information processing.

The proposed dataset is the collection of DES collected during some period of MPSS's operating. Several dataset sources can be used, so the corresponding indication of those sources is included where needed.

The size of dataset is more than 17 000 rows. The example of the raw dataset is shown on Fig. 8 (its description is given in the note under the figure's title). So, the presented digital data for the data preprocessing and EDA were used similar to [17]. Gaining the ability to evaluate the performance of the MPSS depending on the configuration is easily (to a certain degree) doable.

Time of occurrence of the event isn't taken into account for the statistical analysis.

The common operational picture (COP) is a presentation of all detected events on some grid system.

## SOLVING THE PROBLEM OF INEFFICIENCY OF MULTI-POSITION SYSTEM IN UNPREDICTABLE AND VARIABLE ENVIRONMENT

### Experiment Design/Data Collection

To represent results of dataset processing including corresponding diagrams and graphic elements in the best form some abbreviations and acronyms for the terms and processes taking place in our research are proposed (Table 4).

**Table 4.** A physical essence of some processes in MPSS with designations and symbols

| N | The terms and their designation or symbolization | | |
|---|---|---|---|
| | Term or characteristics | Acronym | Symbol |
| 1 | Configuration | Conf. | $\langle Conf \rangle^j$ |
| 2 | Number of sensors in a configuration | – | $K$ |
| 3 | Sensor Post from a configuration with a sequence number[1] $k$ | SP | $SP_k$ |
| 4 | Configuration Duration[2] | CD | $T$ |
| 5 | Configuration Sencitivity Terrain[3] | CST | $(X,Y)$ |
| 6 | Number of Events being Detected[4] | NED | $N$ |
| 7 | Event sequence number | $i$ | $\cdots_i$ |
| 8 | Total Events Space Distribution | TESD | $\{W\}_i$ |
| 9 | Event Space-Gradient Map[5] | ESGM | $\{\widetilde{W}\}_K$ |
| 10 | Locating Posts' Number Portion[6] | LPNP | $P_{\hat{n}}$ |
| 11 | Single Post Operating | SOP | $P_k$ |

*Notes*: 1 — It is generated in MPSS automatically; 2 — Time period of MPSS's operating with some configuration; 3 — Some terrain (with defined size) that is being achievable for MPSS to detect events; 4 — Using upper and lower indexes it is differed a NED for one or another configuration and/or CDs (f.e., per a day, per a week, …); 5 — It shows the terrain (as a gradient surface) in which events were detected by all SPs; 6 — The ratio of number of SPs that detected each event to all SPs from a configuration

### The methodology aspects for analysis of MPPS

Specialized geoinformatics system (GIS) software is usually integrated into MPSS to display the current event situation [18]. Usually, such GIS is a 3D model of terrain with a set of coordinate systems. In the research it was used a relative simulated Cartesian coordinate system (CCS) that "covers" a CST and displays the configuration in the center of it. So, all events detected by SPs are reproduced on this simulated CCS.

The ontology of a configuration is depicted as

$$\langle Conf \rangle^j \triangleright (K, \{\lambda\phi\}_k)^j , \tag{1}$$

where $j$ is a relative sequence number for possible configurations (there were 6 ones); $K$ is the number of sensors in a configuration; $\lambda$ & $\phi$ are the coordinates of SP (a longitude and a latitude for $SP_k$). Sometimes the UTM-like coordinates

are used instead of latitude-longitude, but the conversion between those is a simple computational task:

$$\{\lambda\phi\}_{\dots} \rightarrow \{X\,Y\}_{\dots} . \tag{2}$$

ML system has a typical design for such systems [9].

**The ontology for configuration quality estimation**

A collection of several statistical characteristics that are used for demonstration and estimation of how a specific configuration operates in a CST is called the configuration quality (CQ) within the context of the paper.

TESD shows the density of detected events:

$$\{W\}_i = \sum_{i}^{\hat{N}_{total}} \{X,Y,t\}_i , \tag{3}$$

where $\hat{N}_{total}$ is a total NED for the *j*-configuration.

ESGM shows the terrain area where the events were detected by all SPs:

$$\{\widetilde{W}\}_K = \bigcap_{k=1}^{K} \{W\}_{i/k} , \tag{4}$$

where $\{W\}_{i/k}$ is an event distribution detected by $\mathrm{SP}_k$.

Apparently, each event may be detected by a single SP or by a particular combination of SPs. Consequently, LPNP is defined as:

$$P_{\hat{n}} = N_{\hat{n}} / \hat{N}_{total} , \tag{5}$$

where $1 < \hat{n} \le K$, meaning that $N_{\hat{n}}$ is a sum of NED for the cases detected by a combination of $\hat{n}$ SPs (neglecting the actual composition of SPs in the combination, only the number of SPs is considered).

SOP shows how frequently each single SP detected events registered in the dataset:

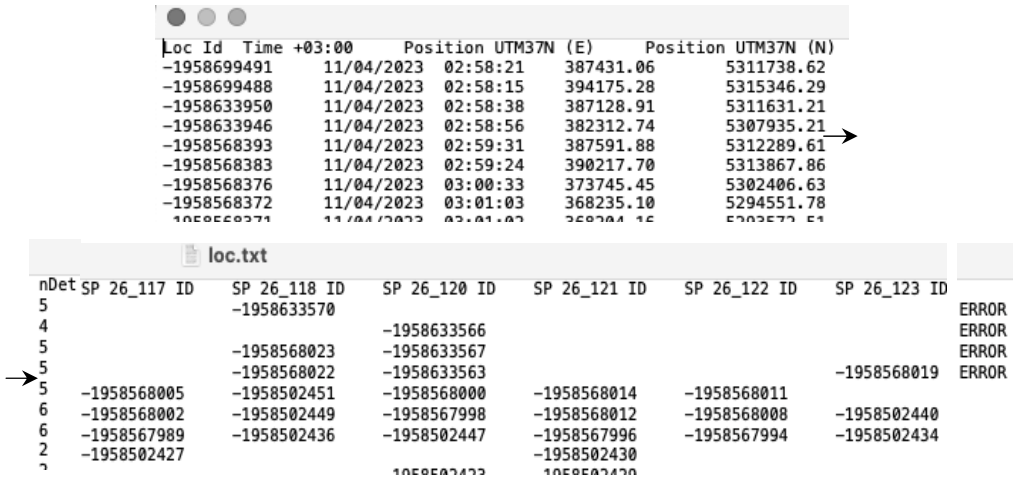
$$P_k = N_k / \hat{N}_{total} , \quad k = (1 \dots K) . \tag{6}$$

**Configuration quality CQ presentation**

So, there were 6 configurations with different parameters according to (1). Each configuration had a NED that is enough for ML (see depiction for figures). To analyze CQs it was proposed to calculate and present five graph materials for each configuration.

TESD presents the distribution of events that were detected in MPSS during time period corresponding to configuration duration *T*. Each point for presentation it was taken from geographic coordinates of corresponding row of dataset (Fig. 8) according to (3). The geographic coordinates were transformed into grid one according to (2). The grid system is conditional and it has no connection with any known grid systems such as MGRS. It was got just for local terrain. The central point for this grid is approximately in the geometric center of MPPS's base that includes all sensors. As common the form for MPSS's base is similar to line. The limits for terrain area are not far than the biggest distance (from central point to event's location) projected on grids. In the presented case there is a 50-km area

from the geometric center (in each of four grid directions). Due to terrain scale restrictions, it was used a gradient scale to present the level of concentration detected events in limited location areas. For example, gradient scale is from 0 to 80 on a Figs. 9, 11, 12, 13, 17, 18.



| Loc Id | Time +03:00 | Position UTM37N (E) | Position UTM37N (N) |
|---|---|---|---|
| -1958699491 | 11/04/2023 02:58:21 | 387431.06 | 5311738.62 |
| -1958699488 | 11/04/2023 02:58:15 | 394175.28 | 5315346.29 |
| -1958633950 | 11/04/2023 02:58:38 | 387128.91 | 5311631.21 |
| -1958633946 | 11/04/2023 02:58:56 | 382312.74 | 5307935.21 |
| -1958568393 | 11/04/2023 02:59:31 | 387591.88 | 5312289.61 |
| -1958568383 | 11/04/2023 02:59:24 | 390217.70 | 5313867.86 |
| -1958568376 | 11/04/2023 03:00:33 | 373745.45 | 5302406.63 |
| -1958568372 | 11/04/2023 03:01:03 | 368235.10 | 5294551.78 |

loc.txt

| nDet | SP 26_117 ID | SP 26_118 ID | SP 26_120 ID | SP 26_121 ID | SP 26_122 ID | SP 26_123 ID | |
|---|---|---|---|---|---|---|---|
| 5 | | -1958633570 | | | | | ERROR |
| 4 | | | -1958633566 | | | | ERROR |
| 5 | | -1958568023 | -1958633567 | | | | ERROR |
| 5 | | -1958568022 | -1958633563 | | | -1958568019 | ERROR |
| 5 | -1958568005 | -1958502451 | -1958568000 | -1958568014 | -1958568011 | | |
| 6 | -1958568002 | -1958502449 | -1958567998 | -1958568012 | -1958568008 | -1958502440 | |
| 6 | -1958567989 | -1958502436 | -1958502447 | -1958567996 | -1958567994 | -1958502434 | |
| 2 | -1958502427 | | | -1958502430 | | | |

*Fig. 8.* The example[1] of real data from a MPSS

*A note*: 1 — MPSS generates automatically unique identification numbers for each sensor that detects a target and for a whole system when at least two sensors detect the same target (the number of sensors that detect a joint target is ina column that is headed as nDet). The corresponding identification numbers (ID from one of six sensors (they are defined as SP_117, SP_118, etc.) and Loc ID from a MPSS are presented on the example — 1958699491, 1958633570, etc.). The time detection for a target and its coordinates — geographic longitude and latitude — are also presented in the proper columns

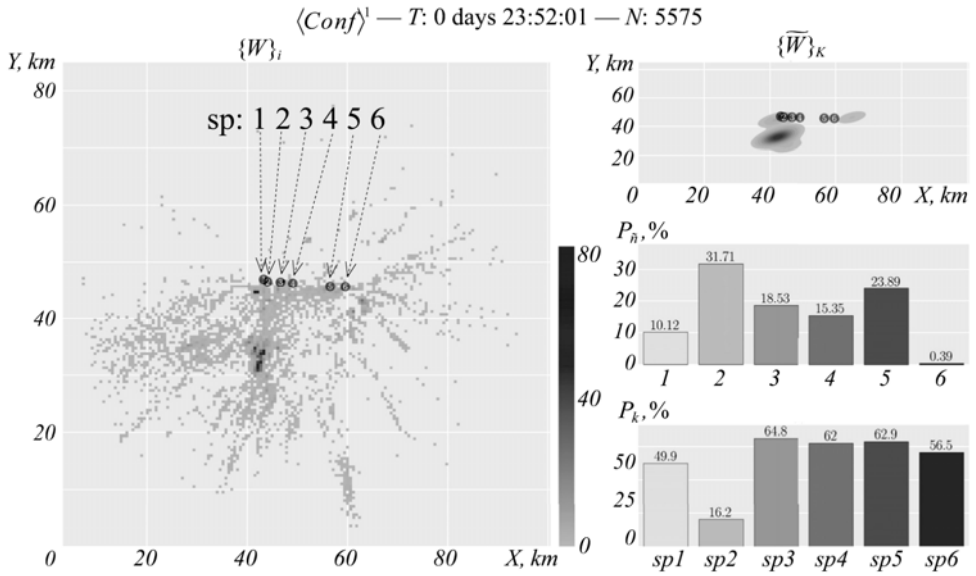



*Fig. 9.* The Total Events Space Distribution $\{W\}_i$ (half-left), Event Space-Gradient Map $\{\widetilde{W}\}_K$ (top-right), Locating Posts' Number Partion $P_{\bar{n}}$ (middle-right) and Single Post Operating $P_k$ (bottom-right) for Configuration Nr. 1 of MPSS ($T \approx d$, NED (corresponds to the size of dataset) $N = 5575$

The next graph of CQs (top-right) shows the gradient distribution of detected events that were detected by all 6 sensors of MPSS simultaneously (or with understandable time delay) according to (4). The graph allows to estimate both some aspects of MPSS's configuration and terrain peculiarities. In first case the operator of MPSS may define ineffective sensors in case when the following result is happened: the event is localized in the whole-area but there is no detected information from one (or more) sensors of 6. In second case the operator could analyze how the terrain influenced on MPSS's possibilities and what variants of changing the configuration would be useful for better detection.

The third illustration of CQs (middle-right) is a diagram showing what is a portion of numbers of sensors in event detection according to (5). As it was mentioned in the Fig. 8's note, there are unique identification numbers in the MPSS — location ones and numbers from each of 6 sensors. The quantity of unique identification numbers from all sensors corresponds to number of events. Some of these events "becomes" targets — it happens when the event has two or more identification numbers from sensors that makes possible event's location with simultaneous generating the location identification number. So, the diagram shows the distribution of identification numbers depending on how many sensors define events one-by-one. It is clear that a portion that is corresponded to a single sensor high-probably identifies the occasional (low, bad, poor, week, strange) signal. It is impossible to locate an event due to "detection" from a single sensor. So, such information could inform the operator that configuration has some week points and it will be good to reduce the portion of identification numbers from one any sensor. The "ID-portions" for two or more sensors are normal for MPSS because of variety of signals on sensors' inputs — the range, terrain features, source of the signal, etc. make probable possibilities for the sensors to detect them. It is obviously that what a concrete combination for two (for three and partly for four) sensors for one-by-one events is could guess the operator some other configuration week points. But the decision needs more high computation for such analysis — the distribution of identification numbers is multiplied (5!+4!+3!=150), so it wasn't executed for this paper.

The fourth diagram of CQs (bottom-right) shows what is a portion of numbers of sensors in event detection. In general, the diagram expresses (6).

The fifth graph material of CQs is a normalized contingency (crosstab) table (see Figs. 10, 14, 15, 16, 19, 20) that shows the relative frequency of detections made by a specific *SP* depending on *SP*s' combination.

$\langle Conf \rangle^I$

| | SP1 | SP2 | SP3 | SP4 | SP5 | SP6 |
|---|---|---|---|---|---|---|
| 1.0 | 1.11 | 0.41 | 1.20 | 2.21 | 3.05 | 2.13 |
| 2.0 | 11.75 | 3.84 | 14.04 | 8.25 | 12.91 | 12.61 |
| 3.0 | 10.85 | 1.51 | 11.91 | 14.74 | 8.86 | 7.71 |
| 4.0 | 10.24 | 1.72 | 13.36 | 12.52 | 13.83 | 9.74 |
| 5.0 | 15.57 | 8.34 | 23.87 | 23.89 | 23.89 | 23.89 |
| 6.0 | 0.39 | 0.39 | 0.39 | 0.39 | 0.39 | 0.39 |

*Fig. 10.* The crosstab of $SP_k$ -detection for Configuration Nr. 1

Let's show how to make the qualitive analysis of the illustrated data for this configuration of MPSS (it is the Conf. Nr 1 above).

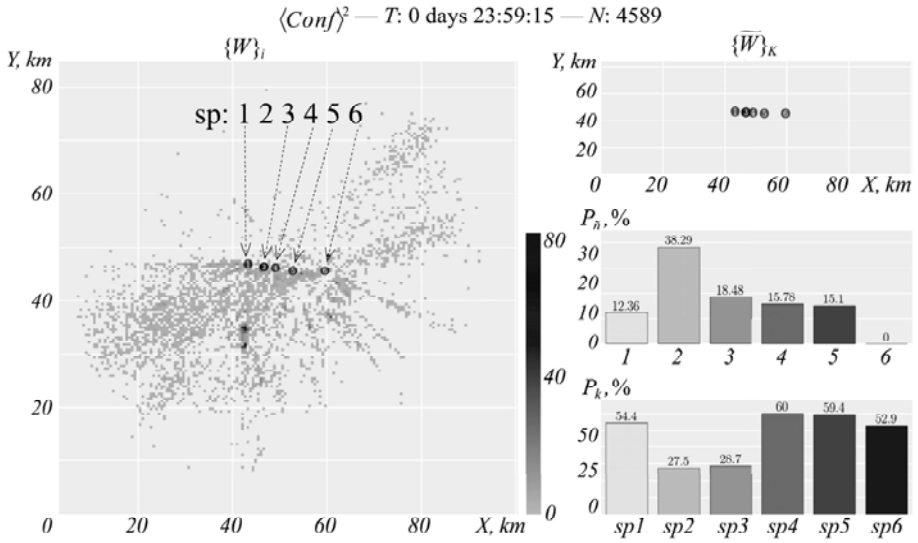



*Fig. 11.* CQs for $\langle Conf \rangle^2$ ($T = d$, dataset size $N = 4589$): $\{W\}_i$, $\{\widetilde{W}\}_K$, $P_{\bar{n}}$, $P_k$

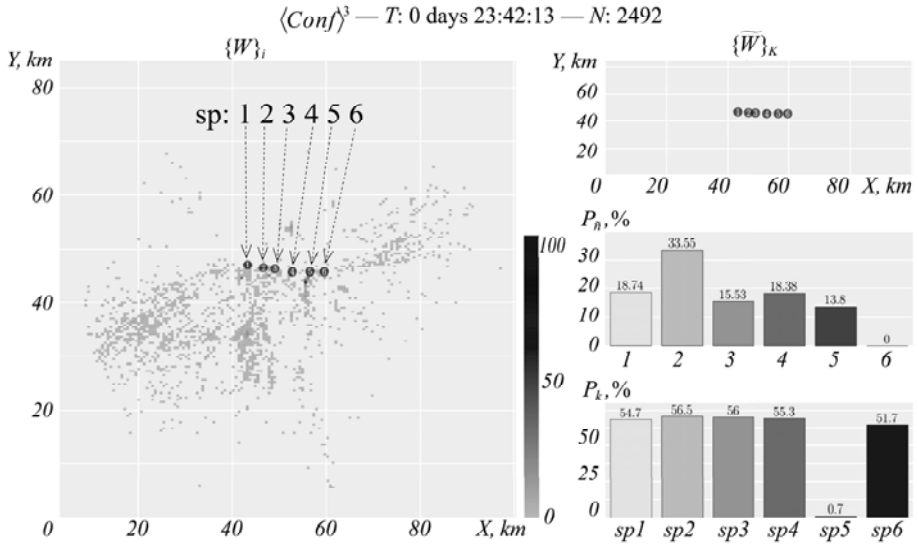



*Fig. 12.* CQs for $\langle Conf \rangle^3$ ($T \approx d$, dataset size $N = 2492$): $\{W\}_i$, $\{\widetilde{W}\}_K$, $P_{\bar{n}}$, $P_k$

Each point on the graph for TESD $\{W\}_i$ shows the sound event that took place during the observation time. It doesn't differ what is a source of the detected sound event — gun fire moment or any kind of explosion caused by a shell, or by a missile, or by a mine, even engine switch-on is possible to detect (it is clear because the SP has a stable determined sensitivity (Fig. 5)). For such circumstances the COP depends on terrain features (relief forms and artificial and natural coverings effect on sound wave propagation). So, making the COP's observation during the determined time period it is possible to get some qualitive analysis — what concrete terrain areas have much more "signals" (they correspond to sound events that were detected in MPSS) and what other terrain areas are "problem-

atic" (there are a smaller number of "signals" then it is expected for typical situation of the initial preposition that there are continuous uniform distributions for signal events — in fact it is necessary to combine the intelligence data with other sources to approve the preposition as it is shown in [17] but for this research it doesn't matter). For the "ideal" efficiency it is demanded to have the COP with no blank terrain areas for the whole attention sector (it corresponds roughly to surveillance area).

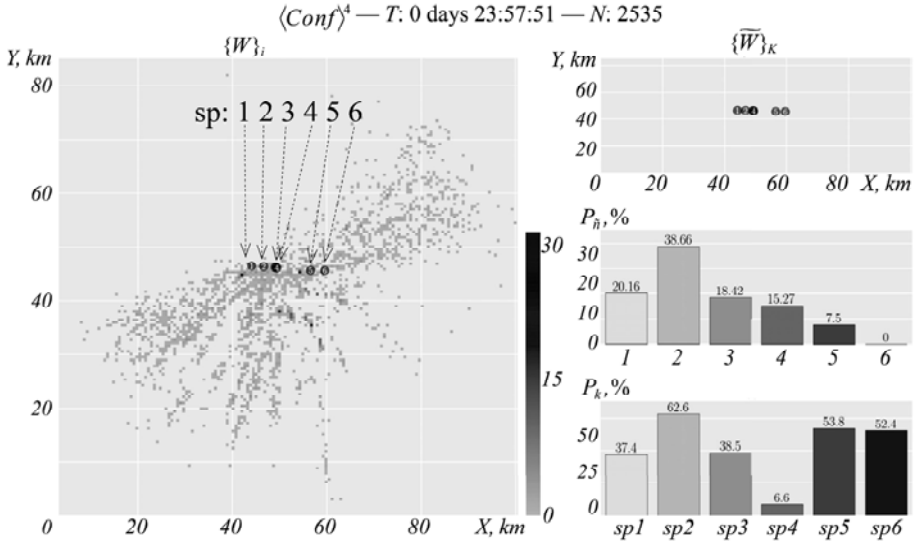



*Fig. 13.* CQs for $\langle Conf \rangle^4$ ($T = d$, dataset size $N = 2535$): $\{W\}_i$, $\{\widetilde{W}\}_K$, $P_{\hat{n}}$, $P_k$

So, for the Fig. 9 it is possible to define some problematic areas (and the directions from the MPSS' configuration center) — they are in the south-east part of the COP (the top-right figure for the ESGM $\{\widetilde{W}\}_K$ shows the general depiction for the COP during the time period observation).

Using the diagram for the SOP $P_k$ (see bottom-right) helps to see the "problematic" SP (SPs). Here for the Conf. Nr 1 the "problematic" SP is the "second" — SP2 with $P_k = 0{,}16$ (16.2%-participation in MPSS' total detection although other SPs give 50% or more results). So, the qualitive analysis for this case shows that it is necessary to reset the SP2 on the terrain — to change its location.

Using the diagram for the LPNP $P_{\hat{n}}$ (see middle-right) allows to identify another "problem" — how often single SP operates. The such qualitive analysis is a base for next statistical and/or technical analysis because all SPs are identical ones and sound wave for typical conditions (for all SPs of MPSS they are similar at least) should be detected by two or more SPs (it depends on distance from the source and angle of the direction of sound wave propagation). So, the qualitive analysis for this case allows operator to define some "blind" directions or other possible reasons that cause less more efficiency for MPSS. The outcome of such analysis should be adjusting of a "problematic" SP (its sensitivity or other organizing measures to stop the detection (to blank) of the unwilling or other parasitic sound wave signals). For this task the normalized contingency table (Fig. 10) should be useful. Here it is presented that SP4 (with 2.2%), SP5 (with 3.1%) and SP6 (with 2.1%) have higher portion of only theirs SOP (without co-

detection). Naturally that the qualitive and statistical analysis should be made for all combinations of different co-detections that took place in the MPSS — such wide analysis of crosstab's data from Fig. 3 will allow to define weak points of the analyzed configuration and to search the ways of MPSS's adjusting or new variant of the configuration. The last approach is presented in the paper furthermore — there are five more variants of MPSS's configuration with CQ's calculated data for each configuration as presented on the Fig. 9 and 10.

Using the crosstab allows to find the SPs with the best "collaborative" features (optimal co-detection). It means that for the detecting of the sound wave and next determination of the sound source's coordinates it is demanded to have the detection from two or more sensors. So, what SPs have the high level of co-detection for two sensors or for three sensors in different combinations it will give the optimal configuration of the MPSS. It is a classic optimization problem for non-optimal terrain accessibility.

So, following CQs are presented for other MPSS's configurations — the figures are grouped correspondingly (see Fig. 11 & Fig. 18 for Conf. Nr. 2, Fig. 12 & Fig. 15 for Conf. Nr. 3, Fig. 13 & Fig. 16 for Conf. Nr. 4, Fig. 17 & Fig. 19 for Conf. Nr. 5 and Fig. 18 & Fig. 20 for Conf. Nr. 6).

⟨*Conf*⟩²

| | SP1 | SP2 | SP3 | SP4 | SP5 | SP6 |
|---|---|---|---|---|---|---|
| 1.0 | 0.96 | 0.33 | 0.31 | 0.94 | 3.38 | 6.45 |
| 2.0 | 15.52 | 7.52 | 7.50 | 14.51 | 17.67 | 13.86 |
| 3.0 | 10.66 | 5.88 | 5.97 | 14.80 | 10.39 | 7.74 |
| 4.0 | 12.20 | 7.21 | 6.41 | 14.62 | 12.88 | 9.78 |
| 5.0 | 15.10 | 6.54 | 8.56 | 15.10 | 15.10 | 15.10 |

*Fig. 14.* The crosstab of $SP_k$-detection for $Conf^2$

⟨*Conf*⟩³

| | SP1 | SP2 | SP3 | SP4 | SP5 | SP6 |
|---|---|---|---|---|---|---|
| 1.0 | 0.40 | 0.36 | 0.28 | 0.88 | 0.00 | 16.81 |
| 2.0 | 15.33 | 15.89 | 11.56 | 14.41 | 0.20 | 9.71 |
| 3.0 | 9.11 | 9.71 | 13.20 | 8.63 | 0.08 | 5.86 |
| 4.0 | 16.05 | 16.77 | 17.13 | 17.62 | 0.12 | 5.82 |
| 5.0 | 13.80 | 13.80 | 13.80 | 13.80 | 0.28 | 13.52 |

*Fig. 15.* The crosstab of $SP_k$-detection for $Conf^3$

⟨*Conf*⟩⁴

| | SP1 | SP2 | SP3 | SP4 | SP5 | SP6 |
|---|---|---|---|---|---|---|
| 1.0 | 0.42 | 5.01 | 3.16 | 1.89 | 4.42 | 4.26 |
| 2.0 | 11.32 | 21.78 | 8.68 | 2.64 | 15.86 | 17.04 |
| 3.0 | 7.46 | 13.93 | 8.72 | 1.03 | 12.70 | 11.44 |
| 4.0 | 9.70 | 14.36 | 10.77 | 0.71 | 13.37 | 12.15 |
| 5.0 | 7.50 | 7.50 | 7.14 | 0.36 | 7.50 | 7.50 |

*Fig. 16.* The crosstab of $SP_k$-detection for $Conf^4$

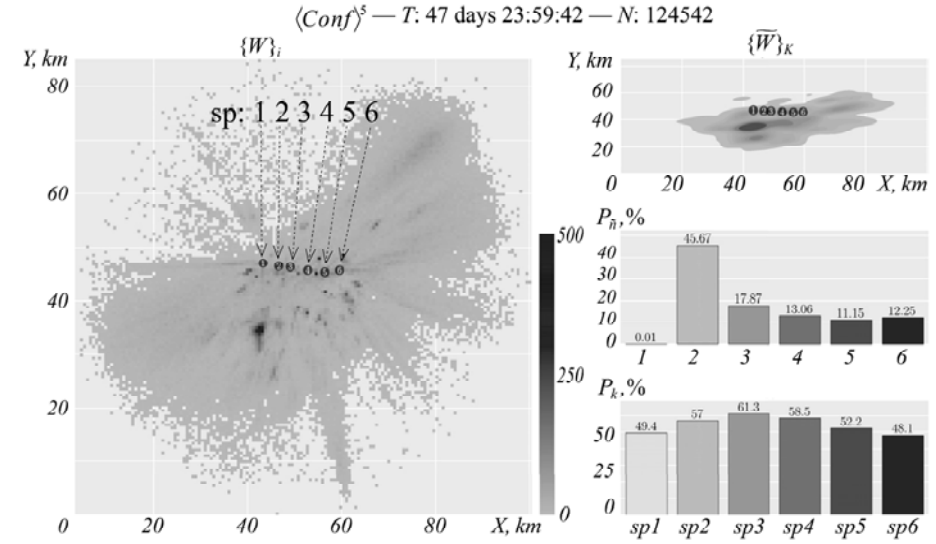



*Fig. 17.* CQs *for* ⟨*Conf*⟩⁵ ($T = 48 \times d$, dataset size $N$=124542): $\{W\}_i$, $\{\widetilde{W}\}_K$, $P_{\hat{n}}$, $P_k$

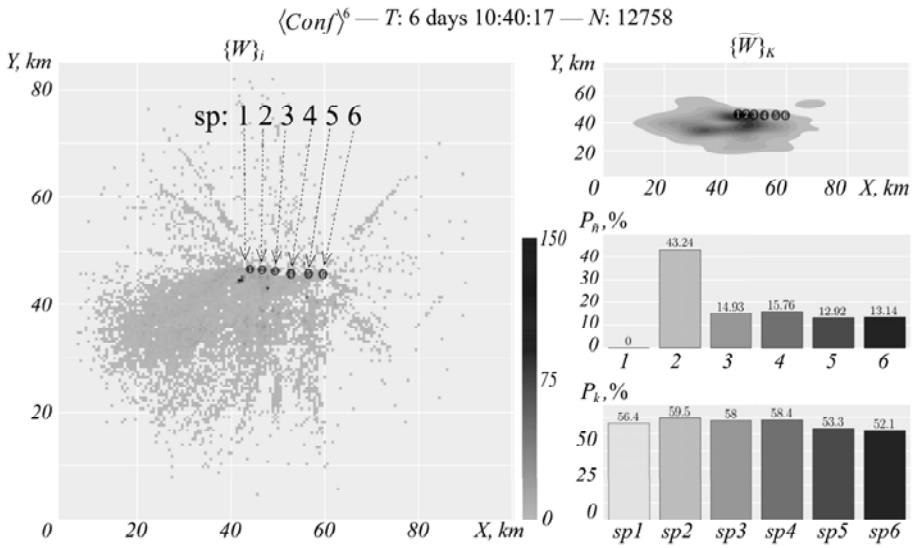



*Fig. 18*. CQs *for* $\langle Conf \rangle^6$ ($T \approx 6 \times d$, dataset size $N$=4589): $\{W\}_i$, $\{\widetilde{W}\}_K$, $P_{\hat{n}}$, $P_k$

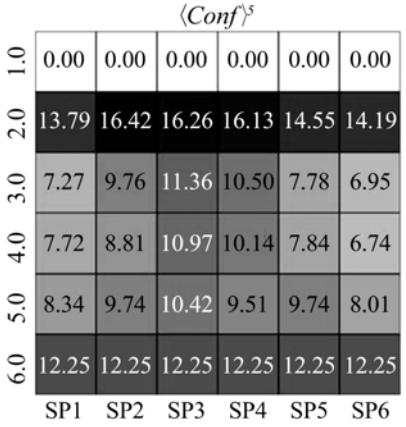



*Fig. 19*. The crosstab of $SP_k$-detection for $\langle Conf \rangle^5$

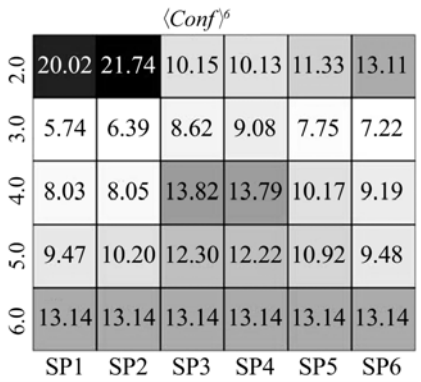



*Fig. 20.* The crosstab of $SP_k$-detection for $\langle Conf \rangle^6$

Having some field configurations (it means they were actual on the terrain) itis possible using some qualitive and statistical analysis of the detection efficiency depending on MPSS's configuration and each or some SP(s) operation to find the best (the optimal for real terrain or other circumstances) configuration for the MPSS. The transforming of the configuration means that some SP(s) are being reset on the terrain (on a new geographic location of the position). Another way of increasing the MPSS efficiency is just the adjusting some features of 'problematic' SP (sensitivity or blank directions (sectors) establishing).

**The mode for MPSS's transiting to achieve better detection possibilities**

The obtained statistics allows to choose the configuration for MPSS for a particular terrain. For instance, using the computations for (5) and (6) the 'weak' points for each configuration — ineffective SP (SPs) — are obtained. The results allowed to build a priorities scheme (an ontology) to determine where the configuration should be moved towards to — see Table 5.

**Table 5.** CQs' parameters for best configuration

| $\langle Conf \rangle^j$ | LPNP | | | | SOP min |
|---|---|---|---|---|---|
| | for $\hat{n} = 2$ | for $\hat{n} = 3$ | for $\hat{n} = 1$ | for $\hat{n} = 6$ | |
| $\langle Conf \rangle^1$ | 31.71 | 18.53 | 10.12 | 0.39 | SP2 |
| $\langle Conf \rangle^2$ | 38.29 | 18.48 | 12.36 | 0 | SP2, SP3 |
| $\langle Conf \rangle^3$ | 31.85 | 18.42 | 18.74 | 0 | SP5 |
| $\langle Conf \rangle^4$ | 38.66 | 15.53 | 20.16 | 12.25 | SP4 |
| $\langle Conf \rangle^5$ | 45.67 | 17.87 | 0 | 0 | - |
| $\langle Conf \rangle^6$ | 43.24 | 14.93 | 0 | 13.14 | - |
| Prioritize | ↑ | ↑ | min | max | none |

The optimal configuration should be one with more portion of two and/or three SPs, with "maximum LPNP" and absent zero or near-zero SOP.

For presented data the optimal configurations were $\langle Conf \rangle^5$ and $\langle Conf \rangle^6$ — approximately the efficiency exceeds on 10…15% (it corresponds to valueless part of single post operating).

## CONCLUSIONS

In the research the variant of additional option in multi-position surveillance system's operating using datasets that are generated in sensors of the system is showed.

One of the key characteristics is a system's configuration which is possible to transit to ensure better efficiency of detection by means of system's sensors. Both the methodology and the ontology for qualitive estimation of different system's configurations are proposed in the paper. The corresponding results are presented using graphic distribution of detected events and diagram for efficiency of system's elements including 6 sensors. The efficiency of MPSS exceeds on 10…15% due to defining the weakest point of multi-position surveillance system and its following transition of system's configuration.

Although the representations with event distributions and SPs' frequencies allow to find the best configuration for MPSS operating, the possibilities of the results are wider. Next step of statistical analysis can be held with respect to the time durations, like estimating average event distributions depending per period, building periodograms and some event dispersions. The ontology and methodology also will be explored in the next researches. In general, the gained results would be useful for system engineering when it's necessary to design both concrete system and system-of-systems including multi-position surveillance system, communication system, GIS, machine learning or deep learning system, etc.

## REFERENCES

1. "Theory and Practice of Control and Systems," *Proc. of the 6th IEEE Mediterranean conf., Alghero, Sardinia, Italy, 9–11 June 1998, A. Tornambe, G. Conte and A.M. Perdon, Eds*, 1999, 864 p.

2. V. Lytvyn, R. Peleshchak, "Directions of using technologies of machine training in the military sphere," *Artificial Intelligence*, 27(1), pp. 161–164, 2022. doi: https://doi.org/10.15407/jai2022.01.161
3. L.Y. Yurchuk, O.V. Rodinin, "Specialized system of collection and treatment of information for the meteorological monitoring," *Visnyk of Vinnytsia Politechnical Institute*, no. 1, pp. 18–22, 2013
4. S. Balovsyak, V. Vasiliev, I. Fodchuk, "Expert System for Supporting the Construction of Three-Dimensional Models of Objects by the Photogrammetry Method," *SISIOT*, vol. 1, no. 2, 02005, 2023. doi: https://doi.org/10.31861/sisiot2023.2.02005
5. C.-E. Lee, J. Baek, J. Son, Y.G. Ha, "Deep AI military staff: cooperative battlefield situation awareness for commander's decision making," *The Journal of Supercomputing*, 79, pp. 6040–6069, 2023. doi: https://doi.org/10.1007/s11227-022-04882-w
6. T. Li, H. Chen, L. Kong, J. Li, Y. Yang, "The development of the combined weapon of light and high maneuvering artillery from the view of the warfare object," in *S. Long, B.S. Dhillon (Eds.) "Man-machine-environment system engineering," Proc. of the 23rd Int. Conf. on MMESE, 2023, Lecture Notes in Electrical Engineering*, 1069, pp. 303–308, 2023. doi: https://doi.org/10.1007/978-981-99-4882-6_43
7. S. Faraj, S. Pachidi, K. Sayegh, "Working and organizing in the age of the learning algorithm," *Information and Organization*, vol. 28(1), pp. 62–70, 2018. doi: https://doi.org/10.1016/j.infoandorg.2018.02.005
8. A. Dagallier et al.,"Long-range acoustic localization of artillery shots using distributed synchronous acoustic sensors," *The Journal of the Acoustical Society of America*, pp. 4860–4872, 2019. doi: https://doi.org/10.1121/1.5138927
9. S. Cheinet, T. Broglin, "Sensitivity of shot detection and localization to environmental propagation," *Applied Acoustics*, 93, pp. 97–105, 2015. doi: https://doi.org/10.1016/j.apacoust.2015.01.021
10. V. Tymchuk, O. Mediakov, A. Poliakov, O. Popov, "The research of the configurations in some locating acoustic system for geospatial modeling in GIS to increase the coordinate accuracy," *Int. Conf. of Young Professionals "GeoTerrace-2023", Lviv Polytecnic National University, 2–4 October 2023*. doi: https://doi.org/10.3997/2214-4609.2023510064
11. "HALO Hostile Artillery Locating System," *Leonardo Ltd.*, 2022. Available: https://electronics.leonardo.com/en/products/halo
12. R. Kochan, B.Trembach, O. Kochan, "Methodical error of targets bearing by sound artillery intelligence system," *Measuring Equipment and Metrology*, 80(3), pp. 10–14, 2019. doi: https://doi.org/10.23939/istcmtm2019.03.010
13. A. Kowalska-Styczen, R. Peleshchak, V. Lytvyn, I. Peleshchak, A. Dyriv, V. Danylyk, "Automatic Identification of Sound Source Position Coordinates Using a Sound Metric System of Sensors Linked with an Internet Connection," *Symmetry*, 14(11), 2338, 2022. doi: https://doi.org/10.3390/sym14112338
14. S. Kapka, M. Lewandowski, "Sound source detection, localization and classification using consecutive ensemble of CRNN models," *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, pp. 119–123, 2019. doi: https://doi.org/10.33682/9f2t-ab23
15. S. Artemuk, I. Mykytyn, "System for determining the coordinates of the acoustic signal source based on the difference-time method and machine learning methods," *Measuring and Computing Devices in Technological Processes*, no. 3, 2023. doi: https://doi.org/10.31891/2219-9365-2023-75-1
16. V. Tymchuk, "Artillery radars of TPQ series: some aspects of design and operation, lessons of modification," *Artillery and Rifle Armaments*, 1(38), pp. 12–18, 2011. Available: http://www.irbis-nbuv.gov.ua/publ/REF-0000534126
17. J. Safari Bazargani, A. Sadeghi-Niaraki, S.-M. Choi, "A Survey of GIS and IoT Integration: Applications and Architecture," *Appl. Sci.* 2021, 11, 10365. doi: https://doi.org/10.3390/app112110365

18. T.-Y. Ho et al., "The dawn of AI-native EDA: promises and challenges of large circuit models," *arxiv.org*, 12 March 2024. doi: https://doi.org/10.48550/arXiv.2403.07257

*Received 02.05.2024*

## INFORMATION ON THE ARTICLE

**Volodymyr Yu. Tymchuk,** ORCID: 0000-0002-3549-2813, Hetman Petro Sahaidachnyi National Army Academy, Ukraine, e-mail: v_tymchuk@yahoo.co.uk

**Oleksandr O. Mediakov,** ORCID: 0000-0002-2580-3155, Lviv Polytechnic National University, Ukraine, e-mail: oleksandr.mediakov@gmail.com

**Oleksandr O. Popov,** ORCID: 0009-0002-7548-6175, Armed Forces of Ukraine, A1108 military unit, Ukraine, e-mail: vparrvia@gmail.com

**Taras V. Trysnyuk,** ORCID: 0000-0002-3672-8242, Institute of Telecommunications and Global Information Space National Academy of Science of Ukraine, Ukraine, e-mail: taras24t@gmail.com

**Serhii A. Tsybulia,** ORCID: 0000-0003-0323-1771, National Defense University of Ukraine, Ukraine, e-mail: kibtor@gmail.com

**ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ БАГАТОПОЗИЦІЙНОЇ СИСТЕМИ СПОСТЕРЕЖЕННЯ НА ОСНОВІ ДОДАТКОВОГО ОБРОБЛЕННЯ ДАНИХ ВІД СЕНСОРІВ ЗІ ЗМІНЮВАНИМ МІСЦЕПОЛОЖЕННЯМ** / В.Ю. Тимчук, О.О. Медяков, О.О. Попов, Т.В. Триснюк, С.А. Цибуля

**Анотація.** Ефективність багатопозиційної системи залежить від її структури, а саме від кількості елементів (сенсорів) у складі системи та від місцеположення самих сенсорів, які визначають конфігурацію системи, а також від того, як на роботу системи впливають середовище і рельєф місцевості. Розглянуто оброблення даних у багатопозиційній системі спостереження як додаткової операції шляхом збирання та обчислювальних маніпуляцій із великими даними (набором датасетів) з елементів системи. Значний обсяг цифрових даних, що циркулюють у багатопозиційній системі (отримуються від сенсорів системи), уможливлює використання статистичних методів оброблення даних. Запропоновано онтологію для оцінювання якості багатопозиційної системи залежно від її конфігурацій. Результати розподілів виявлених подій подано у графічних формах, що дало змогу виконати статистичну оцінку розподілених даних. Зроблено висновки про спосіб поліпшувати ефективність багатопозиційної системи в непередбачуваному змінному середовищі завдяки зміні конфігурації системи, за якої забезпечують кращі показники якості виявлення.

**Ключові слова:** оброблення даних, спостереження, виявлення, багатопозиційна система, система систем, ефективність.

UDC 519.87, 519.17
DOI: 10.20535/SRIT.2308-8893.2025.4.03

# MATRIX-GRAPHIC SIMULATION OF SOCIAL NETWORK: ERGODIC PROPERTIES

**I.Ya. SPECTORSKY, V.M. STATKEVYCH, O.V. STUS**

**Abstract.** We propose mathematical tools for a social network simulation in order to obtain some sufficient conditions of the network's ergodicity, that is, the existence of a steady state as $t \to +\infty$. The proposed model is linear; the elements of the network form a two-dimensional array (i.e., a matrix) $\Omega = \{1,2,\ldots,n\} \times \{1,2,\ldots,m\}$, where $A_{i,j}(t) \in [0,1]$ is the state of the element $(i,j) \in \Omega$, $t \geq 0$ is time. An impact operator $T$ is a four-dimensional array; the element $T_{i,j,k,l} \geq 0$ denotes the impact of the element $(i,j)$ on the element $(k,l)$: $(TA)_{i,j} = \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i,j,k,l} A_{k,l}$. The impact operator $T$ is also presented as a directed graph $G_T$, whose vertices correspond to elements $(i,j) \in \Omega$: a directed edge (an arc) leads from the vertex $(k,l) \in \Omega$ to the vertex $(i,j) \in \Omega$ if and only if $T_{i,j,k,l} > 0$, and this edge is labelled by the number $T_{i,j,k,l}$. A bound $B \subset \Omega$ is introduced in such a way that $T_{i,j,k,l} = 0$ for $(k,l) \in \Omega$, $(i,j) \in B$. The state $A(t+1)$ at time $t+1$ is defined by the state $A(t)$ at the current time $t \geq 0$ via equation $A(t+1) = TA(t) + \Delta$, where matrix $\Delta$ of dimension $n \times m$ defines the states of bound elements $(i,j) \in B$; $\Delta_{i,j} = 0$ for internal elements $(i,j) \in \Omega \setminus B$. Some sufficient conditions for the network's ergodicity are given in the form of connectivity properties of the impact graph $G_T$. This graph must contain paths between all pairs of vertices and loops for all vertices. Suggested conditions provide the spectrum of $T$ (with the possible exception of $\lambda = 1$) to be located inside the open unit disk; we prove that $\lambda = 1$ is an eigenvalue of $T$ if and only if the bound $B \subset \Omega$ is isolated (no bound element impacts any internal one). These spectral properties of $T$ provide that the steady state exists and can be found by the iterative procedure $A(t+1) = TA(t) + \Delta$ with the given $A(0)$; the iterative process converges linearly (geometrically).

**Keywords:** social system, simulation, ergodicity, eigenvalue, Jordan normal form.

## INTRODUCTION

Social network analysis is currently one of the most important methods for scientific investigation in sociology, social psychology and other areas (see, e. g., [1; 2]). A social network is defined by the interaction of network elements, or, in other words, by impact of network elements on other ones.

Various toolkits can be used to simulate a social network. For example, in [3; 4] graph theory methods are used to visualize network elements interaction, in [1] matrix analysis gives a more convenient way to analyze network elements interaction.

Usually, a social network is not a static structure, i.e. the state of network elements changes over time. The social network' behaviour is currently being intensively investigated (see, e.g., [3; 5; 6]), and steady states are of a special interest (see, e.g., [3]).

**The purpose of this work** is to obtain some sufficient conditions for social network ergodicity (independence of a network's behaviour from initial conditions in extremely distant time), using matrix and graph methods of social network representation.

## REPRESENTATION OF A SOCIAL NETWORK AND ITS DYNAMICS

Suppose that the social network (hereinafter referred to as the network) contains $nm$ elements, arranged in $n$ rows and $m$ columns ($n, m \in \mathbb{N}$), i.e. position of each element is defined by a pair $(i, j) \in \Omega$, where $\Omega = \{1,2,\ldots,n\} \times \{1,2,\ldots,m\} = \{(i, j) : 1 \le i \le n,\ 1 \le j \le m\}$ is the set (area) of coordinates of network elements.

The current state of the element $(i, j)$ is defined by a number $A_{i,j}(t) \in [0,1]$, which can be particularly treated as an attitude of the element $(i, j)$ towards some problem arisen in the network ($0$ means completely negative attitude, $1$ means completely positive one); hereinafter $t \in \mathbb{N}_0$ denotes discrete time ($\mathbb{N}_0 = \mathbb{N} \cup \{0\}$). Therefore, the current state of the network can be represented as a matrix $A(t) \in M_{n\times m}[0,1]$ of dimension $n \times m$ with elements $A_{i,j}(t) \in [0,1]$ ($(i, j) \in \Omega$).

Let $B \subset \Omega$ be the bound of the coordinate area $\Omega$. The states of boundary elements are described by the boundary condition matrix $\Delta$ of dimension $n \times m$, assuming $\Delta_{i,j} = 0$ for all $(i, j) \in \Omega \setminus B$. Elements $(i, j) \in \Omega \setminus B$ that do not belong to $B$ are called internal. Hereinafter, assume that $B \ne \Omega$ (excluding a trivial case $B = \Omega$).

In order to simulate network dynamics, introduce a linear impact operator $T : M_{n\times m}[\mathbb{R}] \to M_{n\times m}[\mathbb{R}]$, where $M_{n\times m}[\mathbb{R}]$ denotes a linear space of $n \times m$ matrices with elements from $\mathbb{R}$. The operator $T$ is considered as 4-dimensional $n \times m \times n \times m$ array with elements from $\mathbb{R}$, its action on a matrix $X \in M_{n\times m}[\mathbb{R}]$ is defined point-wise:

$$(TX)_{i,j} = \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i,j,k,l} X_{k,l} , \qquad (1)$$

the element $T_{i,j,k,l}$ ($(i, j, k, l) \in \{1,2,\ldots,n\} \times \{1,2,\ldots,m\} \times \{1,2,\ldots,n\} \times \{1,2,\ldots,m\}$) defines impact of the state of the network element $(k, l)$ on the state of the network element $(i, j)$. The following conditions are assumed for normalization reasons:

$$\forall (i, j, k, l) \in \{1,2,\ldots,n\} \times \{1,2,\ldots,m\} \times \{1,2,\ldots,n\} \times \{1,2,\ldots,m\} : T_{i,j,k,l} \ge 0 ; \quad (2)$$

$$\forall (i, j) \in \Omega \setminus B : \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i,j,k,l} = 1 . \qquad (3)$$

Since the states of elements on the bound $B$ are defined by matrix $\Delta$, assume that

$$\forall (i, j) \in B \ \forall (k, l) \in \Omega : T_{i,j,k,l} = 0 . \qquad (4)$$

Given relation (1), this means that $(TX)_{i,j} = 0$ for all $(i, j) \in B$.

Assume that the network state $A(t+1)$ at time $t+1$ ($t \geq 0$) is defined by the network state $A(t)$ at time $t \geq 0$ according to the equation

$$A(t+1) = TA(t) + \Delta, \tag{5}$$

the network state $A(0)$ at the initial time $t = 0$ is assumed to be defined by the initial condition matrix $A(0)$ of dimension $n \times m$ with elements $(A(0))_{i,j} = A_{i,j}(0) \in [0,1]$ ($(i, j) \in \Omega$). Note that the summand $TA(t)$ in relation (5) defines the states of internal elements $(i, j) \in \Omega \setminus B$, the summand $\Delta$ defines the states of elements $(i, j) \in B$ on the bound $B$.

The correspondence of the initial condition $A(0)$ with the boundary condition $\Delta$ (on the bound $B$) requires assumption

$$(A(0))_{i,j} = \Delta_{i,j} \quad \text{for all} \quad (i, j) \in B. \tag{6}$$

**Lemma 1.** Let $X$ be an arbitrary $n \times m$ matrix with elements from $[0,1]$ ($X \in M_{n \times m}[0,1]$). Then $TX \in M_{n \times m}[0,1]$, i.e. the matrix set $M_{n \times m}[0,1]$ is closed with respect to the operator $T$.

**Proof.** Equation (1) immediately implies nonnegativity of elements $(TX)_{i,j} \in \mathbb{R}$ ($(i, j) \in \Omega$). For upper bounding $(TX)_{i,j} \in \mathbb{R}$ apply relation (1) given condition (3):

$$(TX)_{i,j} = \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i,j,k,l} X_{k,l} \leq \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i,j,k,l} = 1,$$

which proves the statement of the lemma. □

The operator $T$ can be visualized as a labelled directed impact graph $G_T$ with vertices corresponding to elements $(i, j) \in \Omega$: a directed edge leads from the vertex $(k,l) \in \Omega$ to the vertex $(i, j) \in \Omega$ if and only if $T_{i,j,k,l} > 0$, this edge is labelled by the number $T_{i,j,k,l}$. Note that the operator $T$ in fact defines adjacency matrix $G_T$, deployed in 4-dimensional array for convenience.

**Example 1.** Consider the network on the coordinate area $\Omega = \{1,2,\ldots,n\} \times \times \{1,2,\ldots,m\}$ with the bound $B = \{(1, j),(n, j),(i,1),(i,m) : 1 \leq i \leq n, 1 \leq j \leq m\}$, the impact operator $T$ simulates equal impact on each internal element by its 4 neighbours:

$$T_{i,j,i,j} = \alpha \quad \text{for} \quad 2 \leq i \leq n-1 \quad \text{and} \quad 2 \leq j \leq m-1;$$

$$T_{i,j,i-1,j} = T_{i,j,i+1,j} = T_{i,j,i,j-1} = T_{i,j,i,j+1} = 0.25(1-\alpha)$$

$$\text{for} \quad 2 \leq i \leq n-1 \quad \text{and} \quad 2 \leq j \leq m-1;$$

$$T_{i,j,k,l} = 0 \quad \text{for} \quad |i-k| + |j-l| \geq 2; \quad T_{i,j,k,l} = 0 \quad \text{for} \quad i \in \{1,n\} \quad \text{or} \quad j \in \{1,m\}.$$

Here a constant $\alpha \in (0,1)$ defines the impact value on the element by its 4 neighbours and by itself. The corresponding impact graph $G_T$ for a case of $n = 5$, $m = 6$, $\alpha = 0.6$ is depicted in Fig. 1, vertices of boundary elements are denoted by ($\circ$).

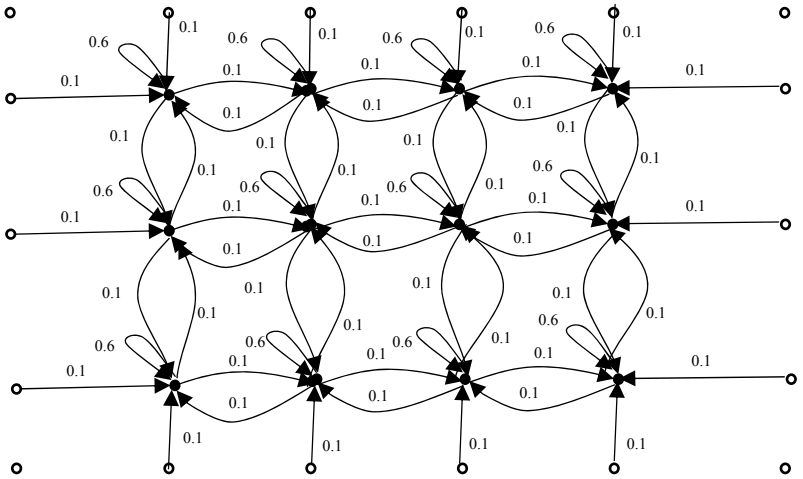


*Fig. 1*

**Remark 1**. Relations (2) and (3) in the case of $B = \varnothing$ define linear operators with stochastic (Markov) matrices (see, e.g., [7; 8]), and some its properties can be extended on a more general matrix types (see, e.g., [9]).

## SPECTRUM OF THE OPERATOR $T$

It is well known that, although $T : M_{n\times m}[\mathbb{R}] \to M_{n\times m}[\mathbb{R}]$ is the linear operator on the linear space $M_{n\times m}[\mathbb{R}]$ (i.e. on the field of real numbers), eigenvalues and eigenvectors of the operator $T$ in a general case are complex. Hereinafter, in the context of the operator $T$, usual notion 'eigenvector' is used (despite that the argument of the operator $T$ is matrix $X \in M_{n\times m}[\mathbb{R}]$).

**Lemma 2.** Let $\lambda \in \mathbb{C}$ be an eigenvalue of the operator $T$, i.e. $TE = \lambda E$ for some nonzero $E \in M_{n\times m}[\mathbb{C}]$. Then $|\lambda| \le 1$.

**Proof.** Let $\left|E_{i_0,j_0}\right| = \max\limits_{(i,j)\in\Omega}\left|E_{i,j}\right|$, i.e. $\left|E_{i,j}\right|$ reaches its maximal value on $(i_0, j_0) \in \Omega$ (obviously, this maximum can be reached at several points). Then, similarly to the proof of Lemma 1, one can obtain:

$$\left|(TE)_{i_0,j_0}\right| = \left|\sum_{k=1}^{n}\sum_{l=1}^{m} T_{i_0,j_0,k,l}E_{k,l}\right| \le \sum_{k=1}^{n}\sum_{l=1}^{m} T_{i_0,j_0,k,l}\left|E_{k,l}\right| \le \left|E_{i_0,j_0}\right|\sum_{k=1}^{n}\sum_{l=1}^{m} T_{i_0,j_0,k,l} = \left|E_{i_0,j_0}\right|.$$

However, on the other hand, $(TE)_{i_0,j_0} = \lambda E_{i_0,j_0}$, thus $|\lambda|\cdot\left|E_{i_0,j_0}\right| \le \left|E_{i_0,j_0}\right|$. Therefore, since $\left|E_{i_0,j_0}\right| = \max\limits_{(i,j)\in\Omega}\left|E_{i,j}\right| \ne 0$, it yields the desired estimate $|\lambda| \le 1$. □

**Theorem 1.** Let $\lambda \in \mathbb{C}$ be an eigenvalue of the operator $T$ that belongs to the unit circle ($|\lambda| = 1$), and let the impact graph $G_T$ satisfy the following conditions:

- for any internal elements $(i_1, j_1), (i_2, j_2) \in \Omega \setminus B$ there exists a directed path from the vertex $(i_1, j_1)$ to $(i_2, j_2)$;
- for any internal element $(i, j) \in \Omega \setminus B$ there exists a 'loop' (an edge leading from the vertex $(i, j)$ to the same vertex $(i, j)$).

Then $\lambda = 1$, and the corresponding eigenspace is generated by the eigenvector $1^{\Omega \setminus B} \in M_{n \times m}[\mathbb{C}]$ such that

$$\left(1^{\Omega \setminus B}\right)_{i,j} = \begin{cases} 1, (i,j) \in \Omega \setminus B, \\ 0, (i,j) \in B. \end{cases}$$

**Proof.** Let $E \in M_{n \times m}[\mathbb{C}]$ be some eigenvector that corresponds to the eigenvalue $\lambda \in \mathbb{C}$ on the unit circle ($|\lambda| = 1$). Note that $E_{i,j} = 0$ for all $(i,j) \in B$ due to relation $TE = \lambda E$.

Firstly prove that $|E_{i,j}|$ is a constant that does not depend on $(i,j) \in \Omega \setminus B$. Let $c = |E_{i_0,j_0}| = \max\limits_{(i,j) \in \Omega} |E_{i,j}|$, i.e. $|E_{i,j}|$ reaches its maximal value on $(i_0, j_0) \in \Omega$. Since any eigenvector is nonzero by definition, constant $c = |E_{i_0,j_0}| = \max\limits_{(i,j) \in \Omega} |E_{i,j}|$ is positive, thus $(i_0, j_0) \in \Omega \setminus B$. Using relation (1), one can obtain:

$$\lambda E_{i_0,j_0} = (TE)_{i_0,j_0} = \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i_0,j_0,k,l} E_{k,l} \,. \tag{7}$$

Given normalizing conditions (2) and (3), equality (7) implies

$$|\lambda| \cdot |E_{i_0,j_0}| = |E_{i_0,j_0}| \le \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i_0,j_0,k,l} |E_{k,l}| \le |E_{i_0,j_0}| \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i_0,j_0,k,l} = |E_{i_0,j_0}| .$$

Therefore, the triangle inequality

$|E_{i_0,j_0}| = \left| \sum\limits_{k=1}^{n} \sum\limits_{l=1}^{m} T_{i_0,j_0,k,l} E_{k,l} \right| \le \sum\limits_{k=1}^{n} \sum\limits_{l=1}^{m} T_{i_0,j_0,k,l} |E_{k,l}|$ turns into equality:

$$|E_{i_0,j_0}| = \left| \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i_0,j_0,k,l} E_{k,l} \right| = \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i_0,j_0,k,l} |E_{k,l}| , \tag{8}$$

which is possible if and only if $|E_{k,l}| = |E_{i_0,j_0}|$ for all $(k,l) \in \Omega \setminus B$ such that $T_{i_0,j_0,k,l} > 0$. Further, recall that for nonzero $z_1, z_2 \in \mathbb{C}$ the equality $|z_1 + z_2| = |z_1| + |z_2|$ holds if and only if their arguments are equal: $\arg z_1 = \arg z_2$. So, relation (7) implies that $E_{k,l} = \lambda E_{i_0,j_0}$ for all $(k,l) \in \mathcal{E}^1(i_0, j_0)$, where the set $\mathcal{E}^1(i_0, j_0) = \{(k,l) : T_{i_0,j_0,k,l} > 0\}$ contains all elements $(k,l) \in \Omega \setminus B$ that directly impact the element $(i_0, j_0)$ (there exists an edge from $(k,l)$ to $(i_0, j_0)$). Therefore, since $(i_0, j_0) \in \mathcal{E}^1(i_0, j_0)$ (by the theorem conditions, there is the loop for the element $(i_0, j_0) \in \Omega \setminus B$), it is easy to see that $\lambda = 1$, and $E_{k,l} = E_{i_0,j_0}$ for all $(k,l) \in \mathcal{E}^1(i_0, j_0)$. Repeating these considerations for each $(\widetilde{k}, \widetilde{l}) \in \mathcal{E}^1(k,l)$, $(k,l) \in \mathcal{E}^1(i_0, j_0)$, one can obtain that $(\widetilde{k}, \widetilde{l}) \in \mathcal{E}(i_0, j_0)$, where the set $\mathcal{E}(i_0, j_0)$ contains all elements $(k,l) \in \Omega \setminus B$ that directly or indirectly impact the element $(i_0, j_0)$ (there exists a path from $(k,l)$ to $(i_0, j_0)$). Finally, the theorem conditions provide that for any $(i_1, j_1), (i_2, j_2) \in \Omega \setminus B$ there exists a directed path from

$(i_1, j_1)$ to $(i_2, j_2)$, thus $\mathcal{E}(i_0, j_0) = \Omega \setminus B$, $E = E_{i_0, j_0} \cdot 1^{\Omega \setminus B}$, which completes the proof of the theorem.□

**Corollary.** Under the conditions of Theorem 1, the eigenvalue $\lambda = 1$ of the operator $T$ is a simple one (a single root of its characteristic polynomial).

**Proof.** Assume that the eigenvalue $\lambda = 1$ is not simple, i.e. it is a root of the characteristic polynomial of multiplicity 2 or more. Then, by Theorem 1, the eigenvalue $\lambda = 1$ corresponds to a one-dimensional eigenspace, and for $\lambda = 1$ there exists (see, e.g., [10, 11]) a generalized eigenvector $\widetilde{1}^{\Omega \setminus B} \in M_{n \times m}[\mathbb{C}]$: $T \cdot \widetilde{1}^{\Omega \setminus B} = \lambda \cdot \widetilde{1}^{\Omega \setminus B} + 1^{\Omega \setminus B} = \widetilde{1}^{\Omega \setminus B} + 1^{\Omega \setminus B}$. So, for an arbitrary $t \in \mathbb{N}$ one can obtain: $T^t \cdot \widetilde{1}^{\Omega \setminus B} = \widetilde{1}^{\Omega \setminus B} + t \cdot 1^{\Omega \setminus B}$, which contradicts to Lemma 1 for sufficiently large $t \in \mathbb{N}$. This contradiction proves that the eigenvalue $\lambda = 1$ is indeed simple. □

Theorem 1 proves that, under the given conditions, the unit circle can contain at most one eigenvalue of the operator $T$, namely $\lambda = 1$. However, the theorem does not exclude the case when the unit circle does not contain any eigenvalue of the operator $T$ (by Lemma 2, in this case all eigenvalues of $T$ are located inside the open unit disk). It is easy to derive from the proof of Theorem 1 that $\lambda = 1$ is an eigenvalue of the operator $T$ if and only if $\sum_{(k,l) \notin \Omega \setminus B} T_{i,j,k,l} = 1$ for all $(i, j) \in \Omega \setminus B$, which, given conditions (2) and (3), is equivalent to the following condition:

$$\forall (i, j) \in \Omega \setminus B \; \forall (k, l) \in B : T_{i,j,k,l} = 0 . \qquad (9)$$

Obviously, condition (9) means that the network bound is isolated from the rest of the network: no element $(k,l) \in B$ can impact any element $(i, j) \in \Omega \setminus B$. If condition (9) does not hold, there is at least one element $(k,l) \in B$ that impacts some element $(i, j) \in \Omega \setminus B$.

To simplify further analysis, consider a block structure for matrices on $\Omega$ with respect to the partition $\Omega = (\Omega \setminus B) \cup B$. For an arbitrary matrix $A \in M_{n \times m}[\mathbb{R}]$ consider a block $A_{\Omega \setminus B}$ of elements from $\Omega \setminus B$. Although the rectangle structure for area $\Omega \setminus B$ may be distorted (see, e.g., Fig. 3), one can treat $M_{\Omega \setminus B}[\mathbb{R}]$ (as well as $M_{\Omega \setminus B}[\mathbb{C}]$ if necessary) as a linear space of real (complex) 'vectors', whose entries are numbered by coordinate pairs $(i, j) \in \Omega \setminus B$. So, for the network in Fig. 3, one can obtain the following block $A_{\Omega \setminus B} \in M_{\Omega \setminus B}[\mathbb{R}]$ (vertices of boundary elements are denoted by «∘»):

$$A_{\Omega \setminus B} = \begin{matrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} \\ A_{2,1} & \circ & \circ & A_{2,4} \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} \end{matrix}$$

Similarly, (provided $B \neq \varnothing$) consider the linear space $M_B[\mathbb{R}]$ and the block $A_B \in M_B[\mathbb{R}]$.

Further, define $T_{\Omega \setminus B, \Omega \setminus B} : M_{\Omega \setminus B}[\mathbb{R}] \to M_{\Omega \setminus B}[\mathbb{R}]$ as a linear operator on the block space $M_{\Omega \setminus B}[\mathbb{R}]$:

$$\forall (i,j) \in \Omega \setminus B \; \forall (k,l) \in \Omega \setminus B : (T_{\Omega\setminus B,\Omega\setminus B})_{i,j,k,l} = T_{i,j,k,l};$$
$$(T_{\Omega\setminus B,\Omega\setminus B} A_{\Omega\setminus B})_{i,j} = \sum_{(k,l)\in\Omega\setminus B} (T_{\Omega\setminus B,\Omega\setminus B})_{i,j,k,l} (A_{\Omega\setminus B})_{k,l}.$$

Similarly (provided $B \neq \varnothing$) one can define a linear operator $T_{\Omega\setminus B,B}$: $M_B[\mathbb{R}] \to M_{\Omega\setminus B}[\mathbb{R}]$:

$$\forall (i,j) \in \Omega \setminus B \; \forall (k,l) \in B : (T_{\Omega\setminus B,B})_{i,j,k,l} = T_{i,j,k,l};$$
$$(T_{\Omega\setminus B,B} A_B)_{i,j} = \sum_{(k,l)\in B} (T_{\Omega\setminus B,B})_{i,j,k,l} (A_B)_{k,l}.$$

Note that one can in a similar way define linear operators $T_{B,\Omega\setminus B}$: $M_{\Omega\setminus B}[\mathbb{R}] \to M_B[\mathbb{R}]$ and $T_{B,B} : M_B[\mathbb{R}] \to M_B[\mathbb{R}]$; however, according to definition of the bound $B$ (condition (4)), these linear operators are zero.

Now equation (5) can be rewritten as a system:

$$\begin{cases} A_{\Omega\setminus B}(t+1) = T_{\Omega\setminus B,\Omega\setminus B} A_{\Omega\setminus B}(t) + T_{\Omega\setminus B,B} A_B(t) + \Delta_{\Omega\setminus B}; \\ A_B(t+1) = \Delta_B. \end{cases}$$

Note that, by definition of the network bound, $\Delta_{i,j} = 0$ in equation (5) for $(i,j) \in \Omega \setminus B$, that is $\Delta_{\Omega\setminus B} = 0$, so (for $t \geq 0$)

$$\begin{cases} A_{\Omega\setminus B}(t+1) = T_{\Omega\setminus B,\Omega\setminus B} A_{\Omega\setminus B}(t) + T_{\Omega\setminus B,B} A_B(t); \\ A_B(t+1) = \Delta_B. \end{cases}$$

Further, the second equation implies $A_B(t) = \Delta_B$ for all $t \geq 1$, and for $t = 0$ the initial condition corresponds to the boundary one by relation (6), so the obtained system can be written for any $t \geq 1$ as

$$\begin{cases} A_{\Omega\setminus B}(t+1) = T_{\Omega\setminus B,\Omega\setminus B} A_{\Omega\setminus B}(t) + T_{\Omega\setminus B,B} \Delta_B; \\ A_B(t+1) = \Delta_B, \end{cases} \tag{10}$$

where $A_B(0)$ is defined by the initial conditions.

Consider system (10) in two cases: when condition (9) holds (isolated bound) and when it does not hold (non-isolated bound).

**A.** Suppose that condition (9) holds. Along with condition (4) it means that in the impact graph $G_T$ all boundary elements are isolated (see, e.g., Fig. 2 and 3), vertices of boundary elements are denoted by «∘».

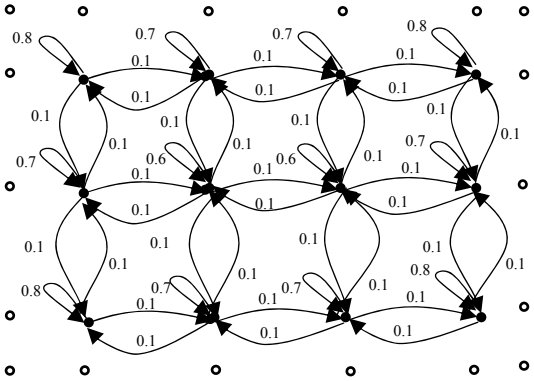


*Fig. 2*

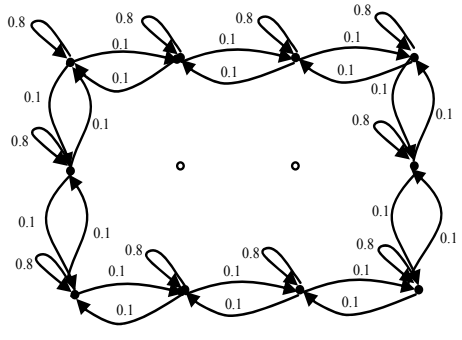


*Fig. 3*

In the case of isolated bound, the structure of the operator $T$ is similar to a block-diagonal matrix: $T_{i,j,k,l}=0$ if either $(i,j)\in\Omega\setminus B$ and $(k,l)\in B$, or $(i,j)\in B$ and $(k,l)\in\Omega\setminus B$. The operator $T_{\Omega\setminus B,B}$ is zero in the case of isolated bound, thus system (10) for $t\geq 0$ takes a form

$$\begin{cases} A_{\Omega\setminus B}(t+1)=T_{\Omega\setminus B,\Omega\setminus B}A_{\Omega\setminus B}(t); \\ A_B(t+1)=\Delta_B. \end{cases} \tag{11}$$

Given normalizing conditions (2) and (3), the linear operator $T_{\Omega\setminus B,\Omega\setminus B}$ can be treated as an operator with stochastic matrix, which has the eigenvalue $\lambda=1$ with the corresponding eigenvector $1_{\Omega\setminus B}\in M_{\Omega\setminus B}[\mathbb{C}]$: $(1_{\Omega\setminus B})_{i,j}=1$ for all $(i,j)\in\Omega\setminus B$.

**B.** Suppose that condition (9) does not hold. It means that in the impact graph $G_T$ at least one boundary element is not isolated (see Fig. 1 in Example 1).

According to condition (4), $T_{i,j,k,l}=0$ for any $(i,j)\in B$ and $(k,l)\in\Omega$, so for the operator $T$ it is worth considering a block structure similar to one considered in case A; in the case of non-isolated bound this structure of course is not block-diagonal.

In the case of non-isolated bound equation (5) can also be written in a general form like system (10) (but not like system (11), since the operator $T_{\Omega\setminus B,B}$ is nonzero).

**Remark 2**. For irreducible matrix Perron–Frobenius theorem is well known (see, e.g., [10; 12]). This theorem is similar to Theorem 1, but does not require positivity for the diagonal elements (existence of loops on the corresponding impact graph), which makes it possible for several eigenvalues to have the maximal absolute value (in the context of Theorem 1 it means that the unit circle can contain several eigenvalues of the operator $T$). Perron–Frobenius theorem is also applied for the Analytic Hierarchy Process, developed by T. Saaty, particularly in practice problems of economic, industrial, administrative and psychological kinds, in problems of conflict analysis and in other areas [12].

## SUFFICIENT CONDITIONS FOR THE NETWORK ERGODICITY

### Jordan normal form of matrix: existence of limit $\lim\limits_{t\to+\infty}Q^t$

To analyze the network's behaviour as $t\to+\infty$, it is essentially important to know the spectral properties of the impact operator $T$, and these properties can be effectively explored via Jordan normal form of the corresponding matrix (see, e.g., [10; 11]). For referring convenience, consider some statements related to Jordan normal form, which are known or can be easily proven.

It is known (see, e.g., [10; 11]) that for any matrix $Q\in M_{N\times N}[\mathbb{C}]$ there exists the nondegenerate transition matrix $V\in M_{N\times N}[\mathbb{C}]$ such that $Q=VJV^{-1}$, where $J\in M_{N\times N}[\mathbb{C}]$ is the following block-diagonal matrix:

$$J=\begin{pmatrix} J_1 & 0 & \cdots & 0 \\ 0 & J_2 & \cdots & 0 \\ \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_p \end{pmatrix};\ J_s=\begin{pmatrix} \lambda_s & 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_s & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_s & 1 & 0 \\ 0 & 0 & 0 & \cdots & \lambda_s & 1 \\ 0 & 0 & 0 & 0 & \cdots & \lambda_s \end{pmatrix}\in M_{N_s\times N_s}[\mathbb{C}],$$

$$\lambda_s \in \mathbb{C},\ 1 \le s \le p.$$

Matrix $J$ is called Jordan, each matrix $J_s$ $(1 \le s \le p)$ is called a Jordan block of dimension $N_s$ corresponding to an eigenvalue $\lambda_s \in \mathbb{C}$; by this construction, $N_1 + N_2 + \cdots + N_p = N$. Note (see, e.g., [10; 11]) that the columns of the transition matrix $V$ are eigenvectors and generalized eigenvectors of matrix $Q$, and they form so called Jordan basis in $\mathbb{C}^N$.

To compute Jordan matrix, it is convenient to use the following well–known formula:

$$J^t = \begin{pmatrix} (J_1)^t & 0 & \cdots & 0 \\ 0 & (J_2)^t & \cdots & 0 \\ \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & (J_p)^t \end{pmatrix};\ (J_s)^t = \begin{pmatrix} C_t^0\lambda_s^t & C_t^1\lambda_s^{t-1} & \cdots & C_t^t\lambda_s^0 & 0 \\ 0 & C_t^0\lambda_s^t & C_t^1\lambda_s^{t-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & C_t^0\lambda_s^t & C_t^1\lambda_s^{t-1} \\ 0 & 0 & \cdots & 0 & C_t^0\lambda_s^t \end{pmatrix}$$

$$(1 \le s \le p,\ t \ge 1). \qquad (12)$$

(see, e.g., [10] for approaches to defining polynomial and even analytic functions of a matrix).

Given the transition equation $Q^t = VJ^tV^{-1}$, formula (12) provides convenient tools to analyze $Q^t$ for different $t \in \mathbb{N}$, particularly as $t \to +\infty$.

Hereinafter in the space $\mathbb{C}^N$ the norm $\|v\|_\infty = \max\limits_{1 \le j \le N} |v_j|$ $(v \in \mathbb{C}^N)$ is used, in the space $M_{M\times N}[\mathbb{C}]$ the corresponding matrix norm is used:

$$\|R\|_\infty = \sup_{v \in \mathbb{C}^n : \|v\| = 1} \|Rv\|_\infty = \max_{1 \le i \le N} \sum_{k=1}^{N} |R_{i,k}|, \qquad (13)$$

the norm of a linear operator is assumed to be defined by the operator norm of the corresponding matrix by formula (13).

The convergence $\lim\limits_{t \to +\infty} R_t = R$ of the matrix sequence $R_t \in M_{M\times N}[\mathbb{C}]$ $(t \in \mathbb{N})$ to matrix $R \in M_{M\times N}[\mathbb{C}]$ with respect to norm (13) is equivalent to the entrywise convergence: $\lim\limits_{t \to +\infty} (R_t)_{i,j} = R_{i,j}$ for all $1 \le i \le M$, $1 \le j \le N$; the convergence of the sequence of the linear operators is treated as the convergence of the sequence of corresponding matrices.

**Lemma 3.** Let $J_s$ be a Jordan block corresponding to an eigenvalue $\lambda_s \in \mathbb{C}$ such that $|\lambda_s| < 1$. Then:

- $\lim\limits_{t \to +\infty} (J_s)^t = 0_{N_s \times N_s}$, where $0_{N_s \times N_s}$ is a zero matrix of dimension $N_s \times N_s$;
- The convergence $\lim\limits_{t \to +\infty} (J_s)^t = 0_{N_s \times N_s}$ is linear, i.e. there exist constants $C_s > 0$, $q_s \in [0,1)$ and a number $t_s \in \mathbb{N}$ such that

$$\|J_s\|_\infty \le C_s \cdot (q_s)^t \text{ for all } t \ge t_s . \quad (14)$$

**Proof.** It is sufficient to prove that each entry of the matrix $J_s$ converges to zero: $0 \le C_t^i |\lambda_s|^{t-i} \le \frac{t^i}{i!} |\lambda_s|^{t-i} \xrightarrow[t\to+\infty]{} 0$, which implies the required convergence $\lim\limits_{t\to+\infty} C_t^i |\lambda_s|^{t-i} = 0$ for all $0 \le i \le N_s$ (assuming $C_t^i = 0$ for $i > t$). To prove estimate (14), one can choose sufficiently large $t_s$ so that for $H(t) = C_t^i |\lambda_s|^{t-i}$ the following estimate holds: $q_s = \frac{H(t_s+1)}{H(t_s)} \le \left(\frac{t_s+1}{t_s}\right)^{N_s} \cdot |\lambda_s| < 1$. □

**Corollary.** Let the matrix $Q$ have the simple eigenvalue $\lambda_{s_0} = 1$ with an eigenvector $v_{s_0} \in \mathbb{C}^N$, and let any other eigenvalue $\lambda_s$ of $Q$ belong to the open unit disk ($|\lambda_s| < 1$ for $s \ne s_0$). Then:

- There is the convergence $\lim\limits_{t\to+\infty} Q^t = \hat{Q} \in M_{N\times N}[\mathbb{C}]$ with $\hat{Q}v = cv_{s_0}$ for any vector $v \in \mathbb{C}^N$, where the constant $c \in \mathbb{C}$ is defined by the vector $v \in \mathbb{C}^N$;
- The convergence $\lim\limits_{t\to+\infty} Q^t = \hat{Q}$ is linear, i.e. there exist constants $C_0 > 0$, $q_0 \in [0,1)$ and a number $t_0 \in \mathbb{N}$ such that

$$\left\|Q^t - \hat{Q}\right\|_\infty \le C_0 \cdot (q_0)^t \text{ for all } t \ge t_0 . \quad (15)$$

**Proof.** The convergence $\lim\limits_{t\to+\infty} Q^t = \hat{Q} \in M_{N\times N}[\mathbb{C}]$ is implied by formula (12) and Lemma 3; moreover, Lemma 3 and the condition of the corollary yield equality $\hat{Q} = V\hat{J}V^{-1}$, where

$$\hat{J} = \begin{pmatrix} 0 & \cdots & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \cdots & 0 \end{pmatrix}$$

(the only nonzero element of the matrix $\hat{J}$ corresponds to the block $J_{s_0} = J_{s_0}^t = (1)$ for all $t \ge 1$). So, $\operatorname{rank}\hat{J} = 1$, whence, due to nondegeneracy of the transition matrix $V$, $\operatorname{rank}\hat{Q} = \operatorname{rank}\hat{J} = 1$. Therefore, the matrix $\hat{Q}$ defines the linear mapping with one-dimensional image generated by the vector $v_{s_0}$, so equality $\hat{Q}v = cv_{s_0}$ holds for some constant $c \in \mathbb{C}$. Finally,

$$\left\|Q^t - \hat{Q}\right\|_\infty = \left\|V(J^t - \hat{J})V^{-1}\right\|_\infty \le \|V\|_\infty \cdot \left\|V^{-1}\right\|_\infty \cdot \left\|J^t - \hat{J}\right\|_\infty =$$

$$= \|V\|_\infty \cdot \left\|V^{-1}\right\|_\infty \cdot \max_{s\ne s_0} \left\|(J_s)^t\right\|_\infty = \|V\|_\infty \cdot \left\|V^{-1}\right\|_\infty \cdot \max_{s\ne s_0} (C_s \cdot (q_s)^t)$$

for all $t \geq t_0 = \max\limits_{s \neq s_0} t_s$, that proves estimate (15) and thereby completes the proof of the corollary. □

**Sufficient conditions for the network ergodicity: isolated bound**

In the case of isolated bound, the network's behaviour is completely described by system (11).

**Theorem 2.** Let the impact graph $G_T$ satisfy the following conditions:

- for any internal elements $(i_1, j_1), (i_2, j_2) \in \Omega \setminus B$ there exists a directed path from the vertex $(i_1, j_1)$ to $(i_2, j_2)$;
- for any internal element $(i, j) \in \Omega \setminus B$ there is a loop (an edge leading from the vertex $(i, j)$ to the same vertex $(i, j)$);
- all boundary elements $(i, j) \in B$ are isolated vertices.

Then:

- $A_{\Omega \setminus B}(t) \xrightarrow[t \to +\infty]{} c \cdot 1_{\Omega \setminus B}$, where the constant $c \in [0,1)$ is defined by the block $A_{\Omega \setminus B}(0) \in M_{\Omega \setminus B}[0,1]$ representing the states of internal elements at the initial time $t = 0$, $1_{\Omega \setminus B} \in M_{\Omega \setminus B}[0,1]$, $\left(1_{\Omega \setminus B}\right)_{i,j} = 1$ for all $(i, j) \in \Omega \setminus B$;
- The convergence $A_{\Omega \setminus B}(t) \xrightarrow[t \to +\infty]{} c \cdot 1_{\Omega \setminus B}$ is linear, i.e. there exist constants $C_0 > 0$, $q_0 \in [0,1)$ and a number $t_0 \in \mathbb{N}$ such that

$$\left\| A_{\Omega \setminus B}(t) - c \cdot 1_{\Omega \setminus B} \right\|_{\infty} \leq C_0 \cdot (q_0)^t \quad \text{for all} \quad t \geq t_0 .$$

**Proof.** The statement of the theorem is implied by Theorem 1 and corollary of Lemma 3. □

Theorem 2 states that (under the given conditions) for the network with isolated bound there is a set of steady states $c \cdot 1_{\Omega \setminus B}$ ($c \in [0,1)$), where $1_{\Omega \setminus B}$ under the given conditions (see also Theorem 1) is an eigenvector of the operator $T_{\Omega \setminus B, \Omega \setminus B}$ corresponding to the eigenvalue $\lambda = 1$. Recall that the linear operator $T_{\Omega \setminus B, \Omega \setminus B}$ under the given conditions can be treated as an operator with stochastic matrix, which always has the eigenvalue $\lambda = 1$ with the eigenvector $1_{\Omega \setminus B} \in M_{\Omega \setminus B}[\mathbb{R}]$ (since the initial and boundary conditions are located inside the line segment $[0,1]$, one can choose $1_{\Omega \setminus B} \in M_{\Omega \setminus B}[0,1]$).

**Remark 3**. Under the fixed initial conditions $A(0) \in M_{\Omega}[0,1]$ (or equivalently, $A_{\Omega \setminus B}(0) \in M_{\Omega \setminus B}[0,1]$), computation of the steady state $c \cdot 1_{\Omega \setminus B}$ (in fact, it means computation of the constant $c \in [0,1)$) can be reduced through decomposition of $A_{\Omega \setminus B}(0)$ by the Jordan basis of the operator $T_{\Omega \setminus B, \Omega \setminus B}$. However, computation of the Jordan basis for real-world networks can become significantly more

complicated due to the large size of the set $\Omega \setminus B$ and (consequently) the dimension of the space $M_{\Omega\setminus B}[\mathbb{R}]$. Therefore, practically reasonable approach is to approximate (numerically) the eigenvector $c \cdot 1_{\Omega\setminus B}$ using the iterative procedure described by system (11).

Obviously, in the case of isolated bound the block of internal elements of the network can be considered as a network with empty bound ($B = \varnothing$), and this network's behaviour is described by the first equation of system (11).

**Example 2.** Consider the network on the coordinate area $\Omega = \{1,2,\ldots,n\} \times \times \{1,2,\ldots,m\}$ with the bound $B = \varnothing$, the impact operator $T$ simulates equal impact on internal elements by its 4 neighbours:

$$T_{i,j,i,j} = \alpha \quad \text{for all} \quad 1 \le i \le n \ \text{and} \ 1 \le j \le m;$$

$$T_{i,j,i\pm1,j} = 0.25(1-\alpha), \ \text{if} \ 2 \le i \le n-1 \ \text{and} \ 1 \le j \le m;$$

$$T_{i,j,i,j\pm1} = 0.25(1-\alpha), \ \text{if} \ 1 \le i \le n \ \text{and} \ 2 \le j \le m-1;$$

$$T_{1,j,2,j} = T_{n,j,n-1,j} = 0.5(1-\alpha), \ \text{if} \ 1 \le j \le m;$$

$$T_{i,1,i,2} = T_{i,m,i,m-1} = 0.5(1-\alpha), \ \text{if} \ 1 \le i \le n;$$

$$T_{i,j,k,l} = 0, \ \text{if} \ |i-k| + |j-l| \ge 2,$$

where $\alpha \in (0,1)$ is a fixed constant. The corresponding impact graph $G_T$ for the case of $n = 3$, $m = 4$, $\alpha = 0.6$ is depicted in Fig. 4.

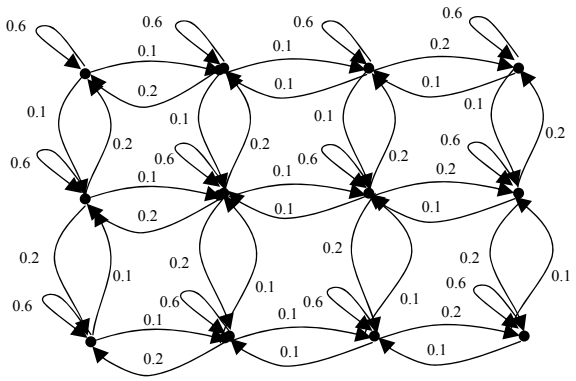


*Fig. 4*

Rewrite equation for transition of state $A(t)$ to the next time moment:

$$(A(t+1))_{i,j} = \alpha(A(t))_{i,j} + 0.25(1-\alpha)((A(t))_{i-1,j} +$$

$$+ (A(t))_{i+1,j} + (A(t))_{i,j-1} + (A(t))_{i,j+1}) \ \text{for} \ 2 \le i \le n-1 \ \text{and} \ 2 \le j \le m-1;$$

$$(A(t+1))_{1,j} = \alpha(A(t))_{1,j} + 0.25(1-\alpha)((A(t))_{1,j-1} +$$

$$+ (A(t))_{1,j+1}) + 0.5(1-\alpha)(A(t))_{2,j} \ \text{for} \ 2 \le j \le m-1;$$

$$(A(t+1))_{n,j} = \alpha(A(t))_{n,j} + 0.25(1-\alpha)((A(t))_{n,j-1} +$$

$$+(A(t))_{n,j+1})+0.5(1-\alpha)(A(t))_{n-1,j} \text{ for } 2\le j\le m-1;$$

$$(A(t+1))_{i,1}=\alpha(A(t))_{i,1}+0.25(1-\alpha)((A(t))_{i-1,1}+(A(t))_{i+1,1})+$$

$$+0.5(1-\alpha)(A(t))_{i,2} \text{ for } 2\le i\le n-1;\ (A(t+1))_{i,n}=\alpha(A(t))_{i,n}+$$

$$+0.25(1-\alpha)((A(t))_{i-1,n}+(A(t))_{i+1,n})+0.5(1-\alpha)(A(t))_{i,n-1} \text{ for } 2\le i\le n-1;$$

$$(A(t+1))_{1,1}=\alpha(A(t))_{1,1}+0.5(1-\alpha)((A(t))_{1,2}+(A(t))_{2,1});$$

$$(A(t+1))_{n,1}=\alpha(A(t))_{n,1}+0.5(1-\alpha)((A(t))_{n,2}+(A(t))_{n-1,1});$$

$$(A(t+1))_{1,m}=\alpha(A(t))_{1,m}+0.5(1-\alpha)((A(t))_{1,m-1}+(A(t))_{2,m});$$

$$(A(t+1))_{n,m}=\alpha(A(t))_{n,m}+0.5(1-\alpha)((A(t))_{n,m-1}+(A(t))_{n-1,m}).$$

Considering coefficients of $(A(t))_{i,j}$ for the different pairs $(i,j)\in\Omega$, one can construct a function $S_w: M_\Omega[0,1]\to[0,1]$ as a 'weighted' sum of $(A(t))_{i,j}$, which is a constant value for all $t\ge 0$:

$$S_w(X)=\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}X_{i,j}+0.5\left(\sum_{j=2}^{m-1}X_{1,j}+\sum_{j=2}^{m-1}X_{n,j}+\sum_{i=2}^{m-1}X_{i,1}+\sum_{i=2}^{m-1}X_{i,m}\right)+$$

$$+0.25(X_{1,1}+X_{n,1}+X_{1,m}+X_{n,m}) \text{ for } X\in M_\Omega[0,1].$$

For $X=A(t+1)$ one can obtain:

$$S_w(A(t+1))=\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}(A(t+1))_{i,j}+$$

$$+0.5\left(\sum_{j=2}^{m-1}((A(t+1))_{1,j}+(A(t+1))_{n,j})+\sum_{i=2}^{n-1}((A(t+1))_{i,1}+(A(t+1))_{i,m})\right)+$$

$$+0.25((A(t+1))_{1,1}+(A(t+1))_{n,1}+(A(t+1))_{1,m}+(A(t+1))_{n,m}).$$

Simplify separately three summands in the right-hand side of the obtained relation:

$$\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}(A(t+1))_{i,j}=$$

$$=\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}(\alpha(A(t))_{i,j}+0.25(1-\alpha)((A(t))_{i-1,j}+(A(t))_{i+1,j}+(A(t))_{i,j-1}+(A(t))_{i,j+1}))=$$

$$=\alpha\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}(A(t))_{i,j}+$$

$$+0.25(1-\alpha)\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}((A(t))_{i-1,j}+(A(t))_{i+1,j}+(A(t))_{i,j-1}+(A(t))_{i,j+1})=$$

$$=\alpha\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}(A(t))_{i,j}+0.25(1-\alpha)\times$$

$$\times\left(\sum_{i=1}^{n-2}\sum_{j=2}^{m-1}(A(t))_{i,j}+\sum_{i=3}^{n}\sum_{j=2}^{m-1}(A(t))_{i,j}+\sum_{i=2}^{n-1}\sum_{j=1}^{m-2}(A(t))_{i,j}+\sum_{i=2}^{n-1}\sum_{j=3}^{m}(A(t))_{i,j}\right)=$$

$$=\alpha\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}(A(t))_{i,j}+0.5(1-\alpha)\left(\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}(A(t))_{i,j}+\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}(A(t))_{i,j}\right)+$$

$$+0.25(1-\alpha)\left(\sum_{j=2}^{m-1}(A(t))_{1,j}-\sum_{j=2}^{m-1}(A(t))_{n-1,j}-\sum_{j=2}^{m-1}(A(t))_{2,j}+\sum_{j=2}^{m-1}(A(t))_{n,j}\right)+$$

$$+0.25(1-\alpha)\left(\sum_{i=2}^{n-1}(A(t))_{i,1}-\sum_{i=2}^{n-1}(A(t))_{i,m-1}-\sum_{i=2}^{n-1}(A(t))_{i,2}+\sum_{i=2}^{n-1}(A(t))_{i,m}\right)=$$

$$=\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}(A(t))_{i,j}+0.25(1-\alpha)\left(\sum_{j=2}^{m-1}((A(t))_{1,j}-(A(t))_{n-1,j})+\sum_{j=2}^{m-1}((A(t))_{n,j}-(A(t))_{2,j})\right)+$$

$$+0.25(1-\alpha)\left(\sum_{i=2}^{n-1}((A(t))_{i,1}-(A(t))_{i,m-1})+\sum_{i=2}^{n-1}((A(t))_{i,m}-(A(t))_{i,2})\right);$$

$$0.5\left(\sum_{j=2}^{m-1}(A(t+1))_{1,j}+\sum_{j=2}^{m-1}(A(t+1))_{n,j}+\sum_{i=2}^{n-1}(A(t+1))_{i,1}+\sum_{i=2}^{n-1}(A(t+1))_{i,m}\right)=$$

$$=0.5\alpha\left(\sum_{j=2}^{m-1}((A(t))_{1,j}+(A(t))_{n,j})+\sum_{i=2}^{n-1}((A(t))_{i,1}+(A(t))_{i,m})\right)+$$

$$+0.125(1-\alpha)\sum_{j=2}^{m-1}((A(t))_{1,j-1}+(A(t))_{1,j+1}+(A(t))_{n,j-1}+(A(t))_{n,j+1})+$$

$$+0.125(1-\alpha)\sum_{i=2}^{n-1}((A(t))_{i-1,1}+(A(t))_{i+1,1}+((A(t))_{i-1,m}+(A(t))_{i+1,m}))+$$

$$+0.25(1-\alpha)\left(\sum_{j=2}^{m-1}((A(t))_{2,j}+(A(t))_{n-1,j})+\sum_{i=2}^{n-1}((A(t))_{i,2}+(A(t))_{i,m-1})\right)=$$

$$=0.5\alpha\left(\sum_{j=2}^{m-1}((A(t))_{1,j}+(A(t))_{n,j})+\sum_{i=2}^{n-1}((A(t))_{i,1}+(A(t))_{i,m})\right)+$$

$$+0.25(1-\alpha)\left(\sum_{j=2}^{m-1}((A(t))_{1,j}+(A(t))_{n,j})+\sum_{i=2}^{n-1}((A(t))_{i,1}+(A(t))_{i,m})\right)+$$

$$+0.125(1-\alpha)((A(t))_{1,1}-(A(t))_{1,m-1}-(A(t))_{1,2}+(A(t))_{1,m})+$$

$$+0.125(1-\alpha)((A(t))_{n,1}-(A(t))_{n,m-1}-(A(t))_{n,2}+(A(t))_{n,m})+$$

$$+0.125(1-\alpha)((A(t))_{1,1}-(A(t))_{n-1,1}-(A(t))_{2,1}+(A(t))_{n,1})+$$

$$+0.125(1-\alpha)((A(t))_{1,m}-(A(t))_{n-1,m}-(A(t))_{2,m}+(A(t))_{n,m})+$$

$$+0.25(1-\alpha)\left(\sum_{j=2}^{m-1}((A(t))_{2,j}+(A(t))_{n-1,j})+\sum_{i=2}^{n-1}((A(t))_{i,2}+(A(t))_{i,m-1})\right)=$$

$$=0.25\left(\sum_{j=2}^{m-1}((A(t))_{1,j}+(A(t))_{n,j})+\sum_{i=2}^{n-1}((A(t))_{i,1}+(A(t))_{i,m})\right)+$$

$$+0.25\alpha\left(\sum_{j=2}^{m-1}((A(t))_{1,j}+(A(t))_{n,j})+\sum_{i=2}^{n-1}((A(t))_{i,1}+(A(t))_{i,m})\right)+$$

$$+0.25(1-\alpha)\left(\sum_{j=2}^{m-1}((A(t))_{2,j}+(A(t))_{n-1,j})+\sum_{i=2}^{n-1}((A(t))_{i,2}+(A(t))_{i,m-1})\right)+$$

$$+0.125(1-\alpha)((A(t))_{1,1}-(A(t))_{1,m-1}-(A(t))_{1,2}+(A(t))_{1,m})+$$

$$+0.125(1-\alpha)((A(t))_{n,1}-(A(t))_{n,m-1}-(A(t))_{n,2}+(A(t))_{n,m})+$$

$$+0.125(1-\alpha)((A(t))_{1,1}-(A(t))_{n-1,1}-(A(t))_{2,1}+(A(t))_{n,1})+$$

$$+0.125(1-\alpha)((A(t))_{1,m}-(A(t))_{n-1,m}-(A(t))_{2,m}+(A(t))_{n,m});$$

$$0.25((A(t+1))_{1,1}+(A(t+1))_{n,1}+(A(t+1))_{1,m}+(A(t+1))_{n,m})=$$

$$=0.25\alpha((A(t))_{1,1}+(A(t))_{n,1}+(A(t))_{1,m}+(A(t))_{n,m})+$$

$$+0.125(1-\alpha)((A(t))_{1,2}+(A(t))_{2,1}+(A(t))_{n-1,1}+(A(t))_{n,2})+$$

$$+0.125(1-\alpha)((A(t))_{2,m}+(A(t))_{1,m-1}+(A(t))_{n-1,m}+(A(t))_{n,m-1}).$$

Collecting these summands together, one can obtain:

$$S_w(A(t+1))=$$

$$=\sum_{i=2}^{n-1}\sum_{j=2}^{m-1}(A(t))_{i,j}+0.5\left(\sum_{j=2}^{m-1}((A(t))_{1,j}+(A(t))_{n,j})+\sum_{i=2}^{n-1}((A(t))_{i,1}+(A(t))_{i,m})\right)+$$

$$+0.25((A(t))_{1,1}+(A(t))_{n,1}+(A(t))_{1,m}+(A(t))_{n,m})=S_w(A(t)),$$

whence

$$S_w(A(0))=S_w\left(\lim_{t\to+\infty}A(t)\right)=S_w\left(c\cdot 1_{\Omega\setminus B}\right)=c\cdot S_w\left(1_{\Omega\setminus B}\right)=$$

$$=c((n-2)(m-2)+0.5\cdot 2\cdot(n-2+m-2)+0.25\cdot 4)=c\cdot(n-1)(m-1),$$

i.e. $c=\frac{S_w(A(0))}{(n-1)(m-1)}$. Particularly, for $n=20$, $m=10$, $\alpha=0.8$, $(A(0))_{i,j}=$ $=\begin{cases}1, & i=j=1,\\ 0, & \text{otherwise},\end{cases}$ one can obtain $c=\frac{0.25}{(20-1)(10-1)}\approx 0.00146$. All data are written with precision up to $0.00001$ which corresponds to relative error $\frac{0.00001}{0.00146}\cong 0.01$, the convergence by the iterative procedure (10) with such precision is achieved

approximately for $t \geq 5000$. It is interesting to note that for this impact operator $T$ the steady state $c \cdot 1_{\Omega \setminus B}$ does not depend on $\alpha \in (0,1)$; however, the value $\alpha \in (0,1)$ affects on the convergence rate (for $\alpha = 0.6$ precision of 0.00001 is achieved approximately for $t \geq 2500$).

**Sufficient conditions for the network ergodicity: non-isolated bound**

In the case of non-isolated bound, the network's behaviour is completely described by system (10).

**Theorem 3.** Let the impact graph $G_T$ satisfy the following conditions:

- for any internal elements $(i_1, j_1), (i_2, j_2) \in \Omega \setminus B$ there exists a directed path from the vertex $(i_1, j_1)$ to $(i_2, j_2)$;
- for any internal element $(i, j) \in \Omega \setminus B$ there is a loop (an edge leading from the vertex $(i, j)$ to the same vertex $(i, j)$);
- at least one boundary element $(i, j) \in B$ is not an isolated vertex.

Then there exists the unique vector $\hat{A}_{\Omega \setminus B} \in M_{\Omega \setminus B}[\mathbb{R}]$ such that:

- $A_{\Omega \setminus B}(t) \xrightarrow[t \to +\infty]{} \hat{A}_{\Omega \setminus B}$;
- the convergence $A_{\Omega \setminus B}(t) \xrightarrow[t \to +\infty]{} \hat{A}_{\Omega \setminus B}$ is linear, i.e. there exist constants $C_0 > 0$, $q_0 \in [0,1)$ and a number $t_0 \in \mathbb{N}$ such that

$$\left\| A_{\Omega \setminus B}(t) - \hat{A}_{\Omega \setminus B} \right\|_{\infty} \leq C_0 \cdot (q_0)^t \text{ for all } t \geq t_0 .$$

**Proof.** System (10) yields the explicit form for $A_{\Omega \setminus B}(t)$ $(t \geq 1)$:

$$A_{\Omega \setminus B}(t) = (T_{\Omega \setminus B, \Omega \setminus B})^t A_{\Omega \setminus B}(0) + \sum_{s=0}^{t-1} (T_{\Omega \setminus B, \Omega \setminus B})^s T_{\Omega \setminus B, B} \Delta_B \qquad (16)$$

Due to Theorem 1, all eigenvalues of the operator $T_{\Omega \setminus B, \Omega \setminus B}$ are located inside the unit disk, so by virtue of Lemma 3 there exist constants $C > 0$, $q \in [0,1)$ and a number $t_0 \in \mathbb{N}$ such that $\left\| T_{\Omega \setminus B, \Omega \setminus B} \right\|_{\infty} \leq C \cdot q^t$ for all $t \geq t_0$, which yields the required convergence:

$$A_{\Omega \setminus B}(t) \xrightarrow[t \to +\infty]{} \hat{A}_{\Omega \setminus B} = \sum_{t=0}^{+\infty} \left( T_{\Omega \setminus B, \Omega \setminus B} \right)^t T_{\Omega \setminus B, B} \Delta_B .$$

Finally,

$$\left\| A_{\Omega \setminus B}(t) - \hat{A}_{\Omega \setminus B} \right\|_{\infty} = \left\| \sum_{s=t+1}^{+\infty} (T_{\Omega \setminus B, \Omega \setminus B})^s T_{\Omega \setminus B, B} \Delta_B \right\|_{\infty} \leq C \left\| T_{\Omega \setminus B, B} \Delta_B \right\|_{\infty} \frac{q^{t+1}}{1-q},$$

so the convergence $A_{\Omega \setminus B}(t) \xrightarrow[t \to +\infty]{} \hat{A}_{\Omega \setminus B}$ is indeed linear. □

Theorem 3 proves that, under the given conditions, for the case of non-isolated bound there exists the unique steady state $\hat{A}_{\Omega \setminus B} \in M_{\Omega \setminus B}[\mathbb{R}]$ (moreover: $\hat{A}_{\Omega \setminus B} \in M_{\Omega \setminus B}[0,1]$, since the initial and boundary conditions are located inside the

line segment $[0,1]$). The equation for the state $\hat{A}_{\Omega\setminus B}$ can be obtained from the first equation of system (10) as $t \to +\infty$:

$$\hat{A}_{\Omega\setminus B} = T_{\Omega\setminus B,\Omega\setminus B}\hat{A}_{\Omega\setminus B} + T_{\Omega\setminus B,B}\Delta_B . \quad (17)$$

**Remark 4**. Since all eigenvalues of the operator $T_{\Omega\setminus B,\Omega\setminus B}$ under the given conditions are located inside the open unit disk, equation (17) under these conditions has the unique solution. However, direct solving equation (17) for real-world networks usually becomes significantly more complicated due to the large size of the set $\Omega\setminus B$ and (consequently) the large dimension of the space $M_{\Omega\setminus B}[\mathbb{R}]$. Therefore, practically reasonable approach is to approximate (numerically) $\hat{A}_{\Omega\setminus B}$ using the iterative procedure described by system (10).

**Example 3.** The network with the impact operator $T$ from Example 1 is obviously the network with non-isolated bound. The conditions of Theorem 3 hold, so the network has the unique steady state $\hat{A}_{\Omega\setminus B} \in M_{\Omega\setminus B}[0,1]$. Equation (17) for this network takes the form:

$$(\hat{A}_{\Omega\setminus B})_{i,j} = 0.25(1-\alpha)((\hat{A}_{\Omega\setminus B})_{i-1,j} +$$

$$+ (\hat{A}_{\Omega\setminus B})_{i+1,j} + (\hat{A}_{\Omega\setminus B})_{i,j-1} + (\hat{A}_{\Omega\setminus B})_{i,j+1}) + \alpha(\hat{A}_{\Omega\setminus B})_{i,j}$$

for all internal $(i,j) \in \Omega\setminus B$ (i.e., for all $2 \le i \le n-1$ and $2 \le j \le m-1$). Thus, given $\alpha \ne 1$, one can obtain:

$$(\hat{A}_{\Omega\setminus B})_{i,j} = \tfrac{1}{4}((\hat{A}_{\Omega\setminus B})_{i-1,j} + (\hat{A}_{\Omega\setminus B})_{i+1,j} + (\hat{A}_{\Omega\setminus B})_{i,j-1} + (\hat{A}_{\Omega\setminus B})_{i,j+1}).$$

Assume that the block $A_B = \hat{A}_B = \Delta_B$ representing the states of boundary elements of the network is defined by four arithmetic progressions:

$$(\Delta_B)_{1,j} = (\Delta_B)_{1,1} + (j-1)\tfrac{(\Delta_B)_{1,m} - (\Delta_B)_{1,1}}{m-1},\ 1 \le j \le m;$$

$$(\Delta_B)_{n,j} = (\Delta_B)_{n,1} + (j-1)\tfrac{(\Delta_B)_{n,m} - (\Delta_B)_{n,1}}{m-1},\ 1 \le j \le m;$$

$$(\Delta_B)_{i,1} = (\Delta_B)_{1,1} + (i-1)\tfrac{(\Delta_B)_{n,1} - (\Delta_B)_{1,1}}{n-1},\ 1 \le i \le n;$$

$$(\Delta_B)_{i,m} = (\Delta_B)_{1,m} + (i-1)\tfrac{(\Delta_B)_{n,m} - (\Delta_B)_{1,m}}{n-1},\ 1 \le i \le n,$$

where the states of the corner elements $b_{\text{top, left}} = (\Delta_B)_{1,1}$, $b_{\text{top, right}} = (\Delta_B)_{1,m}$, $b_{\text{bottom, left}} = (\Delta_B)_{n,1}$, $b_{\text{bottom, right}} = (\Delta_B)_{n,1}$ are the given constants from $[0,1]$. It is easy to see that all elements of matrix $\hat{A}_\Omega \in M_\Omega[0,1]$ (the steady state of the network) also form arithmetic progressions by each row and each column:

$$(\hat{A}_\Omega)_{i,j} = b_{\text{top, left}} + (i-1)\frac{b_{\text{bottom, left}} - b_{\text{top, left}}}{n-1} +$$

$$+ (j-1)\frac{(b_{\text{top, right}} - b_{\text{top, left}}) + (i-1)\frac{(b_{\text{bottom, right}} - b_{\text{bottom, left}}) - (b_{\text{top, right}} - b_{\text{top, left}})}{n-1}}{m-1} =$$

$$= b_{\text{top, left}} + \tfrac{i-1}{n-1}(b_{\text{bottom, left}} - b_{\text{top, left}}) + \tfrac{j-1}{m-1}(b_{\text{top, right}} - b_{\text{top, left}}) +$$

$$+\frac{(i-1)(j-1)}{(n-1)(m-1)}((b_{\text{bottom, right}}-b_{\text{bottom, left}})-(b_{\text{top, right}}-b_{\text{top, left}})) \qquad (18)$$

The table contains the steady state $\hat{A}_\Omega$ of the given network for $n=20$, $m=10$, $\alpha=0.8$, $b_{\text{top, left}}=0.3$, $b_{\text{top, right}}=0.5$, $b_{\text{bottom, left}}=0.8$, $b_{\text{bottom, right}}=$ $=0.9$; all data are written with precision up to $0.01$. Precision $0.01$ is achieved by the iterative procedure (10) approximately for $t\geq 650$. It is interesting to note that for this impact operator $T$ the steady state $\hat{A}_\Omega$ does not depend on $\alpha\in(0,1)$; however, the value $\alpha\in(0,1)$ affects on the convergence rate (for $\alpha=0.6$ precision of $0.01$ is achieved approximately for $t\geq 320$).

**T a b l e**

| *i* | *j* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 1 | 0.30 | 0.32 | 0.34 | 0.37 | 0.39 | 0.41 | 0.43 | 0.46 | 0.48 | 0.50 |
| 2 | 0.33 | 0.35 | 0.37 | 0.39 | 0.41 | 0.43 | 0.46 | 0.48 | 0.50 | 0.52 |
| 3 | 0.35 | 0.37 | 0.39 | 0.42 | 0.44 | 0.46 | 0.48 | 0.50 | 0.52 | 0.54 |
| 4 | 0.38 | 0.40 | 0.42 | 0.44 | 0.46 | 0.48 | 0.50 | 0.52 | 0.54 | 0.56 |
| 5 | 0.41 | 0.43 | 0.45 | 0.46 | 0.48 | 0.50 | 0.52 | 0.54 | 0.56 | 0.58 |
| 6 | 0.43 | 0.45 | 0.47 | 0.49 | 0.51 | 0.53 | 0.55 | 0.57 | 0.59 | 0.61 |
| 7 | 0.46 | 0.48 | 0.50 | 0.51 | 0.53 | 0.55 | 0.57 | 0.59 | 0.61 | 0.63 |
| 8 | 0.48 | 0.50 | 0.52 | 0.54 | 0.56 | 0.57 | 0.59 | 0.61 | 0.63 | 0.65 |
| 9 | 0.51 | 0.53 | 0.55 | 0.56 | 0.58 | 0.60 | 0.62 | 0.63 | 0.65 | 0.67 |
| 10 | 0.54 | 0.55 | 0.57 | 0.59 | 0.60 | 0.62 | 0.64 | 0.66 | 0.67 | 0.69 |
| 11 | 0.56 | 0.58 | 0.60 | 0.61 | 0.63 | 0.65 | 0.66 | 0.68 | 0.69 | 0.71 |
| 12 | 0.59 | 0.61 | 0.62 | 0.64 | 0.65 | 0.67 | 0.68 | 0.70 | 0.72 | 0.73 |
| 13 | 0.62 | 0.63 | 0.65 | 0.66 | 0.68 | 0.69 | 0.71 | 0.72 | 0.74 | 0.75 |
| 14 | 0.64 | 0.66 | 0.67 | 0.69 | 0.70 | 0.72 | 0.73 | 0.74 | 0.76 | 0.77 |
| 15 | 0.67 | 0.68 | 0.70 | 0.71 | 0.72 | 0.74 | 0.75 | 0.77 | 0.78 | 0.79 |
| 16 | 0.69 | 0.71 | 0.72 | 0.74 | 0.75 | 0.76 | 0.78 | 0.79 | 0.80 | 0.82 |
| 17 | 0.72 | 0.73 | 0.75 | 0.76 | 0.77 | 0.79 | 0.80 | 0.81 | 0.82 | 0.84 |
| 18 | 0.75 | 0.76 | 0.77 | 0.78 | 0.80 | 0.81 | 0.82 | 0.83 | 0.85 | 0.86 |
| 19 | 0.77 | 0.79 | 0.80 | 0.81 | 0.82 | 0.83 | 0.84 | 0.86 | 0.87 | 0.88 |
| 20 | 0.80 | 0.81 | 0.82 | 0.83 | 0.84 | 0.86 | 0.87 | 0.88 | 0.89 | 0.90 |

**Remark 5**. In Examples 2 and 3 it is possible to compute analytically the steady state as $t\to+\infty$. However, as it is mentioned in Remark 4, direct solving equation (17) for real-world networks usually becomes significantly more complicated due to the large dimension of the space $M_{\Omega\setminus B}[\mathbb{R}]$. Therefore, practically reasonable is to apply the iterative procedure described by system (10). For more details about analytical solving recurrent relations with multiple indices (with indices $(i,j)\in\Omega$ in the given case of two-dimensional network) see, e.g., [13].

## CONCLUSIONS

- Matrix model for social network is proposed, mutual impact of network elements is represented by the linear impact operator $T$ and the corresponding labelled directed impact graph $G_T$.

- Sufficient conditions for the network ergodicity are given in terms of existence of a steady state, which defines the network's behaviour as $t \to +\infty$.
- For the proposed model, spectral properties of the operator $T$ are explored.
- Sufficient conditions for the network ergodicity are given in the form of existing eigenvalues of the operator $T$ on the unit circle, and in the form of strong connectivity of the impact graph $G_T$.

**REFERENCES**


1. Alan R. Wagner, "Creating and Using Matrix Representations of Social Interaction," *HRI09: International Conference on Human Robot Interaction, 09 March 2009, La Jolla California USA*, pp. 125–132.
2. T.H. Yemelyanenko, A.O. Domashych, "Research of models of social networks," (in Ukrainian), *Actual problems of automation and information technology*, vol. 21, pp. 74–86, 2017.
3. V.V. Breer, D.A. Novikov, A.D. Rogatkin, *Mob Control: Models of Threshold Collective Behavior*. Heidelberg: Springer, 2017, 134 p. doi: https://doi.org/10.1007/978-3-319-51865-7
4. V.N. Burkov, M. Goubko, N. Korgin, D. Novikov, *Introduction to Theory of Control in Organizations*. Boca Raton: CRC Press, 2015, 346 p. doi: https://doi.org/10.1201/b18152
5. A.V. Proskurnikov, R. Tempo, "A tutorial on modeling and analysis of dynamic social networks. Part I," *Annual Reviews in Control*, vol. 43, pp. 65–79, 2017. doi: 10.1016/j.arcontrol.2017.03.002
6. A.V. Proskurnikov, R. Tempo, "A tutorial on modeling and analysis of dynamic social networks. Part II," *Annual Reviews in Control*, vol. 45, pp. 166–190, 2018. doi: 10.1016/j.arcontrol.2018.03.005
7. P.S. Senyo, *Probability theory and mathematical statistic*. Kyiv: Center of educational literature, 2004, 448 p.
8. Richard Bellman, *Introduction to Matrix Analysis; Second Edition*. Philadelphia: Society for Industrial and Applied Mathematics, 1997, 431 p.
9. A. Hallak, G. Dalal, "On the Products of Stochastic and Diagonal Matrices," *NVIDIA Research*, 2023, 6 p. doi: 10.48550/arXiv.2304.11634
10. F.R. Gantmacher, *Theory of matrices; Vol. 1*. New York: Chelsea Publishing Company, 1959, 374 p.
11. V.A. Ilyin, E.G. Poznyak, *Linear Algebra*. 1987, 288 p.
12. Thomas L. Saaty, *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. New York: McGraw-Hill, 1980, 287 p.
13. M. Bousquet-Mélou, M. Petkovšek, "Linear recurrences with constant coefficients: the multivariate case," *Discrete Mathematics*, vol. 225, pp. 51–75, 2000. doi: 10.1016/S0012-365X(00)00147-3


*Received 02.08.2024*

**INFORMATION ON THE ARTICLE**

**Igor Ya. Spectorsky,** ORCID: 0000-0003-4863-7986, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: i.spectorsky@gmail.com

**Vitalii M. Statkevych,** ORCID: 0000-0001-5210-9890, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: mstatckevich@yahoo.com

**Oleksandr V. Stus,** ORCID: 0000-0003-3426-5093, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: o.stus@kpi.ua

**МАТРИЧНО-ГРАФІЧНЕ МОДЕЛЮВАННЯ СОЦІАЛЬНОЇ МЕРЕЖІ: ЕРГОДИЧНІ ВЛАСТИВОСТІ /** І.Я. Спекторський, В.М. Статкевич, О.В. Стусь

**Анотація.** Запропоновано математичний апарат моделювання соціальних мереж, який дозволяє отримати достатні умови ергодичності мережі, тобто існування граничного стаціонарного стану при $t \to +\infty$. Запропонована модель є лінійною: елементи мережі утворюють двовимірний масив (матрицю), елементом матриці в момент часу $t \geq 0$ є стан $A_{i,j}(t) \in [0,1]$ елемента з координатами $(i,j) \in \Omega = \{1,2,\ldots,n\} \times \{1,2,\ldots,m\}$. Взаємний вплив між елементами задано оператором впливу $T$ — чотиривимірним масивом, елементи $T_{i,j,k,l} \geq 0$ якого позначають вплив елемента $(k,l) \in \Omega$ на елемент $(i,j) \in \Omega$: $(TA)_{i,j} = \sum_{k=1}^{n} \sum_{l=1}^{m} T_{i,j,k,l} A_{k,l}$. Для $T$ запропоновано зображення у вигляді графу $G_T$, вершини якого відповідають елементам $(i,j) \in \Omega$: орієнтоване ребро (дуга) з міткою $T_{i,j,k,l}$ веде від вершини $(k,l) \in \Omega$ до вершини $(i,j) \in \Omega$ тоді й тільки тоді, коли $T_{i,j,k,l} > 0$. На $\Omega$ виділено край $B \subset \Omega$: $T_{i,j,k,l} = 0$ для $(k,l) \in \Omega$, $(i,j) \in B$. Стан $A(t+1)$ мережі у момент часу $t+1$ визначається станом $A(t)$ мережі у момент часу $t \geq 0$ згідно з рівнянням $A(t+1) = TA(t) + \Delta$, де матриця $\Delta$ розмірності $n \times m$ визначає стан крайових елементів мережі; $\Delta_{i,j} = 0$ для внутрішніх елементів $(i,j) \in \Omega \setminus B$. Достатні умови ергодичності мережі надано у термінах властивостей зв'язності графу впливу $G_T$: мають існувати шляхи між довільними вершинами та усі петлі. Наведені умови забезпечують розташування спектра оператора $T$ всередині одиничного круга за винятком, можливо, $\lambda = 1$; доведено, що $\lambda = 1$ є власним числом $T$ лише у випадку ізольованого краю (жоден крайовий елемент не впливає на жоден внутрішній елемент мережі). Наведені спектральні властивості $T$ забезпечують існування стаціонарного стану, який можна знаходити ітераційною процедурою $A(t+1) = TA(t) + \Delta$ за заданим $A(0)$ з геометричною (лінійною) швидкістю збіжності.

**Ключові слова:** соціальна мережа, моделювання, ергодичність, власне число, жорданова нормальна форма.

UDC 004.852
DOI: 10.20535/SRIT.2308-8893.2025.4.04

# ANALYSIS OF ACTUARIAL RISK WITH GENERALIZED LINEAR MODELS

**R.S. PANIBRATOV, P.I. BIDYUK**

**Abstract.** The problem of applying generalized linear models to the analysis of actuarial risks in the context of premium charges to clients was considered. The Monte-Carlo method for Markov chains was applied. Two situations were considered for the computational experiment. For the first one, insurance indicators and the target variable were randomly assigned due to the problem of public data access. To create three datasets, charges were generated from normal, gamma, and Pareto distributions with dynamic variance, and noise was added to stimulate a non-stationary process. In the second situation, actual actuarial data from the Singapore Actuarial Society was used. Generalized Linear Models with normal distribution and logarithmic link function, an exponential distribution and logarithmic link function, and Laplace distribution with identity link function were constructed. Based on the model-fitting quality metrics, conclusions were drawn about their structure.

**Keywords:** actuarial risk, generalized linear models, simulation modeling, exponential family of distributions, Bayesian data analysis, Monte Carlo method for Markov chains.

## INTRODUCTION

Since insurance protects people and organizations financially against a variety of risks, it is seen as a fundamental component of the economy. Because it assists in managing and reducing the risks involved in providing insurance to both consumers and businesses: actuarial science is essential to the insurance sector. A thorough understanding of mathematics, statistics, finance, and economics is necessary to work as an actuary. Actuaries apply their knowledge to assist insurance companies in estimating the cost of possible risks and estimating the probability of future events.

In order to reduce the risks and minimize the financial impact of unpredictable events, the insurance sector is essential. The frequency or timing of these occurrences, however, cannot be predicted. Actuarial risk, or the likelihood of an event happening and the possible financial impact it may have, is a key component that insurance companies utilize to prevent themselves from financial catastrophe. Because actuarial risk is a complicated process that calls for certain knowledge and skills, actuaries are important to the insurance sector. Actuarial

risk is fundamentally about estimating the probability of an unfavorable event happening and the possible financial consequences it may have. Actuaries analyze data and forecast the probability of an event by using complex mathematical models. They then use this data to estimate the event's financial effect and compute the premium needed to cover the risk. The business of insurance companies is risk management. Actuaries are essential in assisting insurance firms in figuring out how much risk they may accept while maintaining their financial stability. They accomplish this via examining historical data and applying statistical techniques to forecast the probability that comparable occurrences will take place in the future.

The insurance business uses the Generalized Linear Model (GLM), a statistical technique, to calculate insurance policy prices. In order to analyze and forecast the anticipated cost of claims based on different risk indicators related to the insured entities, generalized linear models are used. Compared to simpler linear models, these models offer a more complex and precise pricing mechanism by allowing actuaries and analysts to include various data types and variable relationships, such as the linear or exponential relationship between risk factors and claim costs.

Linear models are a specific instance of the many models that comprise up GLM. The assumptions of normality, constant variance, and additive effect of that are restricted in linear models are eliminated. Rather, it is assumed that the response variable belongs to the exponential distribution family.

The exponential distributions family consists of the next structure [1]:

$$f(y_i;\theta_i;\varphi)=\exp\left\{\frac{y_i\theta_i-b(\theta_i)}{a_i(\varphi)}+c(y_i,\varphi)\right\},$$

where $a_i(\varphi)$, $b(\theta_i)$ and $c(y_i,\varphi)$ are prior defined functions; $\theta_i$ is parameter, associated with mean; $\varphi$ is parameter, associated with variance.

Additionally, the variance is allowed to change simultaneously with the distribution mean. Lastly, on a transformed scale, it is believed that the variables' effects on the response variable are additive [2].

For GLM, the following assumptions are made:

1. **Stochastic component**: every component of $Y$ comes from the single exponential family distribution and is independent.

2. **Systematic component**: the linear predictor $\eta$ is formed from $p$ explanatory variables:

$$\eta=X\beta,$$

where $X$ is design matrix; $\beta$ is vector of estimation parameters.

3. **Link function**: relationship between stochastic and systematic component is defined by the link function, which is monotonic and differentiable:

$$E[Y]=\mu=g^{-1}(\eta).$$

**Problem Statement.** The purpose of the study is to apply GLM for analysis of actuarial risks using different distributions and specified link functions and previously applying Bayesian data analysis.

**IMPORTANCE OF GLM**

Because they offer a versatile framework for modeling the link between the response variable (say, such as the frequency or cost of a claim) and one or more predictor factors (such as age, vehicle type or geographic area), GLMs are also utilized in insurance pricing.

The authors of [3] emphasized that when doing statistical studies with GLMs, non-robustness against outliers is an important consideration. Additionally, they demonstrated that there aren't many reliable options, particularly when performing Bayesian statistical analysis. Focusing on gamma GLM, a widely used tool in actuarial science, they put forth a robust and efficient modeling-based method that can be applied to both frequentists and Bayesian studies. The suggested model can be easily estimated, at least on small-to-moderate-sized data sets, and is simple to analyze and comprehend.

The authors of [4] presented a brand-new deep learning technique called Deeply-learned Generalized Linear Model with Missing Data (DLGLM), which can make predictions and estimate coefficients even when there is missing not at random (MNAR) data. The creation of the data matrix and the connections between the response variable and the mask of missing values are modeled by DLGLM using deep learning neural network architecture. They were able to generalize the conventional GLM this way, taking into consideration both ignorable and non-ignorable types of missing values in the data, as well as intricate nonlinear relationships between the features. Through simulations and actual data analyses, the authors also showed that DLGLM outperforms alternative impute-then-regress techniques, such as mean and mouse imputation, in terms of coefficient estimation and prediction when MNAR missing values are present.

The problem of GLM transfer learning was studied in [5]. Bounds for estimate error and the prediction error measure with fast and slow rates under various scenarios are derived by the authors, who also suggested GLM transfer learning methods. To create confidence intervals for each coefficient component with theoretical assurances, they took into account the two-step transfer learning approach. At last, they used a real-data research and simulations to show how effective their algorithms were.

In the context of claim counts modeling, the authors of [6] suggested a method for identifying the next-best interaction to be added to an arbitrary but fixed benchmark GLM. They started by training a combined actuarial neural network (CANN) model, which is essentially a neural network that improves the benchmark GLM. Second, they sorted interactions by their strength and quantified the strength of interactions between each pair of characteristics using a quick model-specific technique called Neural Interaction Detection. Third, they compared a few small GLMs that matched the top-ranked interactions to determine the next-best interaction. This technique offers two benefits. First of all, it is completely automatable method of adding the next-best interaction that is absent from the benchmark GLM. Second, according to Friedman's H-statistic, the authors' methodology is quicker than alternative strategies. As a result, enormous data sets containing millions of observations and dozens of attributes are particularly well-suited for the proposed technique. Consequently, it can significantly reduce the time that price actuaries spend looking for interactions to enhance their GLMs, which is often time-consuming and visual process.

It was demonstrated in [7] that GLM is the best choice for estimation of operational risk. This approach demonstrated excellent risk estimating quality with minimum errors.

Alternative methods of estimating parameters of GLM were analyzed in [8].

## MONTE-CARLO METHOD FOR MARKOV CHAIN

Finding the posterior distribution is the primary objective of Bayesian data analysis:

$$P(\theta \mid X) = \frac{P(X \mid \theta)P(\theta)}{P(X)},$$

where $X$ is state space vector; $\theta$ is a parameter of distribution; $P(X \mid \theta)$ is the likelihood; $P(\theta)$ is the prior; $P(X)$ is a normalizing constant, also known as the evidence or marginal likelihood.

The denominator can be expressed as follows:

$$P(X) = \int P(X \mid \theta^*)P(\theta^*)d\theta^*.$$

The challenge of assessing the integral in the denominator is the computing problem. Markov Chain Monte Carlo (MCMC) is the most significant of the Monte Carlo techniques that may be employed.

MCMC is the method that uses a Markov chain mechanism to generate samples $x^{(i)}$ while exploring the state space, $X$. The purpose of this technique is to increase the amount of time the chain spends in the most crucial areas [9]. It is specifically designed to make the samples $x^{(i)}$ resemble samples generated from the desired distribution, $p(x)$.

Monte Carlo is the method for approximating a desired quantity by sampling from a probability distribution. It estimates a deterministic quantity of interest using randomization. The Monte Carlo approach is used to approximate such numbers by averaging over samples. For example, if there is an expectation or expectations to estimate, $s$, they may be extremely complicated integrals or perhaps impossible to estimate:

$$s = \int p(x)f(x)dx = E_p[f(x)],$$

$$\widetilde{s}_n = \frac{1}{n}\sum_{i=1}^{n} f(x^i),$$

where $f(x)$ is the probability density function.

The standard error might be decreased and a reasonably good estimate could be obtained by calculating the average across a large number of samples. One drawback of this approach is that it makes the assumption that sampling from a probability distribution is simple, which isn't always feasible. In many cases, sampling from the distribution is not even feasible. In these situations, we efficiently sample from an intractable probability distribution by using Markov chains.

With a modification, MCMC techniques function similarly to normal Monte Carlo methods, but the produced drawings $x_1,...,x_n$ are serially correlated rather

than independent. Specifically, they are the realizations of a Markov Chain consisting of $N$ random variables, $X_1,...,X_n$.

If and only if, for all positive integers $k$ and, $n$, these future observations $X_{i+n}$ are conditionally independent of the previous values $X_{i-k}$ given the present value p $X_i$, then a random sequence $\{X_i\}$ is Markov chain:

$$P(X_{i+n} = x \mid X_i, X_{i-1},..., X_{i-k}) = P(X_{i+n} = x \mid X_i).$$

This condition that sometimes is referred to as Markov property, indicates that the process is memoryless: the probability distribution of the chain's future values is only dependent on its present value con $X_i$, independent of how the value was arrived at (e. g. the chain's previous transition).

Although MCMC comes in a variety of flavors, the Metropolis–Hastings random walk algorithm is the easiest to implement. Standard uniform distribution, proposal distribution $p(x)$ and the target distribution must be used for applying Metropolis–Hastings algorithm.

The following steps how this algorithm works when given an initial prediction for $\theta$ that has a positive probability of being drawn.

1. Select a new suggested value $\theta_p$ that equals

$$\theta_p = \theta + \Delta\theta,$$

where $\Delta\theta$ has specific distribution for transition (for example, Normal).

2. Calculate the ratio

$$\rho = \frac{g(\theta_p \mid X)}{g(\theta \mid X)},$$

where $g$ is the posterior probability.

3. To preserve the precise balance of the stationary distribution in the event that the proposal distribution is not symmetrical, the acceptance probability must be weighted and then calculated:

$$\rho = \frac{g(\theta_p \mid X)p(\theta \mid \theta_p)}{g(\theta \mid X)p(\theta_p \mid \theta)}.$$

Given that ratios are being taken, any distribution proportional to $g$ will likewise be canceled by denominator, therefor it may be utilized as follows:

$$\rho = \frac{p(X \mid \theta_p)p(\theta_p)}{p(X \mid \theta)p(\theta)}.$$

4. If $\rho \geq 1$, then $\theta = \theta_p$.

If $\rho < 1$, then $\theta = \theta_p$ with probability $\rho$, else $\theta = \theta$, where the uniform distribution is used.

5. Repeat earlier steps.

Authors in [10] showed that MCMC approaches appear to be quite helpful in a wide range of applications. However, because MCMC methods are imprecise, deviations from the correct findings may occur due to their unpredictability. Because no guaranty can be provided, MCMC should only be utilized in extreme cases and only when there are no other options. As the parameters change over

time, performance may also be maximized by dynamically modifying the parameters, especially the covariance matrix, without changing the distribution. Furthermore, for low correlations in higher dimensions, other modifications to Metropolis–Hastings are needed.

The authors of [11] presented a Poisson–Rayleigh model, which is also known as the PR-distribution, with two parameters. They were able to get a number of distinct features. The parameters of the PR distribution have been estimated using Bayesian methods, maximum likelihood, and maximum product spacing. For Bayesian estimation, the estimators were approximated using point and interval estimation using the MCMC approach, which is based on a symmetric loss function. A Bayesian estimator based on gamma priors has been proposed.

New diagnostics for evaluating MCMC algorithms efficiency, reliability, and flexibility using control and attainment maps were presented in [12]. The time needed for hyper-parameter adjustment may be shortened by the results of these new diagnostics. The diagnostics themselves can be carried out on computationally reasonable test problems with known posteriors, as demonstrated there, but they need a non-trivial computational experiment. The results of these diagnostics may be used to determine the optimal algorithm and matching hyper-parameter setup for calibrating a real-world issue that is more computationally demanding and shares traits with the test problems. The convergence of that particular search procedure may then be evaluated by applying the current MCMC diagnostics to the single calibration run of the real-world issue.

In order to increase effectiveness of posterior exploration using MCMC techniques, a Kalman-inspired proposal distribution was presented in [13]. Similar to the analysis stage in the Kalman filter, this novel proposal distribution creates candidate states by taking use the cross covariances of model parameters, measurements, and model outputs. The asymmetric nature of the Kalman-inspired proposal distribution limits its application to a brief burn-in time, following which the chains are evolved using a combination of parallel direction and snooker candidate states. The sampled chains will converge to the precise target distribution thanks to diminishing adaptability. The new proposal distribution may be easily included into any suitable MCMC technique and is not restricted to any particular MCMC methodology.

The authors of [14] investigated Metropolis–Hastings Markov chain convergence rates. The validity of appropriate central limit theorems for Markov chains can be ensured by qualitative convergence rates. The impact of growing dimensions, data size, and other variables on these algorithms' efficiency can be better understood by looking at explicit convergence rates. However, a significant amount of work is still needed in this field since explicit quantitative convergence rates are difficult to establish and remain elusive in many situations of relevance. These subjects are crucial for comprehending Metropolis–Hastings behavior in contemporary issues where there may be a lot of data, a lot of dimensions, or both.

## NUMERICAL EXPERIMENT WITH ARTIFICIAL DATA

Due to the case, that actuarial data is not always available, it was decided to simulate first actuarial insurance data artificially following the next structure. Three datasets for experiment were created. For imitating data of policyholders the next features were used:

1. **Age:** numerical variable, which shows age of client and ranges between 19 and 64.

2. **Sex:** categorical variable, which identifies sex of client and has states ‘M’ for male and ‘F’ for female.

3. **BMI:** numerical variable, which shows body mass index of client. Uniform distribution was used for generation.

4. **Region:** categorical variable, which shows place of client’s residence and has state ‘A’, ‘B’, ‘C’ and ‘D’.

5. **Medical History:** categorical variable, which identifies history of previous illnesses of clients and has state ‘Diabetes’, ‘High blood pressure’ or ‘None’.

6. **Exercise:** categorical variable, which shows if client does exercise. It has states ‘Always’, ‘Rarely’ or ‘Never’.

7. **Worker Status:** categorical variable, which shows working status of client and has states ‘Employed’, ‘Student’ and ‘Unemployed’.

8. **Charges:** numerical variable which shows total charges by the insurance company. This is target variable.

For the last feature next 3 distributions were use:

- Normal;
- Gamma;
- Pareto.

For making charges as non-stationary process, algorithm of mixture distribution was applied, which consist of the next steps:

1. Generate random variable $p$, which has uniform distribution $p \sim U(0,1)$.

2. If $p \in \left[\sum_{i=1}^{k-1} p_i, \sum_{i=1}^{k} p_{i,}\right)$, then generate variable with chosen distribution with fixed parameter of centre and randomly generated scale parameter.

3. Repeat until size of the dataset will be reached.

After generating target variable, the noise, which has zero mean and variable standard deviation was added.

Three GLMs were built for forecasting were implemented with specified link functions:

1. GLM with normal distribution and logarithmic link function.

2. GLM with exponential distribution and logarithmic link function.

3. GLM with Laplace distribution and identity link function.

After implementing GLMs by using MCMC method next metrics of models quality were used:

- Logarithm of maximized value of a likelihood function.
- Akaike information criterion (AIC):

$$AIC = 2 * k - 2 * \ln(\widetilde{L}),$$

where $\widetilde{L}$ is maximized value of likelihood function; $k$ is the number of estimated parameters.

- Bayesian information criterion (BIC):

$$BIC = k * \ln(n) - 2 * \ln(\widetilde{L}),$$

where $\widetilde{L}$ is maximized value of likelihood function; $k$ is the number of estimated parameters; $n$ is the number of data points.

The metric results of GLM parameters estimation for three distinct datasets are shown in Tables 1–3.

**Table 1.** Results of GLM construction using simulated actuarial insurance data, where charges have normal distribution

| Metric | GLM Normal | GLM Exponential | GLM Laplace |
|---|---|---|---|
| Log-Likelihood | 1.248 | 2.552 | 1.943 |
| AIC | 15.503 | 10.896 | 14.1134.11 |
| BIC | 56.464 | 47.305 | 55.0735.0 |

**Table 2.** Results of GLM construction for simulated actuarial insurance data, where claim payments have gamma distribution

| Metric | GLM Normal | GLM Exponential | GLM Laplace |
|---|---|---|---|
| Log-Likelihood | 1.16 | 3.3053. | 2.232 |
| AIC | 15.68 | 9.39 | 13.535 |
| BIC | 56.64 | 35.798 | 54.495 |

**Table 3.** Results of GLM construction for simulated actuarial insurance data, where claim payments have Pareto distribution

| Metric | GLM Normal | GLM Exponential | GLM Laplace |
|---|---|---|---|
| Log-Likelihood | 0.261 | 1.678 | 0.641 |
| AIC | 17.479 | 12.644 | 16.718 |
| BIC | 58.439 | 49.053 | 57.677 |

From the results of fitting GLMs it can be seen, that GLM with exponential distribution and log link function demonstrated the best results for all datasets. On the other side, GLM with Laplace distribution and identity link function also showed acceptable results for dataset with normal distributions of charges.

Results of forecasting for best GLM models using different datasets are shown on Figs. 1–4.

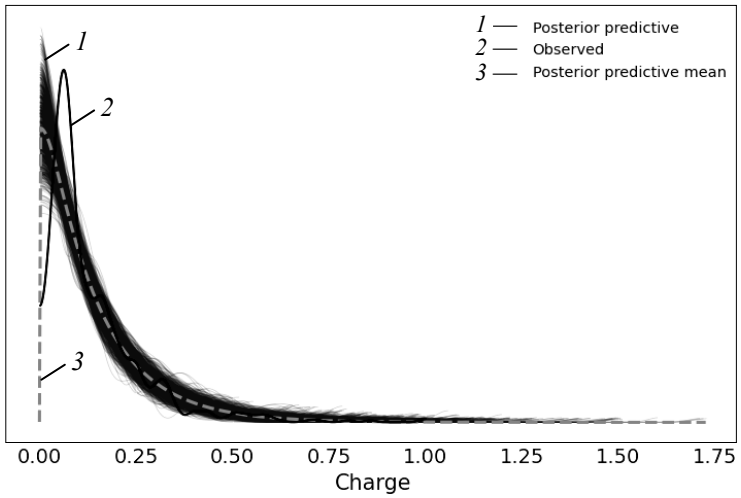



*Fig. 1*. Result of forecasting GLM with exponential distribution and log link function for charges, which have normal distribution

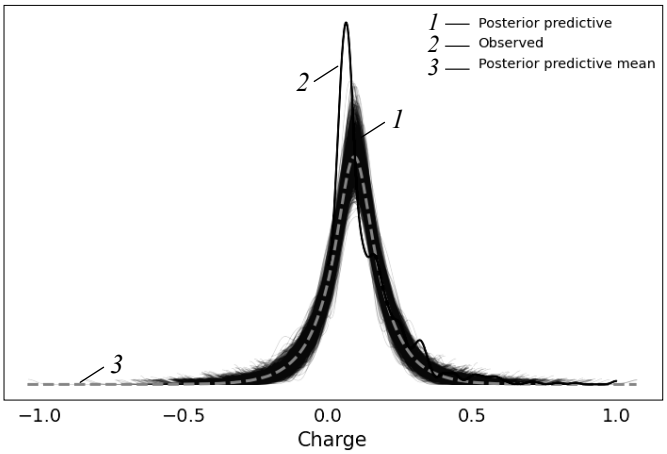



*Fig. 2*. Result of forecasting GLM with Laplace distribution and identity link function for charges, which have normal distribution

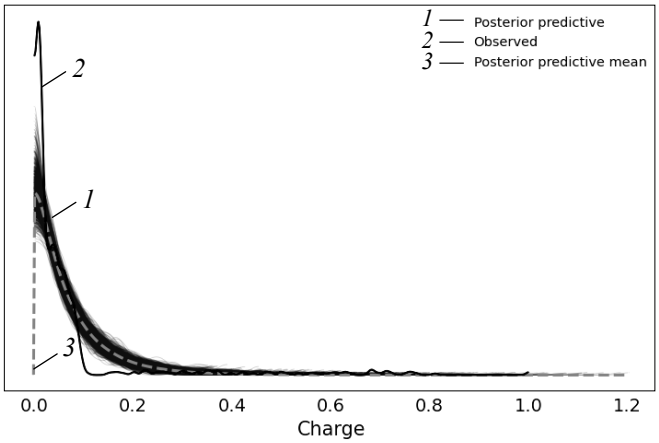



*Fig. 3*. Result of forecasting GLM with exponential distribution and log link function for charges, which have gamma distribution

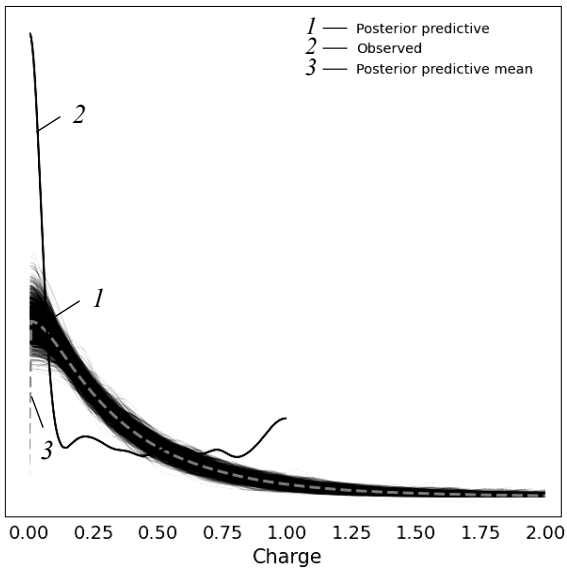



*Fig. 4*. Result of forecasting GLM with exponential distribution and log link function for charges, which have Pareto distribution

Tables 4–7 show numerical summaries of posterior parameter estimates for the best GLMs with different datasets, which include mean value, standard deviation and highest density region (3% and 97%).

**Table 4.** Numerical characteristics of posterior parameter estimates for exponential GLM and charges with normal distribution

| Parameter | Mean | Std | HDI-3% | HDI-97% |
|---|---|---|---|---|
| Intercept | -1.986 | 0.147 | -2.263 | -1.722 |
| Age | 0.047 | 0.124 | -0.192 | 0.283 |
| Sex | -0.035 | 0.076 | -0.181 | 0.097 |
| BMI | 0.071 | 0.131 | -0.149 | 0.338 |
| Region | -0.003 | 0.033 | -0.066 | 0.056 |
| MedHistory | 0.001 | 0.047 | -0.079 | 0.092 |
| Exercise | -0.045 | 0.048 | -0.131 | 0.038 |
| WorkerStatus | 0.000 | 0.046 | -0.089 | 0.088 |

**Table 5.** Numerical characteristics of posterior parameter estimates for Laplace GLM and charges with normal distribution

| Parameter | Mean | Std | HDI-3% | HDI-97% |
|---|---|---|---|---|
| b | 0.082 | 0.003 | 0.077 | 0.088 |
| Intercept | 0.087 | 0.012 | 0.062 | 0.109 |
| Age | 0.019 | 0.012 | -0.003 | 0.043 |
| Sex | -0.012 | 0.007 | -0.025 | 0.001 |
| BMI | 0.015 | 0.012 | -0.007 | 0.038 |
| Region | -0.000 | 0.003 | -0.006 | 0.005 |
| MedHistory | -0.007 | 0.004 | -0.015 | 0.000 |
| Exercise | -0.000 | 0.004 | -0.009 | 0.007 |
| WorkerStatus | 0.003 | 0.004 | -0.005 | 0.010 |

**Table 6.** Numerical characteristics of posterior parameter estimates for exponential GLM and charges with gamma distribution

| Parameter | Mean | Std | HDI-3% | HDI-97% |
|---|---|---|---|---|
| Intercept | -2.975 | 0.157 | -3.266 | -2.659 |
| Age | -0.091 | 0.136 | -0.335 | 0.17 |
| Sex | -0.062 | 0.077 | -0.203 | 0.083 |
| BMI | 0.253 | 0.139 | -0.014 | 0.498 |
| Region | 0.150 | 0.036 | 0.078 | 0.212 |
| MedHistory | -0.001 | 0.049 | -0.087 | 0.1 |
| Exercise | 0.094 | 0.047 | 0.008 | 0.185 |
| WorkerStatus | -0.06 | 0.046 | -0.142 | 0.027 |

**Table 7.** Numerical characteristics of posterior parameter estimates for exponential GLM for claim payments with Pareto distribution

| Parameter | Mean | Std | HDI-3% | HDI-97% |
|---|---|---|---|---|
| Intercept | -1.034 | 0.152 | -1.326 | -0.774 |
| Age | 0.013 | 0.13 | -0.218 | 0.271 |
| Sex | 0.069 | 0.075 | -0.061 | 0.212 |
| BMI | -0.174 | 0.135 | -0.42 | 0.093 |
| Region | 0.067 | 0.037 | -0.004 | 0.136 |
| MedHistory | -0.019 | 0.047 | -0.114 | 0.059 |
| Exercise | -0.087 | 0.047 | -0.17 | 0.002 |
| WorkerStatus | 0.038 | 0.046 | -0.044 | 0.121 |

## NUMERICAL EXPERIMENT WITH ACTUAL DATA

For this scenario the actual actuarial data of insurance company were applied for fitting GLM. Dataset was taken from Singapore Actuarial Society. All of the worker compensation insurance policies in this dataset have experienced an accident. The next features were used:

1. **Age**.
2. **Sex**.
3. **MaritalStatus:** categorical variable, which identifies marital status of clients.
4. **DependentChildren:** numerical variable, which shows number of dependent children.
5. **DependentOthers:** numerical variable, which shows number of dependent, excluding children.
6. **WeeklyWages:** numerical variable, which shows total weekly wage.
7. **PartFullTime:** categorical variable, which shows working mode.
8. **HoursWorkedPerWeek:** numerical variable, which shows total hours worked per week.
9. **DaysWorkedPerWeek:** numerical variable, which shows number of days worked per week.
10. **UltimateIncurredClaimCost:** numerical variable which shows total claims payments by the insurance company. This is target variable.

Results of fitting GLM from previous experiment are shown in Table 8.

**Table 8.** Results of GLM construction using actual insurance actuarial data

| Metric | GLM Normal | GLM Exponential | GLM Laplace |
|---|---|---|---|
| Log-Likelihood | 2.155 | 6.231 | 3.593 |
| AIC | 17.691 | 7.538 | 14.815 |
| BIC | 67.753 | 53.048 | 64.877 |

It can be observed that the exponential GLM with logarithmic link demonstrated best results among others for real dataset.

Results of forecasting for best GLM model for real dataset are shown in Fig. 5.

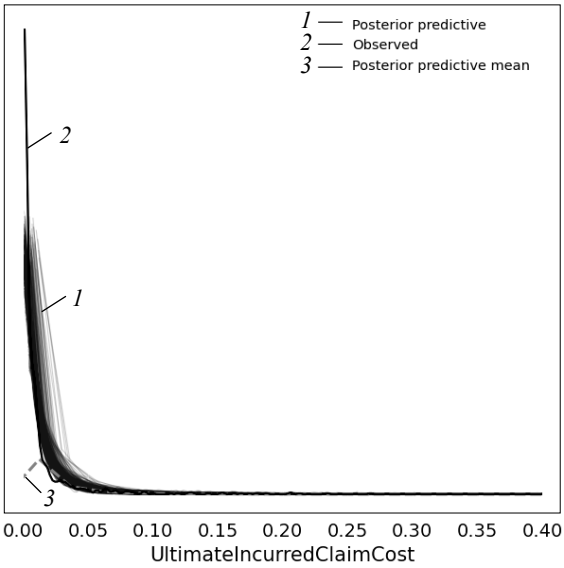



*Fig. 5*. Result of forecasting GLM with exponential distribution and log link function for actual actuarial insurance data

Table 9 show numerical summaries of posterior parameter estimates for best GLM.

**Table 9.** Numerical characteristics of posterior parameter estimates for exponential GLM and claim payments from real dataset

| Parameter | Mean | Std | HDI-3% | HDI-97% |
|---|---|---|---|---|
| Intercept | -4.051 | 0.246 | -4.485 | -3.594 |
| Age | 0.934 | 0.191 | 0.577 | 1.266 |
| Sex | -0.326 | 0.1 | -0.515 | -0.142 |
| MaritalStatus | -0.346 | 0.064 | -0.467 | -0.229 |
| DependentChildren | 3.440 | 0.502 | 2.521 | 4.403 |
| DependentOthers | -0.232 | 0.505 | -1.064 | 0.776 |
| WeeklyWages | 4.722 | 0.459 | 3.882 | 5.582 |
| PartFullTime | -0.217 | 0.199 | -0.579 | 0.146 |
| HourWorkedPerWeek | 0.986 | 0.497 | 0.1 | 1.964 |
| DaysWorkedPerWeek | -1.799 | 0.572 | -2.864 | -0.759 |

## CONCLUSIONS

The application of GLM to the analysis of actuarial risks in the context of client claim payments is taken into consideration. For estimation parameters of models the MCMC method was implemented. The insurance indicators and the target variable were created artificially since actuarial insurance data is frequently not made public: age, sex, BMI, region, medical history, exercise, worker status and charges. The last one was generated by applying algorithm of mixture distribution, using normal, gamma and Pareto distribution with adding Gaussian noise, which had zero mean and variable standard deviation to create non-stationary process. Also real actuarial insurance data from Singapore Actuarial Society were used for experiments. Three GLM were implemented for experiments: normal with logarithmic link function, exponential with logarithmic link function and Laplace distribution with identity link function. Based on the experiment findings, it can be said that exponential GLM generally produced the best results for both artificial and real data. For the case of the normal distribution, Laplace GLM also produced positive results for artificial data.

In future studies it is planned to automatize the process of insurance data analysis using artificial intelligence and simulation techniques. As far as most of financial processes belong to the class of non-linear and non-stationary the methodology will be proposed for constructing such models. It is also planned to apply the methods of generating alternative managerial decision using Bayesian approach to data and expert estimates analysis.

## REFERENCES

1. P. McCullagh, J. Nelder, *Generalized Linear Models*; 2nd edition. Chapman & Hall, 1989, 532 p.
2. D. Anderson et al., *A Practitioner's Guide to Generalized Linear Models – a foundation for theory, interpretation and application*; 3rd edition. Towers Watson, 2007, 122 p.
3. P. Gagnon, Y. Wang, "Robust heavy-tailed versions of generalized linear models with applications in actuarial science," *Computational Statistics & Data Analysis*, vol. 194, pp. 1–16, 2024. doi: 10.1016/j.csda.2024.107920
4. D.K. Lim et al., "Deeply Learned Generalized Linear Models with Missing Data," *Journal of Computational and Graphical Statistics*, vol. 33, no. 2, pp. 638–650, 2024. doi: 10.1080/10618600.2023.2276122

5. Y. Tian, Y. Feng, "Transfer learning under high-dimensional generalized linear models," *Journal of the American Statistical Association*, vol. 118, no. 544, pp. 2684–2697, 2023. doi: 10.1080/01621459.2022.2071278
6. Y. Havrylenko, J. Heger, "Detection of interacting variables for generalized linear models via neural networks," *European Actuarial Journal*, vol. 14, no. 551–580, 2024. doi: 10.1007/s13385-023-00362-4
7. R. Panibratov, P. Bidyuk, "Estimation of the parameters of generalized linear models in the analysis of actuarial risks," *System Research and Information Technologies*, no. 2, pp. 139–148, 2023. doi: 10.20535/SRIT.2308-8893.2023.2.10
8. L. Levenchuk, P. Bidyuk, O. Tymoshchuk, "Operational risk estimation using system analysis methodology," *System Research and Information Technologies*, no. 1, pp. 42–61, 2024. doi: 10.20535/SRIT.2308-8893.2024.1.04
9. C. Andrieu et al., "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, pp. 5–43, 2003. doi: 10.1023/A:1020281327116
10. C. Karras et al., "An overview of mcmc methods: From theory to applications," *Proceedings of international conference on artificial intelligence applications and innovations, IFIP, 2022, Crete, Greece, 17–20 June 2022*, pp. 319–332. Springer International Publishing. doi: 10.1007/978-3-031-08341-9_26
11. N. Alsadat et al, "Bayesian and non-Bayesian analysis with MCMC algorithm of stress-strength for a new two parameters lifetime model with applications," *AIP Advances*, vol. 13, no. 9, pp. 1–20, 2023. doi: 10.1063/5.0167295
12. H. Kavianihamedani, J.D. Quinn, J.D. Smith, "New Diagnostic Assessment of MCMC Algorithm Effectiveness, Efficiency, Reliability, and Controllability," *IEEE Access*, vol. 12, pp. 42385–42400, 2024. doi: 10.1109/ACCESS.2024.3378752
13. J. Zhang et al, "Improving simulation efficiency of MCMC for inverse modeling of hydrologic systems with a Kalman inspired proposal distribution," *Water Resources Research*, vol. 56, no. 3, pp. 1–24, 2020. doi: 10.1029/2019WR025474
14. A. Brown, G.L. Jones, "Convergence rates of Metropolis–Hastings algorithms," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 16, no. 5, pp. 1–15, 2024. doi: 10.1002/wics.70002

*Received 09.01.2025*

## INFORMATION ON THE ARTICLE

**Roman S. Panibratov,** ORCID: 0000-0002-8604-4420, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: roman.panibratov@gmail.com

**Petro I. Bidyuk,** ORCID: 0000-0002-7421-3565, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: pbidyuke_00@ukr.net

**АНАЛІЗ АКТУАРНИХ РИЗИКІВ ЗА ДОПОМОГОЮ УЗАГАЛЬНЕНИХ ЛІНІЙНИХ МОДЕЛЕЙ** / Р.С. Панібратов, П.І. Бідюк

**Анотація.** Розглянуто задачу побудови узагальнених лінійних моделей для аналізу актуарних ризиків із ситуацією виплат премій клієнтам. Для цього застосовано метод Монте-Карло для Марківських ланцюгів. Для дослідження розглянуто дві ситуації. У першій ситуації страхові показники та цільова змінна налаштовувалися випадковим чином через проблему вільного доступу до даних. Для створення трьох наборів даних виплати генерувалися за допомогою нормального, гамма та розподілу Парето зі змінною дисперсією та додаванням шуму для імітації нестаціонарного процесу. У другій ситуації використано реальні актуарні дані, узяті з Singapore Actuarial Society. Побудовано узагальнені лінійні моделі з нормальним розподілом із логарифмічною функцією зв'язку, експоненційним розподілом із логарифмічною функцією зв'язку і розподіл Лапласа з тотожною функцією зв'язку. За метриками якості побудови моделей зроблено висновки щодо їх структури.

**Ключові слова:** актуарний ризик, узагальнені лінійні моделі, імітаційне моделювання, експоненційна множина розподілів, Байєсівський аналіз даних, метод Монте-Карло для Марківських ланцюгів.

UDC 004.932:519.652
DOI: 10.20535/SRIT.2308-8893.2025.4.05

# OVERVIEW OF NEURAL NETWORK OBJECT DETECTION METHODS & MODELES ON THE EXAMPLE OF THEIR USE FOR LAB ANIMAL OBSERVATION

**M.A. SHVANDT, V.V. MOROZ**

**Abstract.** This article provides a brief overview of a set of the most common basic object detection neural network models. Today, the need for automating surveillance and observation processes remains a growing trend. Moreover, one of the key tasks of such processes is usually the detection of an object of interest for further analysis. Previously, many basic object detection algorithms and approaches have been proposed; however, most of them typically have limitations in terms of their applicability. In most cases, these limitations arise due to the nature of the observed environment or because the detection approaches rely on specific object characteristics, such as color or basic shapes only. To address these problems, a new approach for object detection has been developed using neural networks. This paper presents the basis and central aspects of the most common neural network object detection models. The experiment has demonstrated the features, advantages, and disadvantages of the studied methods in the application case of lab animal detection during their behavioral study. Considering this, conclusions and recommendations for their usage cases were made.

**Keywords:** object detection, neural network, neural layer, architecture, model, optimization, estimation, prediction, video, image, frame, background, foreground, experiment, comparison.

## INTRODUCTION

Since the 1990s the fast advancement of computers alongside with the strong development of computer sciences has led to wide automatization of many everyday processes and procedures of our life. From that time and up until today the visual analysis [1] became one of the most used technologies and it is applied everywhere from pedestrian and traffic control to war operations and factory production. In general object detection and tracking usually play important roles in visual analysis. The tasks usually require to detect some object of interest and to track it consequently from frame to frame on either prerecorded video or from some life streaming directly in order to perform some analysis of that object or its behavior.

Object detection and object tracking generally are two separate tasks that require its own special approaches and methods. While some common basic object detection and tracking methods had already been considered in previous articles [2; 3], this time we will take a look on more complex way of object

detection itself. As already mentioned in many cases it can be necessary to detect a specific type of object that has both specific colors and shape/structure. Searching for it with such approaches as, for example, the template matching [2] is not a good option because such operation does not work well with different object scaling and rotations since it requires multiple comparisons of given template with its multiple scaling and rotations in order to find the best matches with objects on image. Such search is not very efficient in terms of performance per frame and is unlikely to be used especially on live video streams of high resolution. Also if object tends to change its shape from frame to frame even slightly, it will affect the comparison with the template and probably will require more templates to check to match each "new" shape. This approach also works mainly with object shape, thus color checking remains as second problem to solve with this approach. Neural detector can come in handy in such cases, as the model coefficients can be trained to recognize a multiple shapes and colors of some particular object. In general the only possible difficulty here can be providing a good dataset for training as it should contain images where the desired type of object can be clearly seen and not mismatched with the background.

One of many processes that can require surveillance and observation automatization is biological research. It often involves the study of life processes of multiple lab animals, for example mice and fish [2; 3]. Usually animals are put in specific conditions so they are easier to observe and note on their behavior. But doing it manually is a time-consuming process that can be automated to save lab personnel some time. The particular case of animal study is gobies behavior observation (Fig. 1). The gobies are kept in a square aquarium with a camera placed right above it recording all their movements during the day to understand the aspects of their activity. While the development of a complex tracking program for its tracking and behavior study is currently being developed, it requires an object detector based on a neural network in order to enhance fish position localization. At this stage in order to choose the best performing detector from the set of most common open-for-use model an experiment was carried to learn which model suits most as such object finder so it can be later integrated into main detection and tracking algorithm. The experiment showed their algorithmic aspects, advantages and disadvantages in case of their application in such test conditions like ours. This analysis might be useful for anyone who plans to use these models in similar conditions as ours and is presented further in this paper.

*a* *b*

*Fig. 1.* Lab fish (gobies) in the study environment (*a*, *b*)

## THE PROBLEM OF NEURAL NETWORK OBJECT DETECTION

The selected models were chosen according to two main criteria. The first is the hardware requirements in terms of performance. The usage of many computational

algorithms on practice is often limited by the hardware it is running on. As fish video analysis is intended to be performed on-site the resulting detection and tracking algorithm should be able to deliver fine performance on usual mass-market hardware instead of cloud servers or big mainframes.

The second criteria also decided to be considered is the possibility to train the model on local mass-market hardware as well. During the experiment (is shown later in this article) it was found out that some model versions could not be trained locally with minimum sufficient image batch size. As for the model acceptable performance it was decided that the batch size of 1 or just 2 images could lead to model poor training.

The third criteria is model detection speed vs accuracy ratio. The models' mAP was evaluated previously [4] with COCO evaluator [5; 6]. While the accuracy is an important feature, the detection speed is also very significant. Since each frame will be additionally preprocessed to remove noise and enhance other color characteristics which will take additional time, running detection on a single frame should not exceed some reasonable time limits as overall video processing should not become several times longer as the video itself. Considering it we took the model versions with highest claimed accuracy that did not exceed the detection time threshold of about 100–110 ms.

An additional difficulty of this experiment is that the objects of interest are gobies which being filmed as mentioned above do not visually contain many significant marks or features compared to other object like cars, other bigger animals or buildings. Thus the lack of visually distinguishable features makes both network training and usage more challenging.

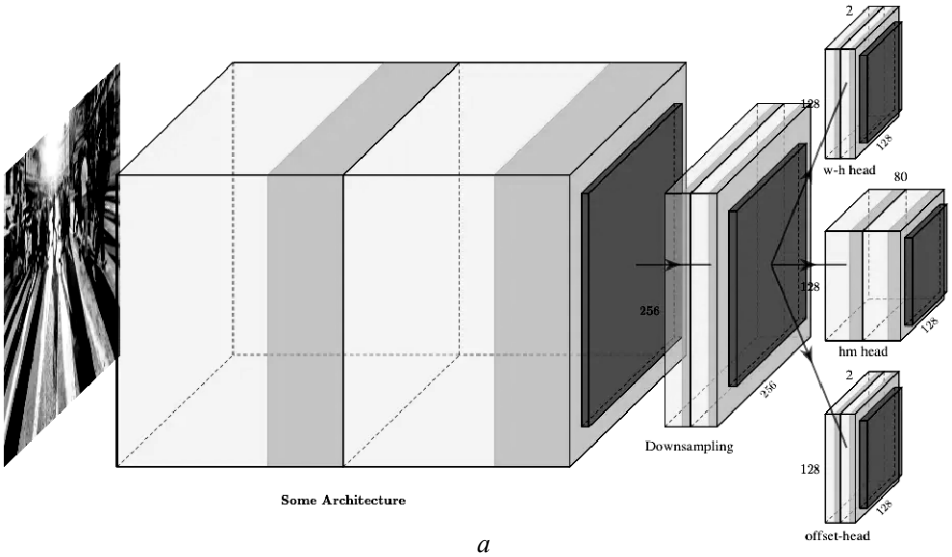
**CenterNet architecture.** The first considered model architecture is the *CenterNet***.** In order to estimate a bounding box of the searched object and to classify it there are two approaches in the Anchor Free Object Detection: the Keypoint-based approach and the *Center-based approach.* The Keypoint-based approach assumes the network predicts the predefined key points and then they are used for bounding box generation around the object and its classification. Examples of such architectures are CenterNet: Keypoint Triplets [7], CornerNet [8], Grid-RCNN [9]. The Center-based approach [7; 10; 11] uses center-point or any part-point of an object to define positive and negative samples. Then it predicts the distance from these positives to four coordinates for the generation of a bounding box. For example such methods as DenseBox [12], FCOS [13], etc. generate positive samples and use them for estimation of boxes and class probabilities.

As for the CenterNet, the main research [10] treats the center of a box as an object as well as a key point. Then it uses this predicted center to detect the coordinates/offsets of the bounding box. Thus the center prediction task is considered as a standard problem for keypoint estimation. When image gets passed through Fully Convolutional Network, the final feature map provides as an output heatmaps for different key points. The peaks of these output feature maps are considered as predicted centers. The network also makes predictions of the width and height of the box for these centers with each center having its unique box width and height. This binding is intended for removing of the Non-Maximal Suppression step in post-processing. The heatmap peaks are also linked to a particular class to which it belongs to and thus it allows object classification, as using these centers, dimensions, and class probabilities, object detection task is achieved.

In general, the CenterNet architecture works in the following way. The input image $I$ having width and height as $W$ and $H$ respectively, and 3 channels for RGB. $R$ is an output stride that will set the resulting dimensions of the given

heads. All the heads will have the same height $H/R$ and width $W/R$, but they will have different $C$ values (depth of the keypoint heatmap). Thus the final head dimensions are ($W/R$, $H/R$, $C$=[<*Classes Num*>/<2>/<2>]). Thus if input dimensions are 512×512, then head dimensions are 128×128 considering stride $R = 4$ (Fig. 2, *a*). The three heads as shown in Fig. 2, *a* are *Heatmap Head, Dimension Head, Offset Head*.

*Heatmap Head* is used for the key points estimation of the given input image. In the case of object detection, keypoints are the box center. One has to predict heatmap $\hat{Y}$ of dimensions ($W/R$, $H/R$, $C$), with $R$ being the output stride, $C$ is the number of classes; $\hat{Y}$ is the functssion of *x, y, c.* A prediction $\hat{Y}(x, y, c) = 1$ corresponds to detected center for that particular class *c*. $\hat{Y}(x, y, c) = 0$ is considered as background. For the loss propagation ground truth heatmaps calculation, these centers are splat using Gaussian Kernels after converting them to low-resolution equivalent (division by stride $R$, denoted as $\tilde{p}$). For example, in case of three classes $C = 3$ and input image dimensions of 400×400, with a given stride $R = 4$ it is necessary to generate 3 heatmaps (as each heatmap corresponds to a given class) of 100×100 dimension as shown in Fig. 2, *b*. The $\sigma$ value used in the kernel is the object-size adaptive standard deviation. Also, if two gaussians of the same class are overlapping, they take element-wise maximum to find the target class.




*a*

*b*

*Fig. 2. a* — Three heads are predicted after one forward pass from the network architecture: Offset Head, Heatmap Head. Dimension Head. Here Some Architecture (FCN) referees to any of the feature extractors which we want to use (specified heads are for object detection; *b* — Left: Ground Truths of different classes, shown in different colors; Right: three centers of respective classes splat into heatmaps using Gaussian Kernel) (source: medium.com/visionwizard [11])

$$Y_{xyc} = \exp\left( -\frac{(x-\tilde{p}_y)^2 + (y-\tilde{p}_y)^2}{2\sigma_p^2} \right).$$

*Dimension Head* is used for the estimation of the dimensions of the boxes width and height. With given box coordinates $(x_1, y_1, x_2, y_2)$ of object *k* and class *c***,** one can regress object sizes $s_k = (x_2 - x_1, y_2 - y_1)$**.** This is achieved by solving a standard $L_1$ distance norm. Dimensions of this heatmap are (*W*/*R, H*/*R,* 2), with *w* being *h* are predicted width and height of the box. To reduce the amount of computation, single sized heatmaps for all object categories are used. *Offset Head* is used to recover from the discretization error caused due to the downsampling of the input. After the center points prediction, one has to map these coordinates to an input image of higher dimension. Since the original image pixel indices are integer values this will cause a value disturbance because one will be predicting the float values. So to solve this issue they make predictions the local offsets $\hat{O}$, as these local offset alues are shared between objects on an image. Offset Head dimensions are **(***W*/*R*, *H*/*R*, 2**)** (here *x* and *y* are the coordinate offsets). The overall detection flow can be seen on Fig. 3, *a*, *b*.

As a *Feature Extractor* CenterNet can use a variety of backbone/feature extraction approaches [8; 10; 11]. With our research we have considered the following ones: *Stacked Hourglass Network* [14] (Hourglass104 version), *Residual Network* [15] (ResNet101 V1 FPN) and the *MobileNet* [16] (MobileNet V2/V1 FPN). For instance the stacked Hourglass Network downsamples the input by 4×, then followed by two sequential hourglass modules, with each hourglass module being made up of a uniform chain of 5-layer down- and up-convolutional network with skip connections. The original paper [10] also used modified (Fig. 3, *c*) ResNet18**,** ResNet1, Deep Layer Aggregation Networks (DLA) [17] with added Deconvolutional and Deformable Convolutional Layers. Standard ResNet modules were extended with three transposed convolutional networks to incorporate higher resolution outputs. Some modifications were done by reducing the output upsampling layers' filters of to 256, 128, and 64 respectively in order to reduce computation. The authors also added an additional 3×3 deformable convolutional layer between each of upsampling layers led to better results on some standard datasets [10; 11].

The main part of CenterNet algorithm is *Loss Calculation and Propagation*. After heatmaps are generated by the network there is a task of loss propagation for training stabilization. In the original paper [8], authors use several loss functions to get over and balance the bias between the training of different heads**.** There are three Loss functions mentioned: *Heatmap Variant Focal Loss, L1 Norm Offset* Loss and *L1 Norm Dimension Size Loss.* For *Heatmap Variant Focal Loss* the Focal Loss function [18; 19] is divided into two parts of positive and negative samples:

$$L_k = \frac{-1}{N} \sum_{xyc} \begin{cases} (1-\hat{Y}_{xyc})^{\alpha} \log(\hat{Y}_{xyc}) & \text{if} \quad Y_{xyc} = 1, \\ (1-\hat{Y}_{xyc})^{\beta} (\hat{Y}_{xyc})^{\alpha} \log(1-\hat{Y}_{xyc}) & \text{otherwise}. \end{cases}$$

If $Y = 1$ when predicted $\hat{Y}$ is close to 1 (ex. $\hat{Y} = 0.95$), it considers as an easy *example (well-classified example)* and thus by the logic of Focal Loss the weightage of the propagated loss will be decreased. The same logic is for *hard examples (misclassified example)* with a difference that instead of decreasing the weight, it will increase the slope of the value by parameter $\alpha$ (here $\alpha := 2$). If

$Y$ != 1 (Otherwise) with predicted $\hat{Y}$ being very close to 0 (ex. $\hat{Y} = 0.005$), then $\hat{Y}^{\alpha}$ will cause the overall loss to be zero, and less weight will be assigned to the propagated loss as stated in the premise of Focal Loss [18]. The particular case is when $\hat{Y}$ is not very close to 0 and has a value near to 1, but it is in the neighborhood of the ground truth heatmap. As the ground truths are the gaussian kernel outputs there is no sudden drop in the values near $Y$=1. It considers values, lying inside gaussian outputs, as possible positives. This is an advantage of this loss. For example, let $\hat{Y} = 0.9$ and being near to the center point peak of ground truth. Here a misclassification takes place as the value should be very near to 0 according to simple logistic regression loss logic. But, as predicted $\hat{Y} \approx 1$, the propagated loss will be less weighted even in a condition of misclassification as

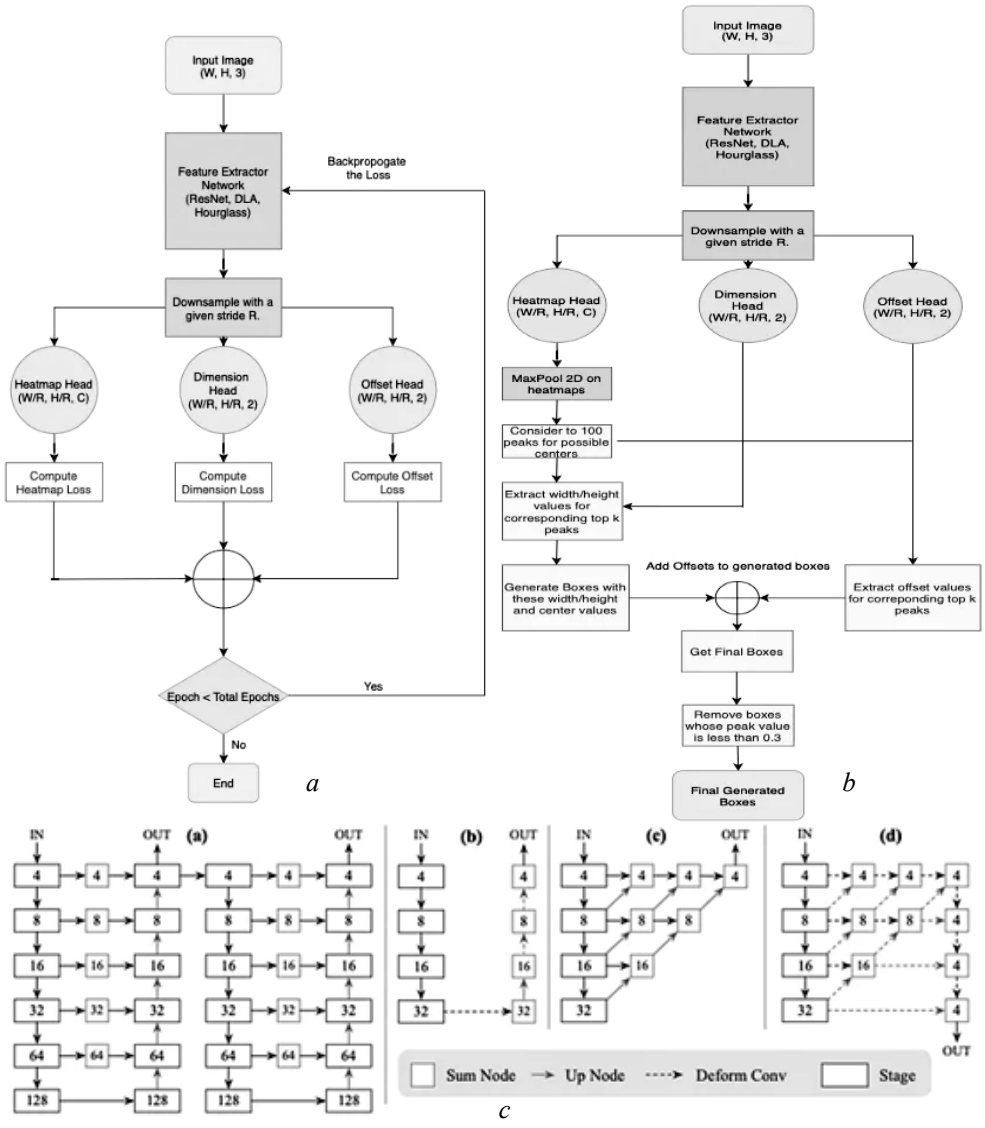



*Fig. 3*. Training — *a*, *b*; Inference flowchart of the explained object detection algorithm — *c*; (here: (a) Stacked Hourglass Network, (b) ResNets with Transposed and Deformable Convolution layers, (c) Original DLA-34, (d) Modified DLA-34 by adding skip connections and Deformable Convolutional Layers.) (source: medium.com/ visionwizard [11])

the loss will be compensated according to the term $(1-Y)^{\beta}$ (value of $Y$ will be close to 1 in a region near center peak). In case when one has $\hat{Y} = 0.9$ and being far from the center point peak, in this condition of misclassification a large loss will be propagated according to term $(1-Y)^{\beta}$ as it does not placed in that splatted region, and the value of $Y$ will be very close to 0. Here $\beta := 4$. The design of this loss function helps to increase the number of positive examples by considering the heatmap values generated by gaussian kernels which, in its turn, help to decrease the bias between positives and negatives.

The *L1 Norm Offset Loss* is a simple L1 Norm of the predicted offset $\hat{O}$ and the ground truth offset values:

$$L_{off} = \frac{1}{N}\sum_{p}\left|\hat{O}_{\tilde{p}} - \left(\frac{p}{R} - \tilde{p}\right)\right|.$$

The meaning of ground truth offset values can be seen on the example: if there is a center point at (18, 22) in an original high-resolution image, when downsampled, with stride size of 4, the mapped coordinates will be (4, 5) on a low-resolution feature map. Here is an offset error of 0.5 in both cases. In the case of keypoint estimation, it becomes important to handle this problem as keypoints are very position sensitive. In order to solve this task the offset loss function is added in order to obtain more accurate results. This supervision only acts at the position of key points, all other locations are ignored.

The *L1 Norm Dimension Size Loss* of the predicted and ground truth width-height coordinates is used for Regression of the width and height of bounding boxes. Here $\hat{S}$ are the predicted dimensions and $s$ are actual ground truth sizes. Raw pixel values are used to calculate the loss instead of normalizing with the feature map size

$$L_{size} = \frac{1}{N}\sum_{k=1}^{N}\left|\hat{S}_{p_k} - s_k\right|.$$

Total Loss propagated by the network is shown in formula, $\lambda_{size} = \dfrac{0.1}{\lambda_{offset}}$.

The *Total Loss* of CenterNet: $L_{\det} = L_k + \lambda_{size}L_{size} + \lambda_{off}L_{off}$.

The example of object detection process is visualized on Fig. 4: during inference process one calculates the peaks of the heatmaps by finding the maximum value near the 8-pixel neighborhood in a heatmap and keeping the first 100 peaks of all the different classes independently. It is achieved by 3×3 MaxPool opera-

*Fig. 4.* Left: keypoint heatmap; middle: keypoint offsets; right: dimensions of box (source: original paper [10])

tion on the resulting feature map with the obtained peak coordinates being used to calculate the dimensions and offset predictions.

**The Residual Network**. As mentioned earlier, CenterNet can use several different backbone architectures for feature extraction. We have considered the models using *Residual Network*, *MobileNet* and *Stacked Hourglass Network*. The first one used in studied models is the *Residual Network* architecture [15; 20; 21]. The Residual Network architecture itself is based on a concept of Residual Blocks. These blocks were designed to address the issue of the *vanishing/exploding gradient*. Inside ResNet a method known as skip connections is applied. This method skips (bypasses) some levels in between link-layer activations to subsequent layers and thus creates a leftover block (Fig. 5, *a*). These leftover blocks are used in stacks to create residual nets (Fig. 6). The main idea of such architecture is to let the network fit the residual mapping instead of having layers learn the underlying mapping and thus, let the network fit instead of using, for example, the initial mapping of $H(x)$:

$$F(x) := H(x) - x \Rightarrow H(x) := F(x) + x\,.$$

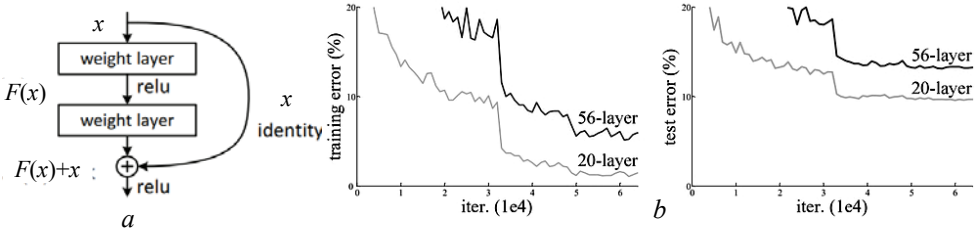



*Fig. 5. a* — Skip connection or Shortcut; *b* — comparison of 26-layer vs 56-layer architecture (source: medium.com/siddheshb008 [20], original paper [15])

Thanks to this skip link, the regularization will skip any layer that worsens the architecture performance and as the training of a very deep neural network becomes possible without getting issues with vanishing or expanding gradients. The general purpose of ResNet is following: in the Deep Neural Networks extra layers are stacked in order to improve accuracy and performance, often to handle a challenging problem. The main idea of layering is that by adding additional layers they will eventually learn features that are more complicated. Ex an example one can take photographs recognition: when recognizing photographs, the first layer may pick up on edges, the second — textures, the third — objects, and so on. However, the traditional convolutional neural network model was found to have the maximum depth threshold. The graphic (Fig. 5, *b*) shows the percentage of errors for training and test data for a 20-layer network and a 56-layer network, respectively.

In both the training and testing situations, we see the higher error percentage for a 56-layer network in comparison with a 20-layer network. It demonstrates that adding additional layers on top of the network will decline its performance. This might be because of with the initialization of the network, the optimization function, and most significantly — because of the vanishing gradient problem. In this case overfitting is not the issue as the 56-layer network's error percentage is the worst on both training and test data, and it does not happen when the model is overfitting.

The ResNet architecture exists in several configurations. For example it can use the VGG-19-inspired 34-layer plain network architecture that is followed by the addition of the shortcut connection. It is subsequently transformed into the residual network by these short-cut connections, as depicted in Fig. 6. Keras, an open-source, Python-based neural network framework offers ResNet V1 and ResNet V2 with 50, 101, or 152 layers: ResNet50, ResNet101, ResNet152, ResNet50V2, ResNet101V2, ResNet152V2; ResNetV2 and the original ResNet (V1) vary primarily in that V2 applies batch normalization before each weight layer. As a brief conclusion, it can be said that ResNet is a robust backbone model and can be used in various computer vision tasks.

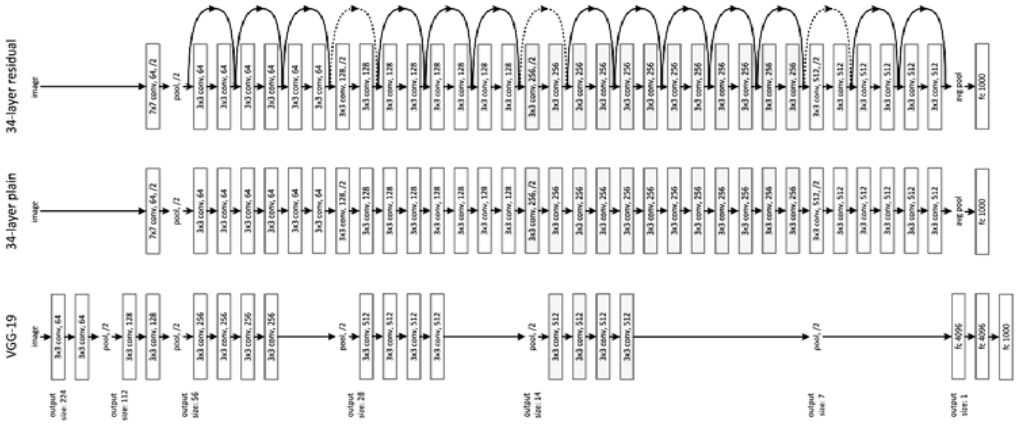
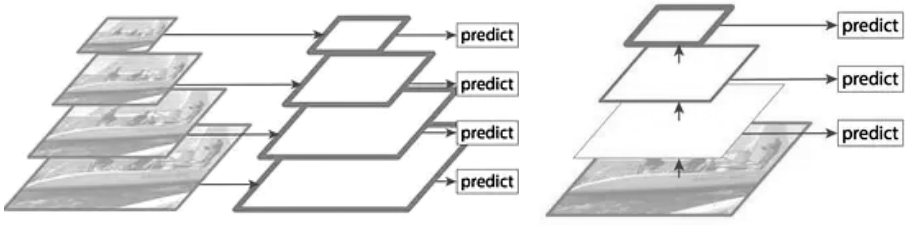



*Fig. 6.* Resnet34 Architecture (source: medium.com/siddheshb008 [20], original paper [15])

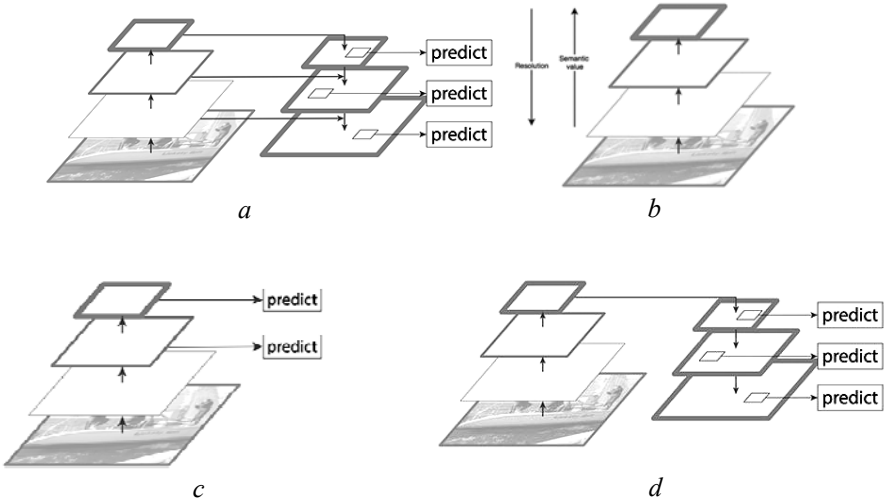
**Feature Pyramid Network (FPN)**. Object detection in different scales is a very difficult task, especially for small objects. The pyramid of the same image at different scale can be used to detect objects (Fig. 7, *a*) but processing multiple scale images too demanding in terms of time and memory to be trained end-to-end simultaneously. That is why it can be used only in inference in order to increase accuracy as high as possible, in particular for cases, when speed is not a concern. As an alternative a feature pyramid can be created and used for object detection (Fig. 7, *b*), but the feature maps, which are closer to the image layer, are composed of low-level structures that are not good for accurate object detection. The Feature Pyramid Network itself is a feature extractor [22–25] created for the described pyramid approach that considers performance speed and accuracy as well. It is used instead of such the feature extractor as Faster R-CNN [26] and generates multi-scale feature maps (multiple feature map layers) and delivers information of better quality than the regular feature pyramid for object detection.




*a* *b*

*Fig. 7. a* — pyramid of images; *b* — pyramid of feature maps (source: medium.com/jonathan-hui [23], original paper [22])

The FPN data flow composes of a *bottom-up* and a *top-down* pathway (Fig. 8, *a*). The bottom-up pathway is represented by a usual convolutional network. During it the features are being extracted: as one goes up, the spatial resolution decreases. After detecting more high-level structures, the *semantic value* for each layer increases (Fig. 8, *b*). The Single Shot MultiBox Detector (SSD) [27] calculates detection from multiple feature maps, but it does not select the bottom layers for object detection (Fig. 8, *c*). Despite being in high resolution their semantic value is not high enough to use it as the speed slow-down is significant. Because of that the SSD uses only upper layers for detection and thus its performance is much worse for small objects. The FPN uses a top-down pathway to construct higher resolution layers from a semantic rich layer (Fig. 8, *d*).




*a* *b* *c* *d*

*Fig. 8. a* — FPN data flow; *b* — feature extraction in FPN; *c* — SSD object detection with top levels; s*d* — FPN top-down pathway (source: medium.com/jonathan-hui [23]; original paper [22])

The reconstructed layers are semantically strong but the objects are not located precisely after all the downsampling and upsampling operations. In order to enhance object location prediction, the lateral connections between reconstructed layers and the corresponding feature maps were added. It also helps to simplify the training as it also acts as skip connections. Similar approach is used in ResNet [15]. For the bottom-up stage the ResNet is used. The bottom-up pathway consists of many convolution modules $Conv_i$, $i=[1,5]$, with each module composing of multiple convolution layers. Also during this stage, one reduces the spatial dimension by 1/2 (i.e. double the stride) with labeling the output of each convolution module as $C_i$ which are later used during in the top-down pathway (Fig. 9, *a*). In the process of top-down pathway one applies a 1×1 convolution filter in order to reduce $C_5$ channel depth to 256-d to obtain $M_5$. Thus, one receives the first feature map layer that will be used for object prediction. With each step down further one upsamples the previous layer by 2 using nearest neighbors upsampling. Again a 1×1 convolution is applied to corresponding feature maps and then they are added element-wise. A 3×3 convolution is applied to all merged layers; this convolution filter is used for reducing the aliasing effect during merging operation with the upsampled layer (Fig. 9, *b*). The same process is repeated for the pyramid feature maps $P_3, P_2$, but it is stopped at $P_2$ because the spatial dimension of $C_1$ is too large. If it is

continued, it will slow down the process too much. As one shares the same classifier and box regressor of every output feature maps, all pyramid feature maps $P_5, P_4, P_3, P_2$ have 256-d output channels.

As for object detection, FPN is not an object detector by itself. This architecture is a feature extractor that works with object detectors. It is used for feature maps extracting and later feeding them into some detector, for example Region Proposal Network (RPN). RPN then applies a sliding window over those feature maps to predict the objectness (i.e. whether there is an object or not) and object boundary box at each location (Fig. 9, *c*). In the FPN framework, for each scale level, for example $P_4$ or $P_3$, one applies 3×3 convolution filter over the feature maps and after that applies separate 1×1 convolution for predictions of objectness and boundary box regression. These 3×3 and 1×1 convolutional layers are called the RPN *head* (Fig. 9, *d*).

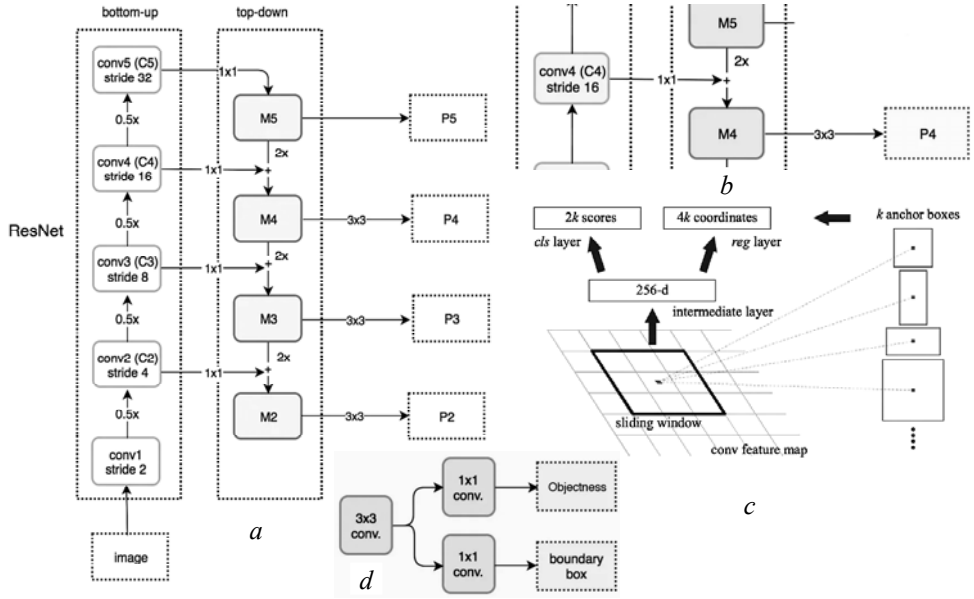



*Fig. 9. a* — ResNet for FPN bottom-up and top-down pathways; *b* — feature merging operation during top-down pathway; *c* — FPN usage with RPN; *d* — RPN head (source: medium.com/jonathan-hui [23])

**MobileNet architecture.** The third considered architecture is MobileNet. This architecture was developed by Google in 2017 [16; 28]. It utilizes the approach called *Depthwise Separable Convolution* in order to reduce the model size and complexity. This architecture was primarily created for use in mobile and embedded vision applications (Fig. 10, *a*). It has following benefits: smaller model size (fewer number of parameters) and smaller complexity (fewer multiplications and additions, aka Multi-Adds). To make MobileNet easy to tune, two parameters were introduced: *Width Multiplier* $\alpha$ and *Resolution Multiplier* $\rho$.

The *Depthwise separable convolution* is a depthwise convolution that is followed by a pointwise convolution (Fig. 10, *b*); the Depthwise convolution is the channel-wise $D_K \times D_K$ spatial convolution. For example, if one has 5 channels, then there are 5 $D_K \times D_K$ spatial convolutions**.** The *Pointwise convolution* actually is the 1×1 convolution intended to change the dimension. Combined with the Depthwise Convolution, the operation cost is:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F ,$$

where the left part of sum is the Depthwise Convolution Cost and the right one is the Pointwise Convolution Cost. Here $M$ is the number of input channels, $N$ is the number of output channels, $D_K$ is kernel size, $D_F$ is the feature map size. For Standard Convolution, its cost is

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F .$$

Thus, the Depthwise Separable Convolution Cost / Standard Convolution Cost is:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2} .$$

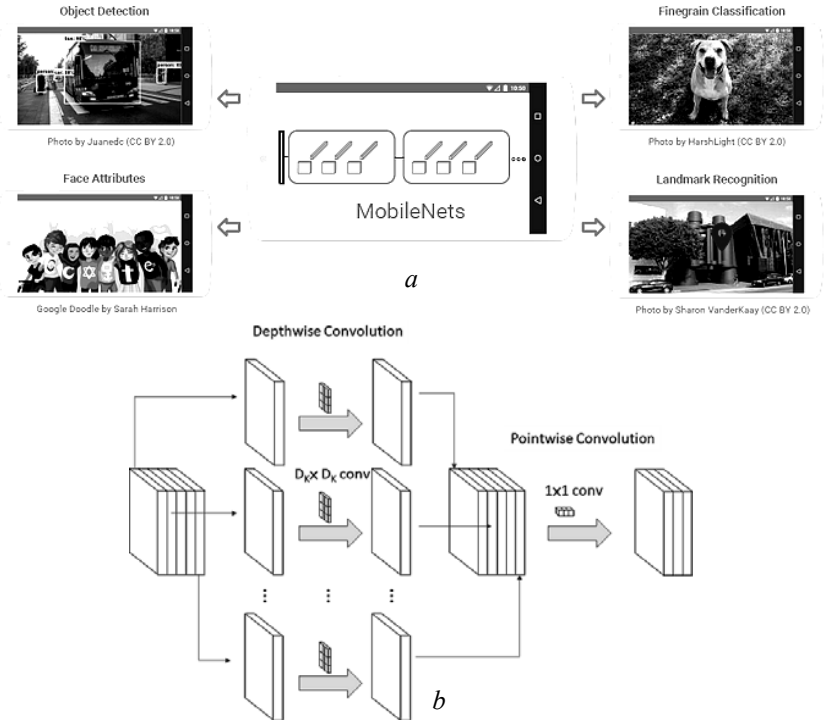



*Fig. 10. a* — MobileNets usage in practice; *b* — Depthwise separable convolution (source: towardsdatascience.com; original paper [16; 28])

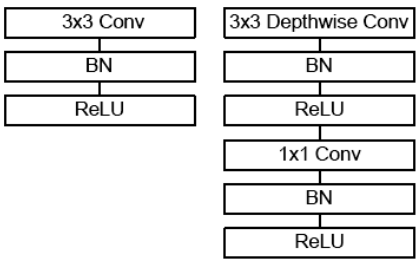



*Fig. 10.* Left: Standard Convolution, right: Depthwise separable convolution (Right) With BN and ReLU (source: original paper [16])

When $D_K \times D_K$ is 3×3, the amount of computation can be reduced from 8 to 9 times, but with only small reduction in accuracy. The Table 1 shows the architecture of MobileNet; the Batch Normalization (BN) and ReLU are applied after each convolution (Fig. 11), with Width Multiplier $\alpha$ being introduced for controlling of the number of channels or channel depth**,** which makes $M$ become $\alpha M$. Thus, the Depthwise Separable Convolution cost (with Width Multiplier $\alpha$) is:

$$D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F ,$$

where $\alpha = [0,1]$, with typical settings of 1, 0.75, 0.5 and 0.25. With $\alpha = 1$, it is the basic MobileNet, and the computational cost and the number of parameters can be reduced quadratically by $\approx \alpha^2$. The Resolution Multiplier $\rho$ is introduced to control the input image resolution of the network and thus the Depthwise Separable Convolution Cost with Both Width Multiplier and Resolution Multiplier is:

$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F ,$$

with $\rho = [0,1]$ and the input resolution of 224, 192, 160, and 128. With $\rho = 1$, it is the basic MobileNe.

**T a b l e 1.** MobileNet Body Architecture (source: original paper [16])

| Type / Stride | Filter Shape | Input Size |
|---|---|---|
| Conv / s2 | 3 × 3 × 3 × 32 | 224 × 224 × 3 |
| Conv dw / s1 | 3 × 3 × 32 dw | 112 × 112 × 32 |
| Conv / s1 | 1 × 1 × 32 × 64 | 112 × 112 × 32 |
| Conv dw / s2 | 3 × 3 × 64 dw | 112 × 112 × 64 |
| Conv / s1 | 1 × 1 × 64 × 128 | 56 × 56 × 64 |
| Conv dw / s1 | 3 × 3 × 128 dw | 56 × 56 × 128 |
| Conv / s1 | 1 × 1 × 128 × 128 | 56 × 56 × 128 |
| Conv dw / s2 | 3 × 3 × 128 dw | 56 × 56 × 128 |
| Conv / s1 | 1 × 1 × 128 × 256 | 28 × 28 × 128 |
| Conv dw / s1 | 3 × 3 × 256 dw | 28 × 28 × 256 |
| Conv / s1 | 1 × 1 × 256 × 256 | 28 × 28 × 256 |
| Conv dw / s2 | 3 × 3 × 256 dw | 28 × 28 × 256 |
| Conv / s1 | 1 × 1 × 256 × 512 | 14 × 14 × 256 |
| 5× Conv dw / s1<br>Conv / s1 | 3 × 3 × 512 dw<br>1 × 1 × 512 × 512 | 14 × 14 × 512<br>14 × 14 × 512 |
| Conv dw / s2 | 3 × 3 × 512 dw | 14 × 14 × 512 |
| Conv / s1 | 1 × 1 × 512 × 1024 | 7 × 7 × 512 |
| Conv dw / s2 | 3 × 3 × 1024 dw | 7 × 7 × 1024 |
| Conv / s1 | 1 × 1 × 1024 × 1024 | 7 × 7 × 1024 |
| Avg Pool / s1 | Pool 7 × 7 | 7 × 7 × 1024 |
| FC / s1 | 1024 × 1000 | 1 × 1 × 1024 |
| Softmax / s1 | Classifier | 1 × 1 × 1000 |

Later the modified MobileNet version was introduced, called V2 [29; 30]. It utilizes the inverted residual structure. In this modification the non-linearities in narrow layers are removed. The difference between V1 and V2 can be briefly described in the following way. The MobileNet V1 has 2 layers, with the first layer, called depthwise convolution, performing lightweight filtering by applying a single convolutional filter per input channel, and the second layer, called pointwise convolution, being a 1×1 convolution and used for building new features through calculating linear combinations of the input channels.

The MobileNet V2 has two types of blocks. One is residual block with stride of 1 and another one is block with stride of 2 used for downsizing; the model has 3 layers for both types of blocks. In this version the first layer is 1×1 convolution with ReLU6, the second layer is the depthwise convolution and the third layer is another 1x1 convolution but without any non-linearity (Table 2). It is also claimed that with using ReLU again, the deep networks only have the power of a linear classifier on the non-zero volume part of the output domain.

**Table 2.** MobileNet V2 layers (source: original paper [29])

| Input | Operator | Output |
|---|---|---|
| $h \times w \times k$ | 1×1 conv2d, ReLU6 | $h \times w \times (tk)$ |
| $h \times w \times tk$ | 3×3 dwise s=$s$, ReLU6 | $h/s \times w/s \times (tk)$ |
| $h/s \times w/s \times tk$ | linear 1×1 conv2d | $h/s \times w/s \times k'$ |

There is also an expansion factor $t$. The authors took $t = 6$ for all main experiments [29; 30]. If the input got 64 channels, the internal output would have $64 \times t = 64 \times 6 = 384$ channels. Table 3 demonstrates the MobileNetV2 Overall Architecture, with $t$ being the mentioned expansion factor, $c$ — number of output channels, $n$ — repeatingnumber, $s$ — stride size; for the spatial convolution 3×3 kernels are used. The authors also note that with the removal of ReLU6 at the output of each bottleneck module, accuracy is improved (Fig. 12, *a*), and with shortcut between bottlenecks, it outperforms shortcut between expansions and the one without any residual connections (Fig. 12, *b*) [29; 30].

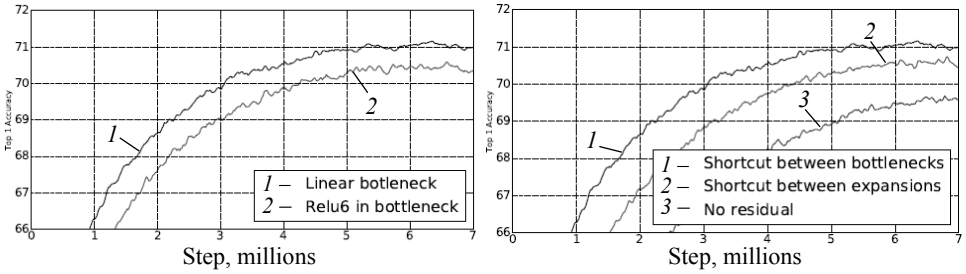



*Fig. 12. a* — Impact of Linear Bottleneck; *b* — impact of Shortcut (source: original paper [29])

**Table 3.** MobileNetV2 Overall Architecture (source: original paper [29])

| Input | Operator | $t$ | $c$ | $n$ | $s$ |
|---|---|---|---|---|---|
| $224^2 \times 3$ | conv2d | - | 32 | 1 | 2 |
| $112^2 \times 32$ | bottleneck | 1 | 16 | 1 | 1 |
| $112^2 \times 16$ | bottleneck | 6 | 24 | 2 | 2 |
| $56^2 \times 24$ | bottleneck | 6 | 32 | 3 | 2 |
| $28^2 \times 32$ | bottleneck | 6 | 64 | 4 | 2 |
| $14^2 \times 64$ | bottleneck | 6 | 96 | 3 | 1 |
| $14^2 \times 96$ | bottleneck | 6 | 160 | 3 | 2 |
| $7^2 \times 160$ | bottleneck | 6 | 320 | 1 | 1 |
| $7^2 \times 320$ | conv2d 1×1 | - | - | 1280 | 1 |
| $7^2 \times 1280$ | avgpool 7×7 | - | - | - | 1 |
| $1 \times 1 \times 1280$ | conv2d 1×1 | - | k | - | - |

**The Hourglass architecture.** The so-called *Hourglass Network* is an architecture that combines a contracting path to extract information and an expanding one to map features into locations [31; 32]. The idea behind its name is that the union of the two paths is usually seen as an hourglass, where information gets narrower before getting expanded again. This architecture takes its beginning from *Fully Convolutional Networks* (FCNs) [34]. Presented in 2015, it is aimed at

modifying the typical structure of Convolutional Neural Networks (CNNs) to obtain segmentation maps as output. As one knows, CNN is a deep network that uses 2 operations: convolution and pooling. A convolution is a mathematical function that, through the use of filters, can extract the presence of different (learned) features in an input. The pooling operation is used to reduce the size of the convoluted matrix, condensing the extracted information (Fig. 13).

The basic structure of a CNN can be seen on Fig. 14. Here convolutions and pooling operations are applied in sequence, resulting in fully-connected layers for the classification of the received input [35]. The FCNs are intended to replace the final fully-connected layers with upsampled versions of the pooling layers output, and thus to retain spatial information and map them into the original input [34]. The FCN also employs the upsampled version of the last pooling layer and combines different upsampling of various pooling layers in the network. Thanks to this the outputs of deeper layers, which contain more information about features, can be combined with the outputs of early layers, which still have information about location [34]. Thus the output of the network's final layer is a segmentation map which is built on information from several previous layers, instead of just the final one. However, this approach is only the basis for hourglass networks. More deeply, the idea of the hourglass architecture can be seen on the following models.

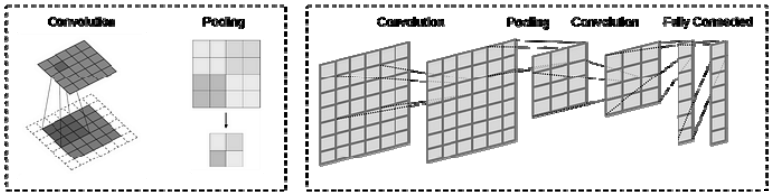



*Fig. 13.* Convolution and Poling operations (source: medium.com/@calleris.enrico [32])

One of further developments of FCNs is an *U-Net* architecture [36]. It is usually considered the earliest example of an hourglass network and was Initially developed for the biomedical images segmentation. U-Net's approach uses the FCN concept to achieve impressive performance, with the main difference between U-Net and FCNs being that the structure of the upsampling operations, which is not a single one anymore but it matches the length of the downsampling path. Now these two paths are now symmetrical and actually lead to model being called U-Net because of the network shape (Fig. 15, *a*). In this architecture after a downward path with the classic CNN structure of sequential convolution-pooling operations, the upward path upsamples the received input and concatenates it with the corresponding layer of the downward path (similar as FCN does). In each upward layer, the previous output gets upsampled and de-convoluted. Then it is combined with the output of the corresponding layer in the downward path (as

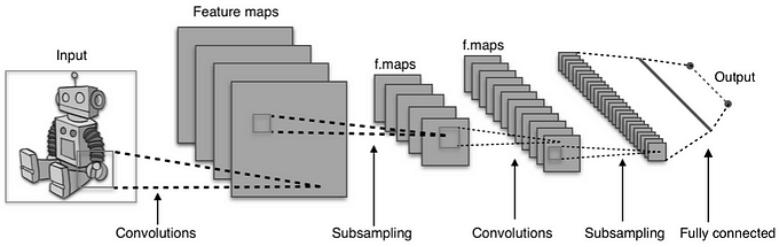
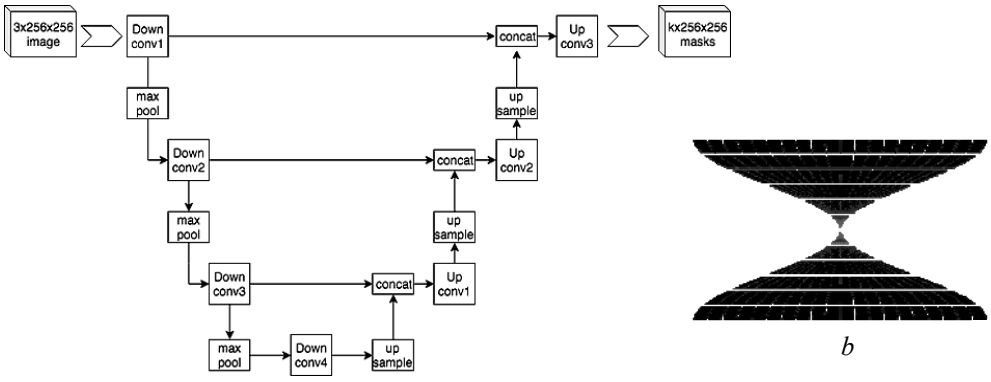



*Fig. 14.* Basic structure of a CNN: Convolutional Operations with Pooling Operations and Fully-Connected Layers. (source: medium.com/@calleris.enrico [32])

shown by the horizontal arrows in Fig. 15, *a*). Thanks to this operation the feature information extracted by the lower layers can be combined with layers with the more spatially-resoluted outputs of the early convolutional layers. Thus, a complete localized feature map is obtained which gives as final output an accurate segmentation map.

Another variant of an hourglass network is the *V-Net*. This model was introduced in 2016 and was intended for segmentation of 3D medical images [38]. This architecture has quite similar approach as U-Net as it is also based on two symmetric contracting and expanding paths. The main difference is that unlike U-Net it uses a fully-convolutional structure where convolution operations are present exclusively and pooling layers are absent. This approach is two-folded because the pooling operation can easily be replaced by a convolution with a larger stride, and thus the network can be trained faster [39]. Also it would be easier to apply the corresponding upsampling operation in the expanding path as de-convolutions are preferred to un-pooling in order to simplify the understanding and analysis [32; 38]. It is also worth mentioning the difference in the training procedure between U-Net and V-Net: the U-Net uses the classic stochastic gradient descent, while V-Net utilizes residual connections [15] to make convergence faster and improve the segmentation results.

But despite all highlighted differences, U-Net and V-Net still share similar approach, based on two different interconnected paths: the downward (contracting) path for progressive feature extraction from the scene and the upward (expanding) path for mapping of the extracted features to specific locations in the original image [37]. The overall concept shows why it was called a hourglass as it mimicks the two paths that contract and expand, meeting midway to form the hourglass shape (Fig. 15, *b*). The output of this type of architecture is as an accurate segmentation map, and it still is one of the most used approaches in computer vision [40].




*a* *b*

*Fig. 15. a* — U-Net architecture with downward path extracting features (left) and an upward path mapping them into locations (right); *b* — conceptualization of the Hourglass Network, with the contracting path (up) and the expanding path (down) (source: medium.com/@calleris.enrico [32])

**EfficientDet architecture.** EfficientDet is a family of object detection models. As model efficiency is very important in computer vision, a lot of research has been made in recent years towards more accurate object detection [41]. But one knows that the more accurate object detection network is more expensive in terms of number of parameters (FLOPS) it gets. The most simple and straightfor-

ward way to increase the accuracy of object detection network is either to make the network deeper by increasing the number of layers, or increase the number of channels, or increase the model input image resolution. However, the random increase of any one among the dimensions mentioned above will diminish the accuracy gain. So in EfficientDet paper [42] authors introduced a systematic way of model scaling and they show that carefully balancing network depth, width and resolution can lead to better performance.

The initial concept of model scaling, i.e. increase of the network depth, width and resolution to enchance its performance, was presented in the EfficientNet paper [43] for image classification, but in the result of EfficientNet testing the authors implement this technique for object detection and called it as EfficientDet [41; 42]. This architecture is based on the paradigm of one-stage detectors: these detectors use ImageNet-pretrained EfficientNets as the backbone network. Thus, the Bidirectional Feature Pyramid Network (BiFPN) was introduced and it serves as the feature network by taking level 3–7 features (P3, P4, P5, P6, P7) from the backbone network and repeatedly applying top-down and bottom-up bidirectional feature fusion. These fused features are then fed to a class and box network predictor, in order to generate object class and bounding box predictions respectively (Fig. 16).

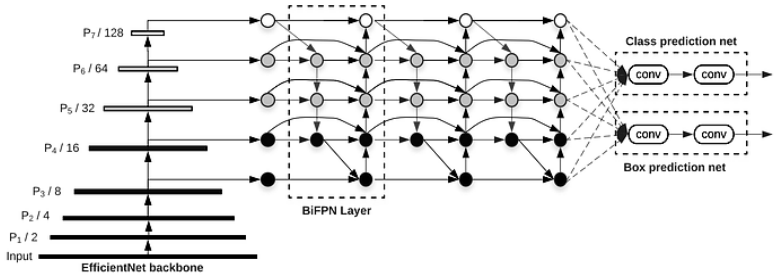
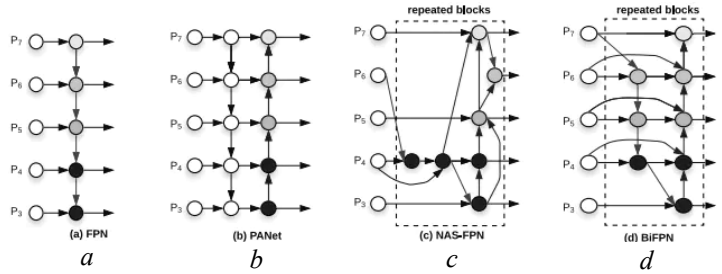



*Fig. 16.* EfficientDet architecture: it uses EfficientNet as the backbone network, BiFPN as the feature network and shared class/box prediction network. (source: original paper [42])

The development of BiFPN can be seen on Figure 17 [42; 45]. Here, FPN [22] is a baseline way to fuse features with a top down flow; Path Aggregation Network (PA Net) [45] allows the feature fusion to go backwards and forwards from smaller to larger resolution; NAS-FPN [46] is also another feature fusion technique created earlier. The EfficientDet architecture uses the edited structure of NAS-FPN to create on the BiFPN blocks and stacks them on top of each other with number of blocks varying in the model scaling procedure. Also, considering that certain features and feature channels might vary in the amount that they contribute to the end prediction, a set of weights was added at the beginning of the channel that are learnable. Before EfficientDet, model scaling for image detection generally scaled portions of the network independently, as for example, the ResNet scales only the size of the backbone network. This idea is similar to the joint scaling approach used to create EfficientNet. For EfficientDet the scaling task was set to vary the size of the backbone network, the BiFPN network, the class/box network, and the input resolution. The backbone network scales up directly with the pretrained checkpoints of EfficientNet-B0 through EfficientNet-B6 and its width and depth are varied along with the number of BiFPN stacks [44].




*a* *b* *c* *d*

*Fig. 17.* Feature network design: *a* — FPN [23] introduces a top-down pathway to fuse multi-scale features from level 3 to 7 (P3 - P7); *b* — PANet [26] adds an additional bottom-up pathway on top of FPN; *c* — NAS-FPN [10] use neural architecture search to find an irregular feature network topology and then repeatedly apply the same block; *d* — is our BiFPN with better accuracy and efficiency trade-offs. (source: original paper [42])

The specifications of EfficientDet are the following: the *Backbone Network* uses the same width/depth scaling coefficients as EfficientNet-B0 to B6 so that ImageNet-pretrained checkpoints can be used. The *BiFPN network* width (number of channels) grows exponentially as done in EfficientNets, but the depth (number of layers) is increased linearly since it needs to be rounded to small integers. After a grid search, 1.35 is detected as best scale factor for width:

$$W_{bifpn} = 64 \cdot (1.35^{\varphi}), \quad D_{bifpn} = 3 + \varphi .$$

*Box/class prediction network* has the same width as the BiFPN but the depth is linearly increased:

$$D_{box} = D_{class} = 3 + \lfloor \phi / 3 \rfloor .$$

As for input image resolution: since feature levels 3–7 are used in BiFPN, the input resolution must be dividable by $2^7 = 128$, so resolutions are increased linearly.

$$R_{input} = 512 + \phi \cdot 128 .$$

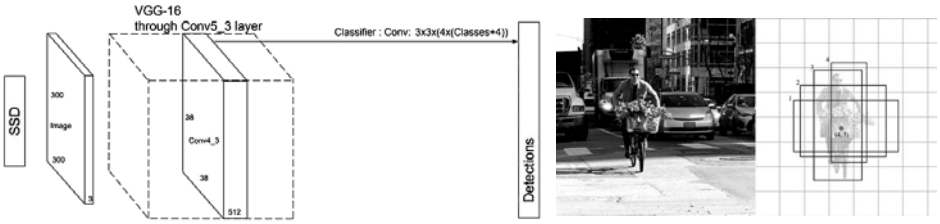
**Table 4.** EfficientDet scaling (source: original paper [42])

| Depth | Input size $R_{input}$ | Backbone Network | BiFPN #channels $W_{bifpn}$ | BiFPN #layers $D_{bifpn}$ | Box/class #layers $D_{class}$ |
|---|---|---|---|---|---|
| D0 ($\phi = 0$) | 512 | B0 | 64 | 3 | 3 |
| D1 ($\phi = 1$) | 640 | B1 | 88 | 4 | 3 |
| D2 ($\phi = 2$) | 768 | B2 | 112 | 5 | 3 |
| D3 ($\phi = 3$) | 896 | B3 | 160 | 6 | 4 |
| D4 ($\phi = 4$) | 1024 | B4 | 224 | 7 | 4 |
| D5 ($\phi = 5$) | 1280 | B5 | 288 | 7 | 4 |
| D6 ($\phi = 6$) | 1280 | B6 | 384 | 8 | 5 |
| D7 ($\phi = 7$) | 1536 | B6 | 384 | 8 | 5 |
| D7*x* | 1536 | B7 | 384 | 8 | 5 |

**Single Shot MultiBox Detector (SSD).** The SSD [27; 47] is a detector designed for real-time object detection. While Faster R-CNN [26] utilizes an RPN for boundary box creation and uses those boxes for object classification, despite its accuracy it does not perform very fast (up to 7 fps) and thus is not suitable for

real-time object detection. The SSD performs faster due to elimination of need for RPN and maintains fine accuracy by using multi-scale features and default boxes as improvements. This allows SSD to increase its speed using images with lower resolution and keep its performance at the same level of accuracy as Faster R-CNN. This fact was confirmed by our experiment, as shown in the model performance comparison at the end of this paper.

*Single Shot MultiBox Detector* composes of 2 parts: feature maps extraction and application of convolution filters for object detection. For feature maps extraction the *VGG16* architecture [48; 49] is used and it detects objects with help of *Conv4_3* layer (Fig. 18, *a*). As for example (Fig. 18, *b*), a $8 \times 8$ Conv4_3 is drawn spatially (should be 38×38) and for each cell (aka location) it produces 4 object predictions. Each prediction is represented by a boundary box and 21 scores for each class (plus 1 extra class for no object) and one picks the highest score as the bounded object's class. The Conv4_3 in total produces 38×38×4 predictions, four predictions per cell regardless of the depth of the feature maps. Since many of these predictions contain no object, SSD reserves a class "0" to mark that the box has no objects (Fig. 19, *a*). The process of making multiple predictions containing boundary boxes and confidence scores is called multibox [47].




*a* *b*

*Fig. 18. a* — General SSD architecture idea; *b* — detector work: left: the original image, right: 4 predictions at each cell (source: article [47])

As for predictors for object detection, SSD does not use a special RPN; instead, it calculates both the location and class scores with help of *small convolution filters*. After feature maps extracting, the detector applies 3×3 convolution filters for each cell to make predictions with these filters calculating predictions in the same way as regular CNN filters. The output of each filter consists of 25 channels: 21 scores for each class + one boundary box (ex. applying four 3×3 filters in Conv4_3 for 512 input channels mapping gives 25 output channels) (Fig. 19, *b*).

$$(38 \times 38 \times 512) \xrightarrow{(4 \times 3 \times 3 \times 512 \times (21+4))} (38 \times 38 \times 4 \times (21+4)).$$

For detection, SSD uses multiple layers (*multi-scale feature maps*) to detect objects independently. The CNN gradually reduces the spatial dimension and thus the resolution of the feature maps also decreases. SSD uses lower resolution layers to detect larger scale objects: for example, the 4×4 feature maps are used for larger scale objects (Fig. 20, *a*). Also, SSD adds 6 more auxiliary convolution layers after the VGG16, with 5 of them added for object detection. In three of those layers, one makes 6 predictions instead of 4 and in total, SSD makes 8732 predictions using 6 layers (Fig. 20, *b*). Thanks to usage of Multi-scale feature maps the accuracy is significantly improved.


*Fig. 19.* Multibox predictor — *a*; convolutional predictors for object detection (source: article — *b* [47])

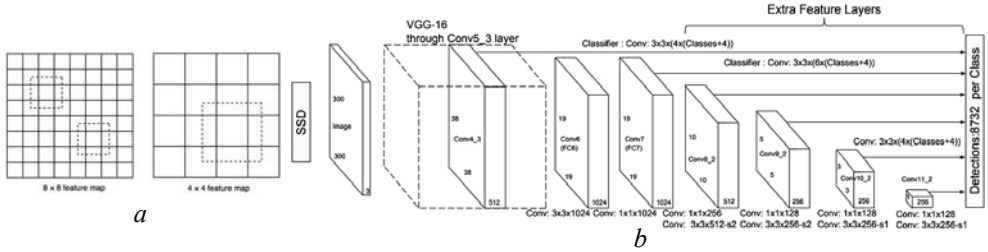



*Fig. 20.* Lower resolution feature maps (right) detects larger-scale objects — *a* [47]; SSD full architecture — *b* [27; 47]

SSD also uses the idea of *default boundary boxes* that are equivalent to *anchors* in Faster R-CNN [26]. For prediction of boundary boxes one can start with random predictions and use gradient descent for model optimization. During the initial training, however, there can be a problem of determining what shapes (cars or pedestrians) may be optimized for which predictions and research showed that early training can be very unstable. The boundary box predictions on Fig. 21, *a* work well for one category but not for others and it is necessary for initial predictions to be diverse and not looking similar. So for detection of number of objects the predictions have to cover more shapes. Such approach makes training easier and more stable (Fig. 21, *b*).

In real-life, boundary boxes do not have arbitrary sizes and shapes, so the ground truth boundary boxes can be partitioned into clusters with each cluster being represented by a default boundary box, i.e., by the centroid of the cluster. In this way instead of making random predictions one can start the guesses based on those default boxes. The SSD detector also keeps the default boxes to a minimum (4 or 6) with one prediction per default box. As for boundary box localization, its predictions do not use global coordinates; instead, they are relative to the default boundary boxes at each cell ($\Delta cx$, $\Delta cy$, $\Delta w$, $\Delta h$), i.e. the offsets (difference) to the default box at each cell for its center ($cx$, $cy$), width and height. Each feature map layer shares the same set of default boxes centered at the corresponding cell, but different layers use different sets of default boxes to adjust object detections at different resolutions (Fig. 21, *c*).

SSD's default boundary boxes are chosen manually and network defines a scale value for each feature map layer. As it was shown on Fig. 20, *b*, starting from the left, Conv4_3 detects objects at the smallest scale of 0.2 or sometimes

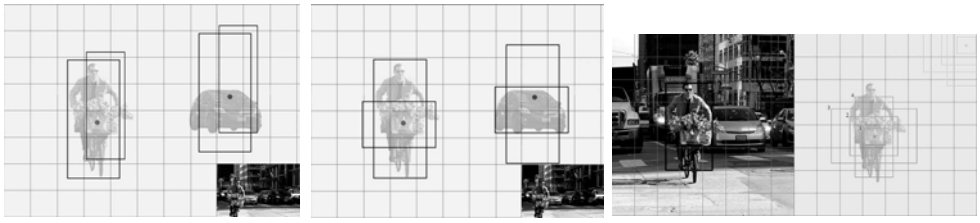
even 0.1. Then, it linearly increases to the rightmost layer at a scale of 0.9. After that one calculates width and height of the default boxes by combining the scale value with the target aspect ratios. As layers make 6 predictions, SSD starts with 5 target aspect ratios of 1, 2, 3, 1/2, and 1/3. Then the width and the height of the default boxes are calculated with $\text{aspect ratio} = 1$. YOLO [52] uses *k*-means clustering on the training dataset to determine those default boundary boxes.

$$w = scale \cdot \sqrt{\text{aspect ratio}},$$

$$h = \frac{scale}{\sqrt{\text{aspect ratio}}};$$

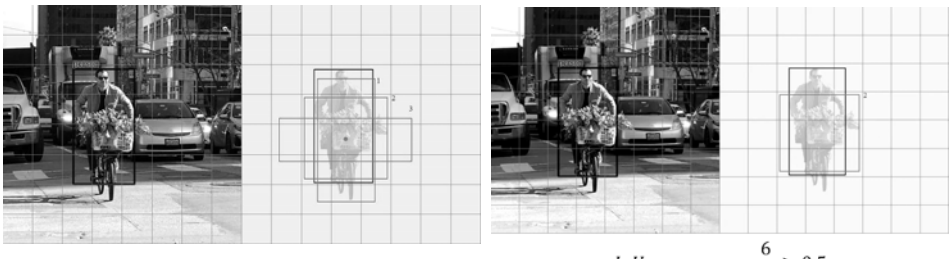
SSD adds an additional default box with scale:

$$scale = \sqrt{\text{scale} \cdot \text{scale at next level}}.$$



*a* *b* *c*

*Fig. 21.* With predictions not being diverse, the model will not perform — *a*; diverse predictions cover more object types — *b*; 4 default boundary boxes — c [47]




*a* *b*

*Fig. 22.* The ground truth object (blue) and 3 default boundary boxes (green) — *a*; default box 2 has IOU > 0.5 with the ground truth — *b* [47]

With SSD's *matching strategy*, predictions are classified as **positive** matches or negative matches. SSD uses only positive matches for localization cost calculation (the mismatch of the boundary box) and if the corresponding *default boundary box* (and not the predicted boundary box) has an $IOU > 0.5$ with the ground truth, then the match is considered positive; otherwise, it is negative. *IOU (Intersection over Union)* is a metric used to measure the degree of overlap between two bounding boxes and it calculates the ratio of the area of overlap between the two boxes to the area of their union. Mathematically, it is represented as $IOU = S_{\text{intersection}} / S_{\text{union}}$. For example, if there are 3 default boxes and only default box 1 and 2 (not 3) have an $IOU > 0.5$ with the ground truth box above (blue box, Fig. 22, *a*). Then only box 1 and 2 are positive matches and once the positive matches are identified, one calculates the cost using the corresponding

predicted boundary boxes (Fig. 22, *b*). Such matching strategy encourages each prediction to predict shapes closer to the corresponding default box, and in this way the predictions are more diverse and stable in the training [47]. Fig. 23 shows how SSD uses the combination of multi-scale feature maps and default boundary boxes to detect objects at different scales and aspect ratios. The dog matches one default box (in red) in the 4×4 feature map layer, but no other default boxes in the higher resolution 8×8 feature map, while the cat, because of being smaller, is detected only by the 8×8 feature map layer in 2 default boxes (in blue). Higher-resolution feature maps are responsible for detecting small objects and the first layer for object detection, Conv4_3, has a spatial dimension of 38×38 which is large reduction from the input image. That is way SSD usually performs badly for small objects comparing with other detection methods but this problem can be reduced by using images with higher resolution.

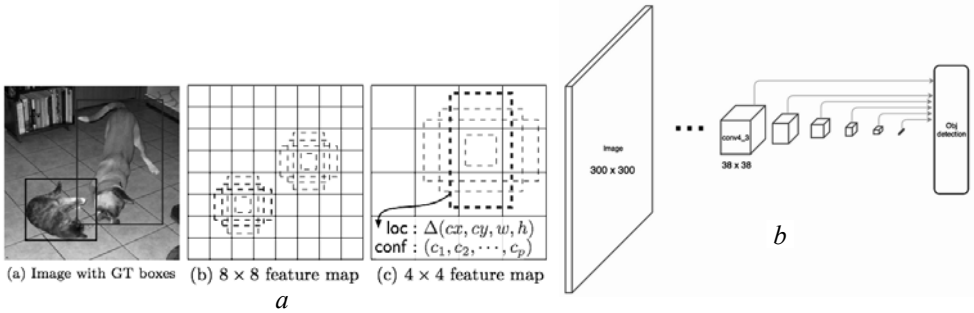



*Fig. 23.* SSD detector work example — *a* [27]; size reduction — *b* [47]

SSD's *localization loss* is the mismatch between the ground truth box and the predicted boundary box and SSD only penalizes predictions from positive matches. It is necessary for the predictions from the positive matches to get closer to the ground truth; negative matches can be ignored. This loss is defined as the smooth *L1* loss with *cx, cy* as the offset to the default bounding box *d* with width *w* and height *h* [27]:

$$L_{loc}(x,l,g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k} \text{smooth}_{L1}(l_i^m - \hat{g}_j^m);$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w, \quad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h;$$

$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right), \quad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right);$$

$$x_{ij}^p = \begin{cases} 1, & \text{if IOU} > 0.5 \text{ between default box } i \text{ and} \\ & \text{ground true box } j \text{ on class } p, \\ 0, & \text{otherwise.} \end{cases}.$$

SSD also uses *confidence loss as* the loss of making a class prediction. It penalizes the loss according to the confidence score of the corresponding class for every positive match prediction. For negative match predictions, it penalizes the loss according to the confidence score of the class "0": class "0" means no object

is detected. The loss is the softmax loss over multiple classes confidences *c* (class score), and *N* is the number of matched default boxes:

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p} \log(\hat{c}_i^p) - \sum_{i\in Neg} \log(\hat{c}_i^0), \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}.$$

The final loss function is calculated as:

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)),$$

where *N* is the number of positive matches and $\alpha$ is the weight for the localization loss.

For the removal of duplicate predictions pointing to the same object SSD uses non-maximum suppression. It sorts the predictions by the confidence scores and starting from the top confidence prediction, SSD evaluates whether any previously predicted boundary boxes have (Intersection Over Union) $IOU > 0.45$ with the current prediction for the same class and if found, current prediction will be ignored. Despite that, there are still much more predictions made than the number of objects present, sothere are many more negative matches than positive matches. This way the class imbalance problem appears and it worsens the training as the model then is training to learn background space rather than detecting objects. But SSD still requires negative sampling so it can recognize what represents a bad prediction and thus it sorts those negatives by their calculated confidence loss instead of using all of them. It takes the negatives with the top loss and makes sure the ratio between the picked negatives and positives is at most 3:1, which makes the training process faster and more stable [47].

**Faster R-CNN architecture.** Faster R-CNN is an object detection architecture presented [26; 50; 51] in 2015. It is one of the famous object detection architectures that use convolution neural networks like SSD (Single Shot Detector) or YOLO (You Look Only Once) [52]. The idea behind the development of Faster R-CNN network was to create a unified architecture that not only detects objects within an image but also locates the objects precisely in the image. Faster R-CNN architecture uses the benefits of deep learning, CNNs and RPNs resulting in a combined network that significantly improves the speed and accuracy of the model [50]. It consists of two key components: Region Proposal Network (RPN) and Fast R-CNN detector and as a backbone it utilizes Shared Convolutional Layers, common CNN layers used for both RPN and Fast R-CNN detector (Fig. 24, *a*)

Faster R-CNNs backbone, the CNN is used for extraction of relevant features from the input image and consists of multiple convolutions layers that apply different convolutions kernel to extract those features. The convolutions kernels are designed to capture the hierarchical representations of the input image. This means that starting from the initial layers, CNN captures the low-level features (basic textures or edges) and then with much deeper layers it captures the high level semantic features like objects parts and shapes. Both RPN and Fast R-CNN detector uses the same extracted hierarchical features. This approach helps to significantly reduce the computing time and memory use as the computations performed by these layers are employed for both tasks.

Previously R-CNN and Fast R-CNN architectures use *Selective Search algorithm* [53] region proposals generation. This process is executed on CPU and thus takes more time in computations. With the introduction of Faster R-CNN

[26] this problem was fixed by using a convolutional-based network i.e. RPN. This step reduced proposal time for each image from 2 seconds to 10 ms and improved feature representation by sharing layers with detection stages. Region Proposal Network is a key component of Faster R-CNN architecture, as it is responsible for generating possible ROIs (regions of interest or region proposals) in images that may contain objects. It is based on the concept of attention mechanism in neural networks that tells the subsequent Fast R-CNN detector where in the image the objects should be searched for. The main components of the Region Proposal Network are as follows [50]:

1. *Anchors boxes*: Anchors are used for region proposals generation in the Faster R-CNN model. It uses a set of predefined anchor boxes with different scales and aspect ratios and these anchor boxes are placed at different positions on the feature maps. An anchor box's two key parameters are scale and aspect ratio

2. *Sliding Window approach***:** The RPN runs as a sliding window over the feature map received from the CNN backbone. It uses a small convolutional network, usually a 3×3 convolutional layer, to process the features within the sensitive field of the sliding window and thus this convolutional operation produces scores that represent the object presence probability and regression values for adjusting the anchor boxes.

3. *Objectness Score***:** This value represents the probability that a given anchor box contains an object of interest rather than being just background. RPN predicts this score for each anchor and this objectness score reflects the confidence that the anchor corresponds to a meaningful object region. This score is also used for anchor classification as either positive (object) or negative (background) during training.

4. *IOU (Intersection over Union)* metric.

5. *Non-Maximum Suppression (NMS)***:** This operation is used to remove redundant and select the most accurate proposals, based on the mentioned objectness scores of overlapping proposals and it keeps only proposals with the highest score while eliminating the others.

The RPN uses feature maps the were produced by CNN backbone and applies on them a sliding window approach (as it was shown in Fig. 9, *c*) with anchor boxes of varying scales and shapes to marks potential object positions. This way these anchor boxes are enhanced in the process of training in order to match better actual object positions and sizes. For each anchor, the RPN predicts two parameters: *objectness score*, i.e. probability that anchor contains object, and adjustments for the anchor coordinates to match the actual object's shape. As a large number of region proposals are generated and many of them overlap and correspond to the same object, Non-Maximum Suppression is used to rank the anchor boxes according to their objectness probabilities and take only the top-*N* anchor boxes with the highest scores. Thus it can be guaranteed that the final selected proposals are both accurate and non-overlapping and algorithm considers these selected anchor boxes as as possible region proposals.

The next key component of Faster R-CNN network is the Fast R-CNN detector itself as it is responsible for object detection within the region proposals suggested by the RPN. It operates in several stages [50]:

1. *Region of Interest (RoI) Pooling:* at this first step ROI pooling is applied to the region proposals suggested by the RPN. This operation transforms RPN's variable-sized region proposals into fixed-size feature maps that will be handed

into the network's subsequent layers. RoI pooling divides each region proposal into a grid of cells of equal size and then applies max pooling within each cell, thus, generating a fixed-size feature map for each region proposal (Fig. 24, *b*).

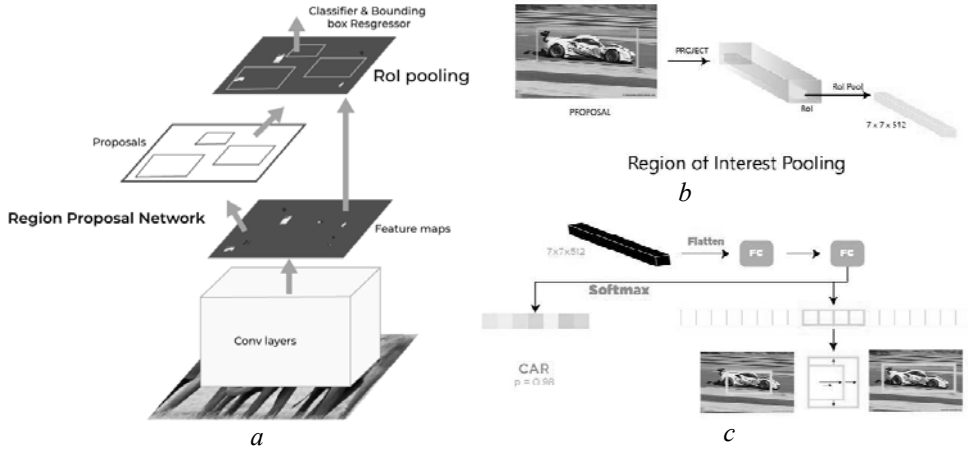



*Fig. 24.* Faster R-CNN architecture idea — *a*; ROI Pooling —*b*; bounding box regression — c [26; 50]

2. *Feature Extraction:* in this stage feature maps, obtained after ROI Pooling, are fed into the CNN backbone (the same one used in the RPN for feature extraction) in order to extract meaningful features that capture object-specific information and thus one draws hierarchical features from region proposals. Hierarchical features keep the spatial information while separating low-level details and thus allow the network to understand the content of proposed regions.

*3. Fully Connected Layers:* the algorithm passes the RoI-pooled and feature-extracted regions through a series of fully connected layers as they are used for object classification and bounding box regression.

*a*. As for *Object Classification*, network predicts class probabilities for each region proposal and points out the possibility that the proposal contains an object of a specific class, and then, the classification is performed by combining the features pulled out from the region proposal with the shared weights of the CNN backbone.

*b. Bounding Box Regression:* Together with class probabilities, the network predicts bounding box adjustments for each region proposal, which refine the position and size of the region proposal's bounding box and align it more accurately with the actual object boundaries. The first layer is a softmax layer of *N+1* output parameters that predicts the objects in the region proposal, with *N* being the number of class labels and background. The second layer is a bounding box regression layer with $4 \cdot N$ output parameters and is used for bounding box regression of the object in the image.

Fast R-CNN detector uses the *Multi-task Loss Function* as the loss function [26]. It combines classification and regression losses, with classification loss calculating the difference between predicted and true class probabilities and the regression loss calculating the difference between predicted and actual bounding box adjustments:

$$L(p_i,t_i,v_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i,p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i,v_i)$$

where, $N_{cls}$ is the number of ROIs used for classification, $N_{reg}$ is the number of ROIs used for bounding box regression; $p_i$ is the predicted probability of classifying the *i-th* ROI; $p_i^*$ is binary (1 or 0) ground-truth indicator for the *i-th* ROI being a foreground or background object; $t_i$ represents the ground-truth bounding box parameters for the *i-th* ROI; $v_i$ represents the predicted bounding box adjustments for the *i-th* ROI; $L_{cls}$ is the classification loss function, usually computed using cross-entropy loss; $L_{reg}$ is the regression loss function, usually computed using smooth *L1* loss and $\lambda$ is a balancing parameter for controling the trade-off between the two components of the loss.

After predicting class probabilities and bounding box changes, the final detection results are refined using a *post-processing procedure*. In this step, non-maximum suppression (NMS) is used to reduce redundant detections while keeping the most confident and non-overlapping ones.

Several additional notes on Faster R-CNN Process can be made [50]. For Fast R-CNN object detection network two fully convolution models are commonly used: Zeiler and Fergus model (ZF) [54] with 5 shareable convolutional layers or Simonyan and Zisserman model (VGG-16) [48] with 13 shareable convolutional layers. With the Sliding Window Approach, RPN operates on an $n \times n$ spatial window of the input convolutional feature map with each sliding window being mapped to a lower-dimensional feature. Its dimensions are 256-d for Zeiler and Fergus model (ZF) and 512-d for Simonyan and Zisserman model (VGG-16). Further it is followed by a Rectified Linear Unit (ReLU) activation. The sliding window architecture is effectively realized using an $n \times n$ convolutional layer, followed by two $1 \times 1$ convolutional layers for box regression and box classification; if for example, $n = 3$ for the sliding window, it leads to a large effective receptive field on the input image: 171 pixels for ZF and 228 pixels for VGG). For each window position, $K$ region proposals are generated with proposal being defined by an *anchor box*, which is set by scale and aspect ratio. Multiple *anchor boxes* are created by varying these parameters and it results in different scales and aspect ratios. Thus a set of anchor boxes is created, usually $K = 9$, allowing the model to consider various object sizes and shapes. These anchor variations allow the model to handle scale invariance and share features between the RPN and Fast R-CNN.

For each generated region proposal, a feature vector is extracted with a length of 256 (for ZF net) or 512 (for VGG-16 net) and is then processed by two sibling fully-connected (FC) layers: the lower-dimensional feature extracted from the sliding window is fed into two sibling fully-connected layers. The *Box-Classification FC Layer (cls)* predicts an objectness score for the proposed region, it is a binary classifier that assigns an objectness score to each region proposal and determines whether the proposal contains an object or is part of the background. This layer also produces two outputs: one for classifying the region as background and another for classifying the region as an object. The objectness score assigned to each anchor helps to generate the classification label. The *Box-Regression FC Layer (reg)* predicts adjustments for the bounding box of the proposed region and returns a 4-D vector that defines the bounding box of the region proposal.

The Region Proposal Network (RPN) is trained end-to-end using backpropagation and stochastic gradient descent (SGD), i.e. entire network, including the newly added RPN layers and the shared convolutional layers, is optimized together to minimize the loss function. The training strategy is an "image-centric" sampling [50], in which each mini-batch is derived from a single image. This image contains both positive (with object) and negative (background) example anchors and instead of optimizing the loss function for all anchors, the network randomly picks 256 anchors from the image to calculate the loss for the mini-batch. The sampled positive and negative anchors are balanced at a ratio of up to 1:1. Also in order to overcome the potential bias towards negative samples, the training makes sure that each mini-batch contains a *balanced mix* of positive and negative examples. Thus if an image has less than 128 positive samples, additional negative samples are added to the mini-batch to keep the correct ratio.

Regarding the layer initialization, new layers added to the architecture are initialized by getting weights from a Gaussian distribution with a mean of zero and a standard deviation of 0.01. This random initialization is applied to the layers specific to the RPN. The existing shared convolutional layers are initialized using weights pretrained on the ImageNet classification task according to standard practice.

**YOLO detector.** You Only Look Once (YOLO) [52] is an architecture, that uses end-to-end neural network and allows to makes predictions of bounding boxes and class probabilities all at once [55]. The main difference from other object detection algorithms is that they repurposed classifiers to make detections. YOLO model is based on fundamentally different object detection approach and thus it performs extremely fast, significantly outperforming other real-time object detection algorithms. This detector does all of its predictions with the help of a single fully connected layer in contrast to other networks, like Faster RCNN, which detect possible regions of interest with help of RPN and then do recognition on those regions separately. In other words, for the same image YOLO does a single iteration when RPN-baset networks perform multiple iterations.

The YOLO architecture [52; 55] is depicted on Fig. 25: the algorithm receives an image as input, then detects objects on this image with help of a simple deep convolutional neural network as its backbone. In this model the first 20 convolution layers are pre-trained using ImageNet by plugging in a temporary average pooling and fully connected layer and then, this pre-trained model is converted to perform detection, because thanks to earlier studies it was carried out that adding convolution and connected layers to a pre-trained network helps to improve performance. The model's final fully connected layer makes predictions on both class probabilities and bounding box coordinates.

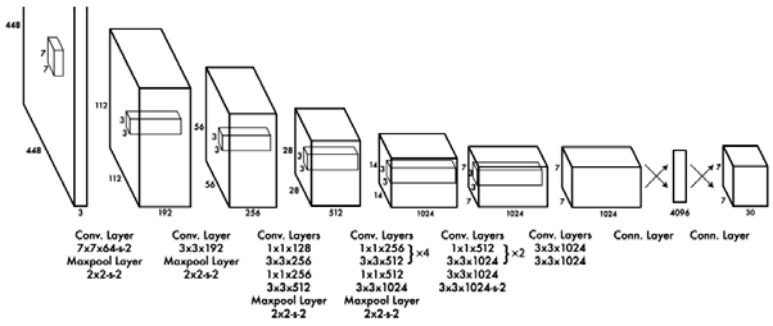



*Fig. 25.* YOLO original architecture (source: original paper [52; 55])

During processing, YOLO's algorithm divides an input image into an $S \times S$ grid. Within this grid it checks, if the center of an object gets placed within some grid cell, then that grid cell is responsible for detecting that object. Thus, each grid cell predicts $B$ bounding boxes and confidence scores for those boxes and these confidence scores indicate how confident the model is that some box contains an object and how accurate the model thinks that predicted box is. The algorithm predicts multiple bounding boxes per grid cell. As at training stage it is necessary that for each object only one bounding box predictor should be responsible for this object, YOLO assigns one predictor to be "responsible" making predictions of an object based on which prediction has the highest current IOU with the ground truth. This task leads to bounding box predictors being specialized for different tasks: each predictor gets better at forecasting certain sizes, aspect ratios, or classes of objects, improving the overall recall score. YOLO models also use one key approach called *Non-Maximum Suppression (NMS)*, a post-processing step used for improvement of object detection accuracy and efficiency. In the process of object detection it is a common situation when for a single object in an image multiple bounding boxes are generated. As such bounding boxes can be located at different positions or may overlap while still representing the same object, NMS is used to identify and remove redundant or incorrect bounding boxes and to output a single correct bounding box for each object in the image [55].

Since the initial release of YOLO in 2015, it had received several modifications and improvements and thus new versions have appeared. For our research the YOLO v5 version was chosen as it is still one of the most recent modifications and is easy to install for instant usage. The v5 version [56] was introduced in 2020 by developers of original YOLO. Unlike the original model, v5 version uses EfficientDet-based architecture as a backbone. It allowed the new model to increase accuracy and generalization to a wider range of object categories. The v5 version also uses a new method for generating the anchor boxes, called "dynamic anchor boxes", which involves using a clustering algorithm to group the ground truth bounding boxes into clusters and then it uses the centroids of the clusters as the anchor boxes. Thanks to this the anchor boxes can be more closely aligned with the size and shape of detected objects. One more new idea that was introduced in YOLO v5 is the concept of the *Spatial Pyramid Pooling (SPP)*. It is a type of pooling layer used to reduce the spatial resolution of the feature maps. It allows the model to see the objects at multiple scales and thus improves the detection performance on small objects. In addition, v5 model has introduced a new variant of the IOU loss function called "CIOU Loss" and designed to improve the model's performance on imbalanced datasets [55].

**Training and experiment.** The whole experiment was performed locally on a laptop with an Intel i9-13980HX CPU, NVidia 4090 16GB laptop GPU and 64GB RAM; CUDA 11.2. Due to hardware limitations the training was decided to be limited by 5000 train steps for general comparison. The test runs for each model were performed on the same machine. The dataset used for training consisted of 562 images (selected video frames). The test set consisted of 63 images. The part of test is not only to evaluate the trained model accuracy on train dataset, but to see how fast the models train considering their different internal architecture. As mentioned previously, we did not measure FPS because each video frame

is additionally preprocessed to improve its color and remove noise, and thus this processing takes some additional time.

Also because of hardware limitations (GPU memory size) each model was trained with maximum possible image batch size. For example, we did not consider the EfficientDet D3 or higher versions since on current hardware it was possible only to train with a batch of 4 which was considered not enough for good model training. Also as the CenterNet MobileNetV2 FPN 512×512 training results were very poor despite 5.5h of training it was decided to exclude it from the comparison.

Note on detection speed benchmark: for the first image, the detection time was always much longer comparing to other images in test set due to the model being loaded by program for the first time. Thus we excluded the detection time for the first image from average detection speed calculation as this delay occurs for the very first image only and is insignificant for the further detection period. Fig. 26 shows some examples of work of the neural network object detector.

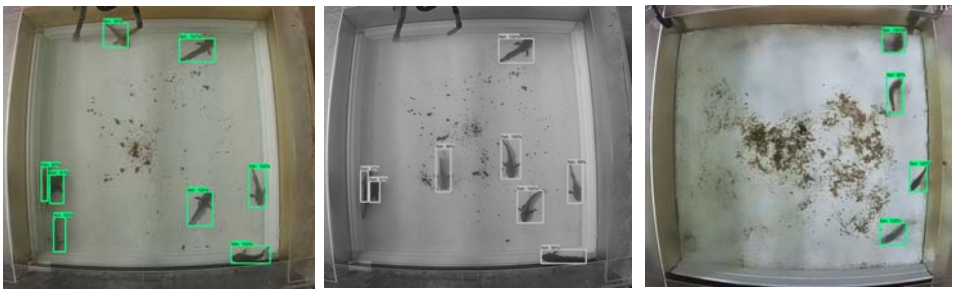


*Fig. 26.* NN object detection results

**CONCLUSIONS**

The experiment results are shown on Table 5. According to them, it was carried out, that such models as the EfficientDet, SSD net with ResNet50 V1, ResNet101 V1 backbone and the Faster R-CNNs are not very suitable for training on usual hardware as they operate with large amounts of data during training and thus require a lot of memory when trying larger batch sizes which can become problematic. Using smaller batch size can cause model not to train good enough even despite that its training time was sometimes less compared to other models. Extending the training time also cannot be good option as seen EfficientDet 10K step training. As for other mentioned models (SSDs and Faster R-CNNs), with the same training amount of 5K steps these models also showed smaller accuracy.

The detection speed benchmark results also show that for example, only CenterNet Resnet101 V1 FPN 512x512, SSD MobileNet V1 FPN and YOLO v5 manage to fit within the given detection time threshold of about 110 ms. Other SSD nets managed to show longer detection time with a bit smaller accuracy then the SSD MobileNet V1 FPN.

Thus, for our further research it is decided to consider the SSD MobileNet V1 FPN and YOLO as the most suitable for detection of relatively simple objects with minimum visual details. This result might be useful for anyone trying to use any of studied models for similar tasks. From all tested model only these ones can be trained with minimum significant amount of data and perform accurate and fast enough even at less capable hardware then used for our experiment. Thus we

will consider them as primary neural object detectors for fish detection and tracking. But we still might consider CenterNet with HourGlass104 and Resnet101 V1 FPN for some other tasks as they also maintain relatively good ratio of detection speed and especially training possibilities and resulting accuracy.

**Table 5.** Model training and evaluation results

| Architecture | Backbone | Network input size | Training batch size | Training time (approx.) | AP (%) | AP50 IOU (%) | AP75 IOU (%) | AP(M) (%) | AP(L) (%) | Avg detection speed (ms) |
|---|---|---|---|---|---|---|---|---|---|---|
| CenterNet | Hour-Glass104 | 512x512 | 10 | 5.5 h | 79.16 | 99.23 | 93.64 | 62.02 | 80.9 | 183 |
| CenterNet | Resnet101 V1 FPN | 512x512 | 20 | 5 h | 74.59 | 98.19 | 91.05 | 58.01 | 76.31 | 83 |
| Efficient-Det D2 | Efficient-Net | 896x896 | 6 | 3 h | 55.83 | 92.31 | 62.23 | 32.65 | 58.32 | 233 |
| Efficient-Det D2 | Efficient-Net | 896x896 | 6 (10K steps) | 3 h | 65.47 | 93.9 | 82.41 | 45.31 | 67.44 | 263 |
| SSD | MobileNet V1 FPN | 640x640 | 32 | 10 h | 87.42 | 99.03 | 98.96 | 75.58 | 88.8 | 112 |
| SSD | MobileNet V2 FPNLite | 640x640 | 28 | 4.5 h | 73.43 | 97.5 | 89.95 | 55.31 | 75.22 | 116 |
| SSD | ResNet50 V1 FPN | 1024x1024 | 8 | 7.5 h | 73.53 | 96.6 | 88.62 | 47.13 | 76.17 | 141 |
| SSD | ResNet101 V1 FPN | 1024x1024 | 5 | 3.5 h | 23.29 | 44.47 | 24.45 | 3.04 | 25.46 | 175 |
| Faster R-CNN | ResNet50 V1 | 800x1333 | 6 | 2 h | 71.41 | 98.5 | 88.89 | 48.0 | 73.67 | 143 |
| Faster R-CNN | ResNet101 V1 | 1024x1024 | 5 | 3 h | 70.85 | 97.65 | 88.72 | 51.48 | 72.85 | 241 |
| | | | | | AP50-95 | | | Precision | Recall | |
| Yolo V5 | CSPDark-net53 | 640 x 640 | 80 | 1.5 h | 98.5 | 98.9 | - | 98.1 | 97.7 | 8 |

**Acknowledgements.** Special thanks for the research assistance and provided test videos and images of lab animals to Faculty of Biology of Odesa I.I. Mechnikov National University.

## REFERENCES

1. A.R. Smith, “Color Gamut Transform Pairs,” in *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pp. 12–19,1978. doi: 10.1145/800248.807361
2. M.A. Shvandt, V.V. Moroz, “Overview Of The Detection And Tracking Methods Of The Lab Animals”, *System Research & Information Technologies*, no. 1, 2022, pp. 124–148. doi: 10.20535/SRIT.2308-8893.2022.1.10
3. V.V. Moroz, M.A. Shvandt, “Study of movement and behavior of laboratory animals by methods of object detection and tracking”, *Herald of the National Technical University ‘KhPI’, Series of ‘Informatics and Modeling’, Kharkiv: NTU ‘KhPI’, Kharkiv*, vol. 13, no. 1338, pp. 93–103, 2019. doi: 10.20998/2411-0558.2019.13.09
4. TensorFlow 2 Detection Model Zoo. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md
5. T.-Y. Lin et al., Microsoft COCO: *Common Objects in Context*. 2014, 15 p. doi: 10.48550/ARXIV.1405.0312. Available: https://arxiv.org/abs/1405.0312
6. L. Wood, F. Chollet, *Efficient Graph-Friendly COCO Metric Computation for Train-Time Model Evaluation*. 2022, 7 p. doi: 10.48550/ARXIV.2207.12120. Available: https://arxiv.org/abs/2207.12120

7. K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, *CenterNet: Keypoint Triplets for Object Detection*. 2019, 10 p. doi: 10.48550/ARXIV.1904.08189. Available: https://arxiv.org/abs/1904.08189
8. H. Law, J. Deng, *CornerNet: Detecting Objects as Paired Keypoints*. 2018, 14 p. doi: 10.48550/ARXIV.1808.01244. Available: https://arxiv.org/abs/1808.01244
9. X. Lu, B. Li, Y. Yue, Q. Li, J. Yan, *Grid R-CNN. 2018*, 9 p. doi: 10.48550/ARXIV.1811.12030. Available: https://arxiv.org/abs/1811.12030
10. X. Zhou, D. Wang, P. Krähenbühl, *Objects as Points*. 2019, 12. doi: 10.48550/ARXIV.1904.07850. Available: https://arxiv.org/abs/1904.07850
11. S. Trivedi, *CenterNet: Objects as Points - A Comprehensive Guide*. 2020. Available: https://medium.com/visionwizard/centernet-objects-as-points-a-comprehensive-guide-2ed9993c48bc
12. L. Huang, Y. Yang, Y. Deng, Y. Yu, *DenseBox: Unifying Landmark Localization with End to End Object Detection*. 2015, 13 p. doi: 10.48550/ARXIV.1509.04874. Available: https://arxiv.org/abs/1509.04874
13. Z. Tian, C. Shen, H. Chen, T. He, *FCOS: Fully Convolutional One-Stage Object Detection*. 2019, 13 p. doi: 10.48550/ARXIV.1904.01355. Available: https://arxiv.org/abs/1904.01355
14. A. Newell, K. Yang, J. Deng, *Stacked Hourglass Networks for Human Pose Estimation*. 2016, 17 p. doi: 10.48550/ARXIV.1603.06937. Available: https://arxiv.org/abs/1603.06937
15. K. He, X. Zhang, S. Ren, J. *Sun, Deep Residual Learning for Image Recognition*. 2015, 12 p. doi: 10.48550/ARXIV.1512.03385. Available: https://arxiv.org/abs/1512.03385
16. A.G. Howard et al., *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017, 9 p. doi: 10.48550/ARXIV.1704.04861. Available: https://arxiv.org/abs/1704.04861
17. D. Wang, E. Shelhamer, T. Darrell, *Deep Layer Aggregation*. 2017, 10 p. doi: 10.48550/ARXIV.1707.06484. Available: https://arxiv.org/abs/1707.06484
18. T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. *Dollár, Focal Loss for Dense Object Detection*. 2017, 10 p. doi: 10.48550/ARXIV.1708.02002. Available: https:// arxiv.org/abs/1708.02002
19. S. Trivedi. *Understanding Focal Loss-A Quick Read*. 2020. Available: https:// medium.com/visionwizard/understanding-focal-loss-a-quick-read-b914422913e7
20. S. Bangar, Resnet Architecture Explained. 2022. Available: https://medium.com/@siddheshb008/resnet-architecture-explained-47309ea9283d
21. P. Ruiz, *Understanding and visualizing ResNets*. 2018. Available: https://towards-datascience.com/understanding-and-visualizing-resnets-442284831be8
22. T.-Yi Lin et al., *Feature Pyramid Networks for Object Detection*. 2016, 10 p. doi: 10.48550/ARXIV.1612.03144. Available: https://arxiv.org/abs/1612.03144
23. J. Hui, *Understanding Feature Pyramid Networks for object detection (FPN)*. 2018. Available: https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c
24. S.-H. Tsang, *Review: FPN - Feature Pyramid Network (Object Detection)*. 2019. Available: https://towardsdatascience.com/review-fpn-feature-pyramid-network-object-detection-262fc7482610
25. S. Tanwar, FPN (*feature pyramid networks*). 2020. Available: https://medium.com/analytics-vidhya/fpn-feature-pyramid-networks-77d8be41817c
26. S. Ren, K. He, R. Girshick, J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2015, 14 p. doi: 10.48550/ARXIV.1506.01497. Available: https://arxiv.org/abs/1506.01497
27. W. Liu et al., *SSD: Single Shot MultiBox Detector*. 2015, 17 p. doi: 10.48550/ARXIV.1512.02325. Available: https://arxiv.org/abs/1512.02325
28. S.-H. Tsang, Review: MobileNetV1 - Depthwise Separable Convolution (Light Weight Model). 2018. Available: https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69
29. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2018, 14 p. doi: 10.48550/ARXIV.1801.04381. Available: https://arxiv.org/abs/1801.04381v4

30. S.-H. Tsang, Review: *MobileNetV2 - Light Weight Model* (*Image Classification*). 2019. Available: https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c
31. A. Newell, K. Yang, J. Deng, *Stacked Hourglass Networks for Human Pose Estimation*. 2016, 17 p. doi: 10.48550/ARXIV.1603.06937. Available: https://arxiv.org/abs/1603.06937
32. E. Calleris, *The Hourglass Network*. 2022. Available: https://medium.com/@calleris.enrico/hourglass-network-6e74cdb9ce2f
33. S. Li, *Simple Introduction about Hourglass-like Model*. 2017. Available: https://medium.com/@sunnerli/simple-introduction-about-hourglass-like-model-11ee7c30138
34. J. Long, E. Shelhamer and T. Darrell, *Fully Convolutional Networks for Semantic Segmentation*. 2014, 10 p. doi: 10.48550/ARXIV.1411.4038. Available: https://arxiv.org/abs/1411.4038
35. A. Krizhevsky, I. Sutskever, G.E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - NIPS'12, Curran Associates Inc., Red Hook, NY, USA, 2012*, vol 1., pp. 1097–1105.
36. O. Ronneberger, P. Fischer, T. Brox, U-Net: *Convolutional Networks for Biomedical Image Segmentation*. 2015, 8 p. doi: 10.48550/ARXIV.1505.04597. Available: https://arxiv.org/abs/1505.04597
37. S. Minaee et al., *Image Segmentation Using Deep Learning: A Survey*. 2020, 22 p. doi: 10.48550/ARXIV.2001.05566. Available: https://arxiv.org/abs/2001.05566
38. F. Milletari, N. Navab, S.-A. Ahmadi, *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation*. 2016, 11 p. doi: 10.48550/ARXIV.1606.04797. Available: https://arxiv.org/abs/1606.04797
39. J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, *Striving for Simplicity: The All Convolutional Net*. 2015, 14 p. doi: 10.48550/ARXIV.1412.6806. Available: https://arxiv.org/abs/1412.6806
40. D. Oñoro-Rubio, M. Niepert, *Contextual Hourglass Networks for Segmentation and Density Estimation*. 2018, 3 p. doi: 10.48550/ARXIV.1806.04009. Available: https://arxiv.org/abs/1806.04009
41. R. Sharma, *EfficientDet: Scalable and Efficient Object Detection*. 2021. Available: https://medium.com/analytics-vidhya/efficientdet-scalable-and-efficient-object-detection-384a5df9011a
42. M. Tan, R. Pang, Q.V. Le, *EfficientDet: Scalable and Efficient Object Detection*. 2019, 10 p. doi: 10.48550/ARXIV.1911.09070. Available: https://arxiv.org/abs/1911.09070
43. M. Tan, Q.V. Le, *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020, 11 p. doi: 10.48550/ARXIV.1905.11946. Available: https://arxiv.org/abs/1905.11946
44. J. Solawetz, *A Thorough Breakdown of EfficientDet for Object Detection*. 2020. Available: https://towardsdatascience.com/a-thorough-breakdown-of-efficientdet-for-object-detection-dc6a15788b73
45. S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, *Path Aggregation Network for Instance Segmentation*. 2018, 11 p. doi: 10.48550/ARXIV.1803.01534. Available: https://arxiv.org/abs/1803.01534
46. C. Peng et al., *MegDet: A Large Mini-Batch Object Detector*. 2017, 9 p. doi: 10.48550/ARXIV.1711.07240. Available: https://arxiv.org/abs/1711.07240
47. J. Hui, *SSD object detection: Single Shot MultiBox Detector for real-time processing*. 2018. Available: https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06
48. K. Simonyan, A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014, 14 p. doi: 10.48550/ARXIV.1409.1556. Available: https://arxiv.org/abs/1409.1556
49. J. Boschman, *VGG16* (*2014*) *– one minute summary*. 2021. Available: https://medium.com/one-minute-machine-learning/very-deep-convolutional-networks-for-large-scale-image-recognition-2014-one-minute-summary-44a8f04586ab
50. Faster *R-CNN – ML*. Available: https://www.geeksforgeeks.org/faster-r-cnn-ml/

51. A. Khazri, *Faster RCNN Object detection*. 2019. Available: https://towardsdatascience.com/faster-rcnn-object-detection-f865e5ed7fc4
52. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*. 2016, 10 p. doi: 10.48550/ARXIV.1506.02640. Available: https://arxiv.org/abs/1506.02640
53. J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, A.W.M. Smeulders, "*Selective Search for Object Recognition", in International Journal of Computer Vision*, 2013, 14 p. doi: 10.1007/s11263-013-0620-5
54. M. D. Zeiler and R. Fergus, *Visualizing and Understanding Convolutional Networks*. 2013, 11 p. doi: 10.48550/ARXIV.1311.2901. Available: https://arxiv.org/abs/1311.2901
55. R. Kundu, YOLO: *Algorithm for Object Detection Explained [+Examples]*. 2023. Available: https://www.v7labs.com/blog/yolo-object-detection
56. S.-H. Tsang, Brief Review: *YOLOv5 for Object Detection*. 2023. Available: https://sh-tsang.medium.com/brief-review-yolov5-for-object-detection-84cc6c6a0e3a

*Received 25.12.2023*

## INFORMATION ON THE ARTICLE

**Maksym A. Shvandt,** ORCID: 0000-0002-4580-3961, Odesa I.I. Mechnikov National University, Ukraine, e mail: maxim.shvandt@gmail.com

**Volodymyr V. Moroz,** ORCID: 0000-0002-3240-4590, Odesa I.I. Mechnikov National University, Ukraine, e mail: v.moroz@onu.edu.ua

**ОГЛЯД МЕТОДІВ І МОДЕЛЕЙ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ НА БАЗІ НЕЙРОННИХ МЕРЕЖ НА ПРИКЛАДІ ЇХ ЗАСТОСУВАННЯ ДЛЯ СПОСТЕРЕЖЕННЯ ЗА ЛАБОРАТОРНИМИ ТВАРИНАМИ** / М.А.Швандт, В.В. Мороз

**Анотація.** Наведено стислий огляд найпоширеніших базових моделей нейронних мереж для виявлення об'єктів. Потреба в автоматизації процесів спостереження та нагляду постійно зростає. Одним із ключових завдань таких процесів є виявлення об'єкта, що цікавить, для подальшого його аналізу. Було запропоновано багато основних алгоритмів і підходів виявлення об'єктів, але більшість із них, зазвичай, мають деякі обмеження щодо області застосування. Здебільшого ці обмеження зумовлені характером спостережуваного середовища або через те, що підходи до виявлення залежать від окремих характеристик об'єкта, як-от лише колір або деякі основні форми. Для вирішення цих проблем був розроблений загалом новий підхід до виявлення об'єктів із використанням нейронних мереж. Подано основи та основні аспекти найбільш поширених моделей нейронних мереж для виявлення об'єктів. Експеримент продемонстрував особливості, переваги та недоліки досліджуваних методів при застосуванні для виявлення лабораторних тварин під час вивчення їх поведінки. З огляду на це зроблено висновки та надано рекомендації щодо їх використання.

**Ключові слова:** детектування об'єктів, нейронна мережа, нейронний шар, архітектура, модель, оптимізація, оцінка, прогноз, відео, зображення, кадр, задній фон, передній фон, експеримент, порівняння.

UDC 519.816; 004.032.26; 004.9; 004.85
DOI: 10.20535/SRIT.2308-8893.2025.4.06

# QUALITY ASSESSMENT OF MODELS AND DEEP LEARNING METHODS FOR SUPER-RESOLUTION IMAGE FORMATION

**N. NEDASHKOVSKAYA, A. LANKO**

**Abstract.** This article examines evaluation metrics for the results of super-resolution image generation in solving the SISR task. The study comprises two experiments: the implementation of custom network architectures for SRGAN, VDSR, and SRCNN, and fine-tuning of pre-trained SRGAN, VDSR, and SRCNN models. An algorithm for assessing the quality of models and deep learning methods for generating super-resolution images is suggested. The VDSR model performed best in terms of pixel, structural, and perceptual metrics, as well as training time and visual confirmation by a human, highlighting that residual learning is more effective than recursive learning under the conditions of the two conducted experiments. Threshold values for practically acceptable and high-quality results were determined through visual analysis of many generated images and their corresponding quality metrics, including those reported by other researchers.

**Keywords:** single image super-resolution, quality assessment, generative models, deep learning methods, convolutional neural network, residual learning, recursive learning, fine-tuning of pre-trained models, perceptual metric, LPIPS, multicriteria decision analysis, DIV2K dataset, thresholds for practically acceptable and high-quality generated images.

## INTRODUCTION

The task of Single Image Super-Resolution (SISR) involves the formation of highly detailed versions of low-resolution images [1]. Despite significant progress in modern imaging technologies, this task remains relevant due to such factors as image quality deterioration after transmission through communication channels and hardware failures, image compression for compact storage on data carriers, and the inability to use professional equipment in certain natural conditions.

The goal of SISR methods is to create high-quality images by restoring or adding details missing in the original low-resolution images. To achieve this, generative models and deep learning methods are used [2].

Generative models form new parts by simulating the data distribution in the training selection [2]. Among them, the most common for SISR are modifications of generative adversarial networks (GAN); diffusion models are more complex and efficient, the use of streaming models and autoencoders is also known [3].

Deep learning methods analyze important features of training images to reconstruct image details [2]. These include convolutional neural networks (CNN), recurrent neural networks (RNN), and residual neural networks (ResNet) [3]. It is important to note that they are often part of architecture of generative models that implement a particular learning principle. For example, the generator and discriminator in a GAN are deep neural networks.

SISR models are trained by learning pairs of low- and high-resolution images from the training selection. The effectiveness of super-resolution image gen-

eration is assessed based on a set of indicators, which must include both quantitative and perceptual metrics. An important step in evaluating the results of SISR is the visual analysis of the generated images by a human.

It should be noted that SISR algorithms are complex and time-consuming, so they require powerful computing resources, and model optimization is still the main focus of researchers' work on this topic. That is why, when choosing the optimal model, technical indicators are added to the evaluation criteria, including time of training, training cost, and the availability of a hardware accelerator in the form of a graphics processing unit (GPU) [4].

**PROBLEM STATEMENT**

Let us introduce the notation $H$ for height, $W$ for width, and $C$ for the number of image channels (e.g. RGB). Let $I_{LR} \in R^{H\times W\times C}$ be a low-resolution image, and $I_{HR} \in R^{H\times W\times C}$ be its corresponding high-resolution image. The goal of the SISR problem is to find the following mapping

$$f : I_{LR} \to I_{HR}, \tag{1}$$

that will ensure the most accurate recovery of the details of the $I_{HR}$ image based on the information from the $I_{LR}$.

Mapping (1) is a formalization tool, as it can describe different processes depending on the resolution enhancement method. That is why we will further consider the implementation of (1), the model $f_\theta \in F$, where $\theta$ are the model parameters, $F$ is the set of all SISR models. The target super-resolution image is the output of $f_\theta$ and the result of solving the problem:

$$I_{SR} = f_\theta(I_{LR}).$$

An important step in the process of training models from $F$ is to solve the optimization problem

$$\min_\theta L(I_{HR}, I_{SR}),$$

where $L(I_{HR}, I_{SR})$ is the model loss function. The objective is to find such model parameters $\theta$ that the value of the loss function $L$ is minimal.

In this paper, the task of multicriteria quality assessments of images generated (formed) by different models and deep learning methods is set. Let $A = \{a_i \mid i = 1,2,\ldots,n\}$ be a set of super-resolution images $I_{SR}$, generated by different deep learning models based on a single low-resolution image $I_{LR}$; $C = \{c_j \mid j = 1,2,\ldots,m\}$ be a set of quality criteria for the generated images and technical characteristics of model training. In the following, $a_i$ will be considered as alternatives, and $c_j$ as decision criteria.

The task is to find the aggregated or global weights

$$W^{aggr} = \{w_i^{aggr} \mid i = 1,2,\ldots,n\} \tag{2}$$

of alternative generated (formed) images according to a set of criteria from $C$ and selection of the best generated image.

The quality criteria for the generated images are:

- traditional quantitative metrics PSNR [5], SSIM [6], MSSIM [6] (1st group of criteria);
- perceptual indicators BRISQUE [7], NIQE [8], PIQUE [9], LPIPS [10] and their modifications (e.g., LR-PSNR) (2nd group).

The decision criteria also include technical characteristics (3rd group):

- training time and cost;
- availability of a hardware accelerator in the form of a graphics processing unit (GPU).

The purpose of the studied generative models and deep learning methods is to increase the resolution of images, scale them by 4, 8, or more times, and generate realistic and beautiful images based on a given low-resolution image for further display of the generated images on large screens and human perception. Therefore, another group of criteria (4th group) ensures that the generated image is evaluated directly by a human: effects of smoothing, blurring, edge lightening, and photorealism of the image.

The coefficients of relative importance of decision criteria are determined by decision support methods [11–13] using expert pairwise comparison judgements depending on the application. The interdependence between individual decision criteria and the need to take into account fuzzy judgements provided by an expert require the use of hybrid methods [14; 15].

## MATERIALS AND METHODS

### Deep learning models for generating super-resolution images

The following models were used in the study, representing generative and deep learning methods.

1. **SRGAN (Super-Resolution Generative Adversarial Network)** is a generative adversarial network for increasing the resolution, where the generator creates super-resolution images, and the discriminator is trained to recognize real and generated images. The generator is optimized using a combination of loss functions: adversarial loss for plausibility and content loss for pixel accuracy. Full implementations also use a perceptual loss function to improve textures [16].

2. **VDSR (Very Deep Super Resolution)** is a very deep convolutional neural network for resolution enhancement tasks [17]. Its main advantage is usage of residual connections, which allow the model to learn from the difference between the input low-resolution image and the corresponding super-resolution image. This reduces the risk of gradient vanishing during training, accelerates convergence and increases training stability. Due to a large number of convolutional layers, VDSR effectively captures both fine textures and complex structures of objects in the image, which ensures high-quality results.

3. **DRCN (Deeply-Recursive Convolutional Network)** uses the concept of recursive blocks, where the same set of parameters is applied repeatedly. This allows for significant depth without increasing the number of model parameters, which reduces its computational complexity and memory requirements. As a result, DRCN effectively recovers the details of a high-resolution image while maintaining resource efficiency. The network also uses methods of averaging the

output results, supervised skip connections, which increase the stability and accuracy of recovery of details [18].

4. **SRCNN (Super-Resolution Convolutional Neural Network)** is a convolutional neural network for resolution enhancement that performs the following three sequential operations: interpolation of the input image to high resolution, feature extraction using convolutional layers, and reconstruction of the super-resolution image [19]. The model is simple and efficient, but limited in depth and ability to reconstruct complex textures. In this study, it is used as a discriminator in our implementation of SRGAN, as well as a separate pre-trained model in the framework of retraining experiments.

Two types of blocks were also used in the networks:

1) a **residual block** to maintain the stability of the gradients;

2) a **recursive block** that repeats convolutional layers with the same weights multiple times to enhance the selected features and create a more complex architecture.

The architecture of the implemented models [20] is shown in Table 1, and the architecture of their component blocks is further explained in Table 2.

**Table 1.** Architecture of the implemented models in-house

| Model | | Architecture |
|---|---|---|
| SRGAN | Generator SRResNet | Consists of an initial 9×9 convolutional layer, 5 residual blocks (ResidualBlock), an intermediate 3×3 convolutional block, a resolution upscaling block (2 3×3 convolutional layers with PixelShuffle), and a final 9×9 convolutional layer |
| | Discriminator SRCNN | Consists of 8 3×3 convolutional layers with increasing number of channels with normalization (BatchNorm2d) and LeakyReLU activation, 1 adaptive averaging layer and 2 final fully connected layers. The filter size for all convolutional layers is 3×3 |
| VDSR | | Consists of an initial convolutional layer, 18 convolutional layers with ReLU activation, and an output layer that adds the residual to the input image. The filter size for all convolutional layers is 3×3 |
| DRCN | | Consists of an input convolutional layer, a recursive block (RecursiveBlock) that is repeated a specified number of times (16), and an output convolutional layer. The filter size for all convolutional layers is 3×3 |

**Table 2.** Architecture of the model components

| Model | Architecture |
|---|---|
| ResidualBlock | Contains 2 3×3 convolutional layers, a normalization layer (BatchNorm2d) after each convolutional layer, and a PReLU activation function after the 1st layer |
| RecursiveBlock | Contains 1 3×3 convolutional layer with ReLU activation |

### Algorithm for training and evaluation models from scratch

The following algorithm for training SRGAN, VDSR, and DRCN models for generating super-resolution images and evaluation of these models in terms of quantitative and perceptual indicators is suggested:

1. Splitting the set into training and validation samples. In the case of using the DIV2K set [1], this stage is skipped, since the images are already distributed in the set.

2. Initialization of model weights using the methods of Kaiming He [21] or Xavier Glaurot [22], depending on the characteristics of the model to be trained.

3. Training on a given number of epochs (200 for the generating model with a batch size of 16; and 100 epochs for deep learning methods with a batch size of 32) on the training set with tracking the values of the loss function (adversarial loss (MSE+BCE) for the generating model, MSE for deep learning methods).

4. Saving model weights in case of training interruption or early stopping.

5. Calculating the training time of models.

6. Evaluation of the results on the test sample: calculation of the quantitative indicators PSNR, SSIM, MSSIM and the perceptual indicator LPIPS of the generated images. The pre-trained VGG network19 is used to calculate the LPIPS metric. The average value of the indicators for each model is presented for 10 random images.

**Algorithm for training models using pre-training technology**

An algorithm for training of pre-trained models for the formation of super-resolution images is suggested, which consists of the following steps:

1. Careful selection of a pre-trained model, which must be aimed at the same task and preferably trained on a large universal data set.

2. Loading the weights for the selected model, with the values of which training will continue.

3. Determine the number of epochs for which the model should be retrained.

4. Fine-tuning the model: freezing layers (usually the initial ones) and adding new ones which extract high-level features (residual blocks, convolutions with small kernels, normalization layers, Upsampling or PixelShuffle), using a low learning rate to ensure its stability, combining the main loss with the perceptual loss to focus on the visual quality of the generated images.

5. Applying early stopping in case of signs of model overfitting according to metrics PSNR, SSIM, MSSIM and a perceptual metric LPIPS.

The experiment on retraining of pre-trained models was conducted on 20 epochs. The purposes of the experiment are: to improve the result of image generation, as well as to check whether it is possible to obtain a result better than that of other researchers [23], and whether overfitting is occur.

**Quantitative and perceptual metrics and indicators**

The quality of SISR models is traditionally evaluated based on metrics and indicators that compare the SR image generated by the model with the original HR image from a labeled test image set [24].

The classical **PSNR** (Peak Signal-to-Noise Ratio) metric has limitations for evaluating structured data such as images, as it assumes pixel independence. PSNR measures the difference between pixels of a pair of images as a ratio between the maximum possible signal strength and noise. For example, blurring an image can cause a large perceptual change and at the same time a small change in the $L_2$ measure. SSIM [6] index assesses structural similarity of two images.

The perceptual distance estimates the similarity of high-level features of two images similar to human visual perception. Perceptual indicators such as BRISQUE [7], NIQE [8], PIQUE [9], LPIPS [10], and others have been suggested. Let us describe some of them in more detail.

**SSIM** (Structural Similarity Index Measure) evaluates the similarity of two images $x$ and $y$ based on three image components: brightness, contrast, and structure [6]:

$$SSIM(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma},$$

where $\alpha, \beta, \gamma > 0$ are the coefficients of relative importance of the three components, are the parameters.

The SSIM satisfies the symmetry properties $SSIM(x,y) = SSIM(y,x)$; boundedness $SSIM(x,y) \le 1$; and unique maximum: $SSIM(x,y) = 1$ if and only if $x = y$.

Later, the authors of [6] move on to a following simplified expression:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \qquad (3)$$

where $\mu_x$ is the average image intensity value $x$; $\sigma_x$ is the standard deviation for image $x$, which serves as an unbiased estimate of its contrast; $\sigma_{xy}$ is the covariance between two images $x$ and $y$, which is the basis for comparing image structures after subtracting brightness and normalizing variance, and also use the following modified estimates of local statistics $\mu_x$, $\sigma_x$ та $\sigma_{xy}$:

$$\mu_x = \sum_{i=1}^{N} v_i x_i, \quad \sigma_x = \left( \sum_{i=1}^{N} v_i (x_i - \mu_x)^2 \right)^{1/2};$$

$$\sigma_{xy} = \sum_{i=1}^{N} v_i (x_i - \mu_x)(y_i - \mu_y)$$

with a circularly symmetric normalized Gaussian weight function $v = \{v_i \mid i = 1,2,\ldots,N\}$ with a standard deviation of 1.5 samples, $\sum_{i=1}^{N} v_i = 1$, and a sliding window approach that ensures the property of local isotropy of the quality maps.

The constants $C_1$ і $C_2$ are included in (3) to avoid instability when the expressions $\mu_x^2 + \mu_y^2$ і $\sigma_x^2 + \sigma_y^2$ are practically zero. $C_1 = (K_1 L)^2$ and $C_2 = (K_2 L)^2$ are defined, where $L$ is the dynamic range of pixel values, e.g., $L = 255$ for 8-bit grayscale images, and $K_1 << 1$ and $K_2 << 1$ are small constants, for example, $K_1 = 0.01$, $K_2 = 0.03$ [6].

In practice, in cases where a single overall measure of quality of the entire image is required, the average value of SSIM indices (3) over a set of image pixels called MSSIM is suggested, which aggregates the structural similarity between the reference and distorted images. MSSIM is calculated as the arithmetic mean of $SSIM(x_j, y_j)$ over the image content in the $j$-th local window [6].

In this paper, a weighted average of different samples in the SSIM index map is proposed:

$$WM\_SSIM(X,Y) = \sum_{j=1}^{M} w_j \, SSIM(x_j, y_j),$$

where $M$ is the number of local windows in the image, $x_j$ and $y_j$ are the content of the reference $X$ and distorted $Y$ images at the $j$-th local window, and $w_j$ are weighting coefficients for different samples (e.g. different image textures attract a person's attention with varying degrees). Weights $w_j$ are calculated depending on the practical problem by analyzing decision hierarchies or networks with the consideration of human assessments [11; 12; 14].

**LPIPS** (Learned Perceptual Image Patch Similarity) is a perceptual metric that aimed at evaluating the visual perception of an image by a person at the level of details and uses deep neural networks to assess the visual similarity of a pair of features based on extracted features [10]:

$$LPIPS(I_{SR}, I_{HR}) = \sum_l w_l \cdot \frac{\left\| \phi_l(I_{SR}) - \phi_l(I_{HR}) \right\|_2^2}{H_l \cdot W_l \cdot C_l},$$

where $\phi_l(I_{SR})$ is an activation of VGG or another deep network on the $l$-th layer for the image $I_{SR}$; $H_l$, $W_l$, $C_l$ are the height, width and number of channels of the $l$-th feature map; $w_l$ is a weighting factor that adjusts the contribution of different layers.

An explanation of the values for each indicator is provided in Table 3. Through visual analysis of a large number of generated images and the corresponding values of quality indicators, thresholds for practically acceptable and high-quality results were obtained, which are given in the last two columns of Table 3.

**T a b l e 3 .** Indicator analysis criteria for the SISR task [20]

| Indicator | Value range | Practically acceptable result | High-quality result |
|---|---|---|---|
| PSNR↑ | [0; 1] | >20 | >30 |
| MSSIM↑ | [0; 1] | >0.7 | >0.9 |
| LPIPS↓ | [0; 1] | <0.3 | <0.1 |

For an objective evaluation of the models, it is necessary to add the training time of the models to the indicator analysis. Attention should also be paid to the fact that the indicator values are not worse than the bicubic increase (scaling LR to HR), as this will indicate extremely poor quality of the models even if practically acceptable values are obtained.

## Algorithm for assessing the quality of models and deep learning methods in terms of multiple quantitative and qualitative criteria

Generative models and deep learning methods, which are studied, are aimed at increasing the resolution of images, scale them by 4 or more times, and as a result generate realistic and beautiful images for further human perception. Therefore, it is necessary to add another group of qualitative decision criteria, including effects of smoothing, blurring, edge lightening, and photorealism of the image. In terms of these criteria, we evaluate the set of images (decision alternatives) generated by different generative models and deep learning methods. Evaluation is made directly by a human using one of the paiwise comparison methods [11–15]. The decision support (DS) problem of multiple criteria evaluation of decision alterna-

tives can be solved using a systematic approach and methodology based on hierarchical and network models [25]. On their basis, an algorithm to solve the problem is suggested, which has the following five stages:

1. Determine interdependencies among decision criteria and decision alternatives. A hierarchy or DS network is formed, which includes the overall goal — selection of the best generated image, qualitative decision criteria: effects of smoothing, blurring, edge lightening, and photorealism of the image, and decision alternatives: image_SRGAN, image_VDSR and image_DRCN (Fig. 1).

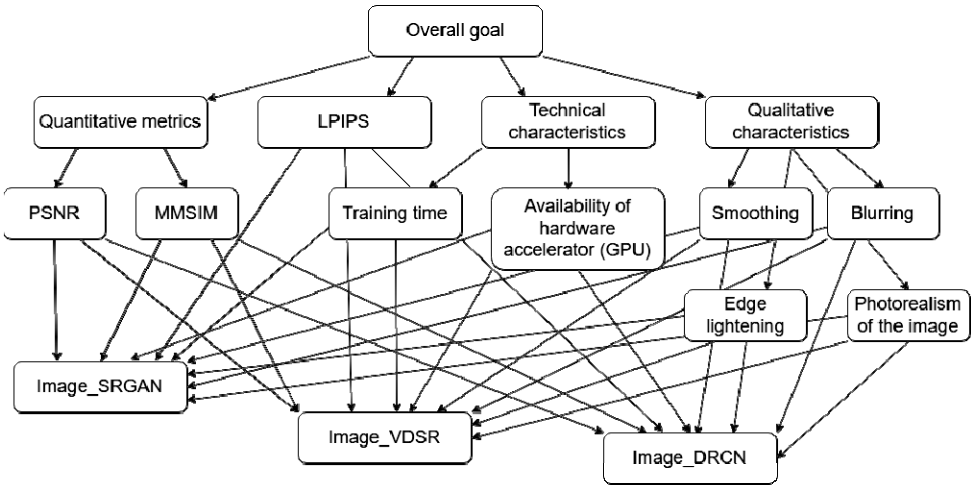



*Fig. 1.* An hierarchy for assessing the quality of images generated by different models

2. The importance of the decision criteria in relation to the main goal is assessed by experts using the pairwise comparison method on a special scale. Based on the results of the assessment, pairwise comparison matrices (PCMs) are constructed, and the quality of expert opinions is analyzed and, if necessary, improved using the method of evaluation and consistency improvement. The most inconsistent expert opinion is founded. As a result, for all elements of the hierarchy or the DS network, we obtain a set of PCMs of acceptable quality.

3. The coefficients of relative importance (local weights) of the elements of the hierarchy or the DS network are calculated based on the PCMs.

4. The local weights are aggregated using different methods depending on whether the decision criteria are independent (hierarchy case), interdependent (hierarchy case with a loop at the criterion level), or whether there are feedbacks from alternatives to decision criteria (DS network case).

5. The sensitivity analysis of aggregated results (2) is performed.

The purposes of the algorithm are: to calculate local weights for decision alternatives (image_SRGAN, image_VDSR, and image_DRCN) in terms of each decision criteria, as well as to calculate aggregated weights and perform their sensitivity analysis.

## RESULTS OF THE EXPERIMENTS

### Dataset

The DIV2K dataset [1] was introduced as part of the NTIRE 2017 Challenge on Single Image Super-Resolution, held during the CVPR Workshops 2017 conference. It was created to enhance the effectiveness of solving the SISR problem by

addressing the limitations of existing datasets, namely insufficient scene diversity and the limited number of images.

DIV2K consists of a labeled set of 1000 pairs of low-resolution (LR) and high-resolution (HR) color images. The dataset is divided into three subsets: 800 samples for training, 100 samples for testing, and 100 samples for validation. Historically, the test set was designed for contestants to evaluate their models after training, while the validation set was reserved for organizers to determine the winners. The validation set initially included only LR images, and participants were required to generate their super-resolution (SR) counterparts. Once the HR versions of the validation set were made publicly available, both the test and validation sets could be utilized to assess model performance (Fig. 2).

DIV2K 100 validatiiion images DIV2K 100 test images

*Fig. 2.* Example of images for model evaluation from the DIV2K set [1]

The low-resolution (LR) images in the DIV2K dataset are derived from the original high-resolution (HR) images using either bicubic downscaling or more advanced methods that simulate real-world degradations. These methods include modeling blurring caused by motion, introducing fractional noise, and applying distortions due to uneven pixel mapping, among others.

The dataset includes images reduced by scaling factors of 2 (×2), 3 (×3), and 4 (×4). Greater downscaling significantly diminishes image quality (Fig. 3) while also reducing the time required for model training. The classical approach to Single Image Super-Resolution (SISR) typically employs LR images generated through a 4-fold reduction of the original HR images using bicubic interpolation.


*Fig. 3.* Demonstration of image quality deterioration with a 2 and 4 times reduction in resolution

After its introduction in 2017, the DIV2K dataset has been extensively used to evaluate various super-resolution (SR) models, including in studies conducted in 2019 [26], 2020 [23], and 2023 [27].

**Training process and results**

In the first experiment (Section 3.2), we trained our own implementations of the SRGAN, VDSR, and DRCN models from scratch using the DIV2K dataset. The optimization processes of their respective loss functions during training are illustrated in Figs. 4 and 5, while the metric values obtained are presented in Table 4.

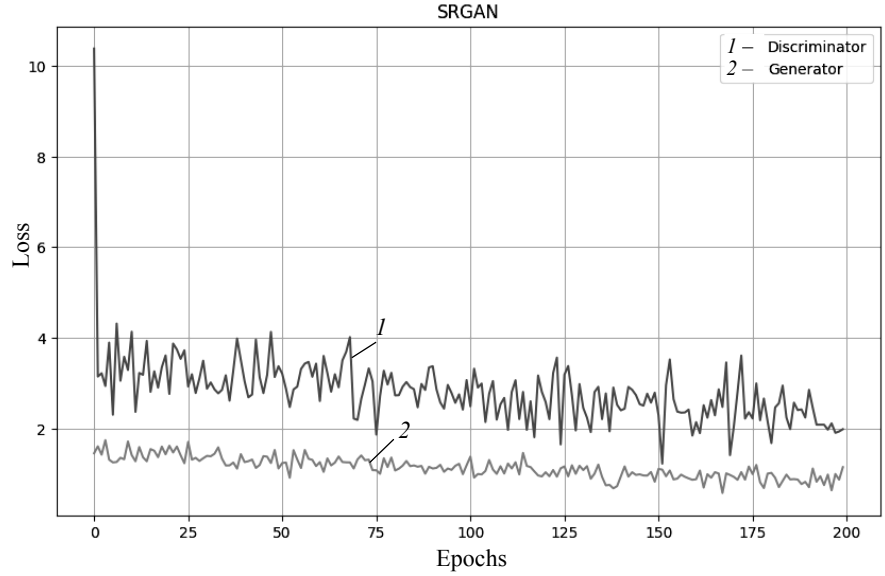



*Fig. 4.* The process of optimising the loss functions of the generator and discriminator of the SRGAN model [20]

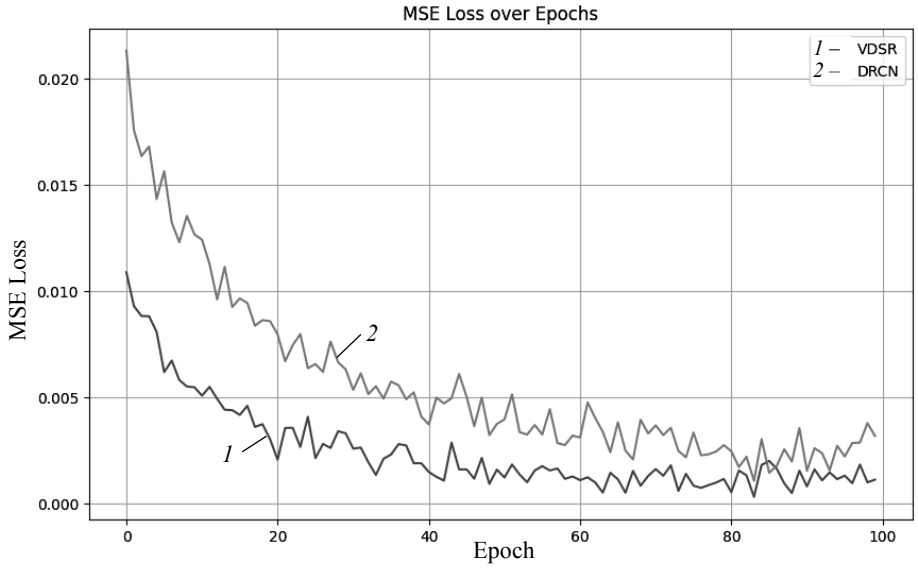



*Fig. 5.* The process of optimising the loss functions of VDSR and DCRN networks

The second experiment (Section 3.3) involved retraining the previously trained SRGAN, VDSR, and DRCN models. The results of this retraining process are provided in Table 5, and the evolution of perceptual quality, as measured by the LPIPS metric, is shown in Fig. 6. For the pre-trained models, we used implementations of SRGAN [16; 28], VDSR [17; 29], and SRCNN [19; 30].

**Table 4.** Values of quality indicators of the generated super-resolution images for our own model implementations at 4-fold image magnification [20]

| Model | Indicator | | | Training time (h) |
|---|---|---|---|---|
| | PSNR↑ | MSSIM↑ | LPIPS↓ | |
| Bicubic | 25.80 | 0.74 | 0.46 | – |
| SRGAN | 24.50 | 0.71 | 0.33 | 32 |
| VDSR | 26.73 | 0.77 | 0.31 | 16 |
| DRCN | 26.41 | 0.76 | 0.37 | 25 |

**Table 5.** Values of quality indicators of images enlarged by 4 times as a result of retraining of pre-trained models

| Model | Indicator | | | Training time (min) |
|---|---|---|---|---|
| | PSNR↑ | MSSIM↑ | LPIPS↓ | |
| Bicubic | 25.80 | 0.74 | 0.46 | – |
| EDSR [31] | 28.98 | 0.83 | 0.270 | – |
| RRDB [32] | 29.44 | 0.84 | 0.253 | – |
| ESRGAN [32] | 26.22 | 0.75 | 0.124 | – |
| pre-trained SRGAN | 26.9 | 0.79 | 0.16 | 27 |
| pre-trained VDSR | 28.9 | 0.84 | 0.1 | 11 |
| pre-trained SRCNN | 27.5 | 0.81 | 0.12 | 2 |

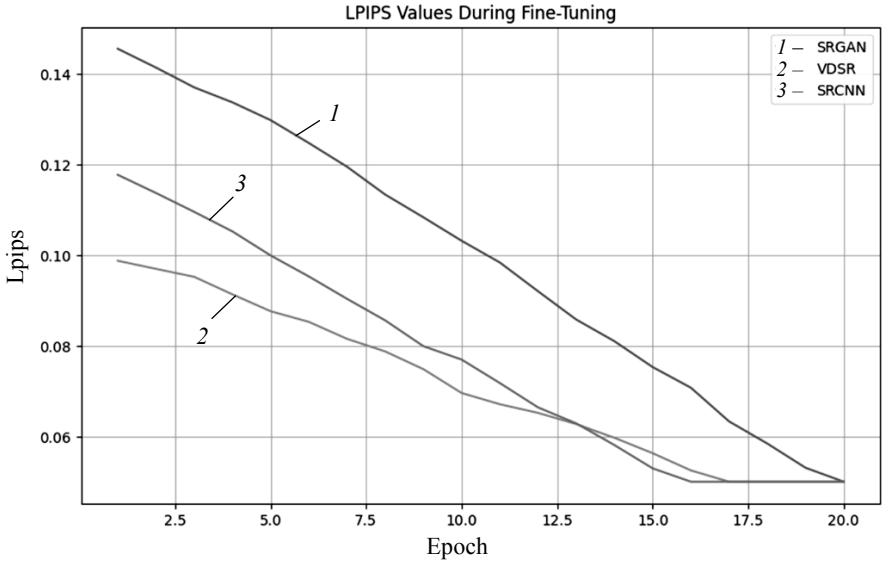



*Fig. 6.* Change in the perceptual quality of LPIPS images enlarged by a factor of 4 when retraining pre-trained SRGAN, VDSR and SRCNN models

The software solutions for these experiments were developed in the Jupyter Notebook environment using Python, along with the PyTorch library for model development and the matplotlib library for visualization. The models were trained on a PC equipped with an Nvidia GeForce RTX 4060 GPU accelerator.

## ANALYSIS OF THE RESULTS AND DISCUSSION

The results of the first experiment (Section 3.2, Figs. 4, 5, Table 4) demonstrate practically acceptable outcomes for all considered models, with VDSR perform-

ing the best. This highlights, in particular, that residual learning proved to be more effective than recursive learning. The SRGAN architecture, in this experiment, was too simplistic for the given task, as generating new details often outperforms feature refinement.

A comparison of the results in Table 4 with those obtained by other researchers [23] indicates that the metrics in Table 4 are worse than those reported for other SISR models [23]. However, the visual comparison of the generated super-resolution (SR) images with their low-resolution (LR) and high-resolution (HR) counterparts (Fig. 7) shows satisfactory results, provided that the models were trained using the algorithm proposed in Section 3.2.


*Fig. 7.* Visual comparison of the generated SR images with the high-resolution (HR) original and low-resolution (LR) input image for the proprietary implementation of the VDSR model

The results of the second experiment (Section 3.3, Table 5), which employed pre-training techniques, are comparable to those achieved by other researchers [23]. Specifically, the VDSR model, implemented and fine-tuned using the algorithm proposed in this study, achieved an MSSIM value of 0.84, which is on par with the RRDB model [32] and surpasses the MSSIM values of other models developed and fine-tuned in this study: SRGAN (MSSIM = 0.79), SRCNN (MSSIM = 0.81), as well as EDSR [31] and ESRGAN [32].

In terms of the perceptual quality metric LPIPS, the VDSR model trained with the proposed algorithm outperformed other SRGAN and SRCNN models implemented in this study, as well as the EDSR [31], RRDB [32], and ESRGAN [32] models.

The second experiment (Section 3.3) revealed no signs of overfitting, and the generated SR images demonstrated high quality compared to the input LR-HR pairs (Fig. 8). The VDSR model consistently produced the best visual results, underscoring the advantage of feature enhancement when addressing SISR tasks for highly detailed data and complex real-world scenes.

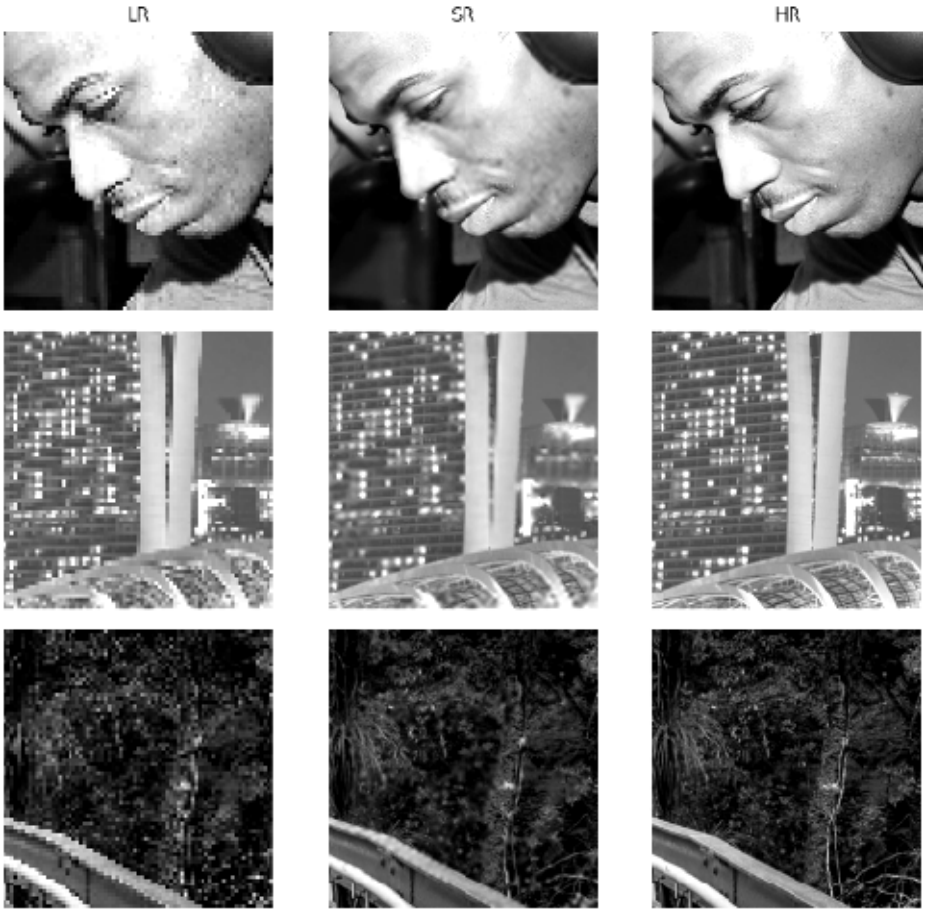



*Fig. 8.* Visual comparison of the generated SR images with the high-resolution (HR) original and low-resolution (LR) input image for the VDSR model trained with the suggested algorithm

## CONCLUSIONS

This study presents an algorithm for the comprehensive evaluation of image super-resolution results based on quantitative metrics, perceptual indicators, technical characteristics, and aspects of human image perception. Threshold criteria for practically acceptable and high-quality results were determined through visual analysis of many generated images and their corresponding quality metrics, including those obtained by other researchers.

The VDSR model was identified as the optimal one (among those considered) in terms of pixel, structural, and perceptual metrics, as well as training time. The absence of overfitting and the quality of super-resolution images generated by VDSR were visually confirmed on selected test set samples depicting various

shapes, textures, and color combinations. Overall, deep learning methods demonstrated superiority over generative models in the conducted experiments based on the results of the comprehensive evaluation.

## REFERENCES


1. E. Agustsson, R. Timofte, "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017*. doi: https://doi.org/10.1109/cvprw.2017.150
2. Z. Wang, J. Chen, S.C.H. Hoi, "Deep Learning for Image Super-resolution: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3365–3387, 2020. doi: https://doi.org/10.1109/tpami.2020.2982166
3. R. Timofte et al., "NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017*. doi: https://doi.org/10.1109/cvprw.2017.149
4. T. Ausare, "Ultimate Guide to Selecting a GPU for Deep Learning. Latest AI, ML & GPU Updates," *NeevCloud*. Available: https://blog.neevcloud.com/ultimate-guide-to-selecting-a-gpu-for-deep-learning
5. F.A. Fardo, V.H. Conforto, F.C. de Oliveira, P.S. Rodrigues, *A Formal Evaluation of PSNR as Quality Measurement Parameter for Image Segmentation Algorithms*. 2016. doi: https://doi.org/10.48550/arXiv.1605.07116
6. Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, Eero P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, issue 4, pp. 600–612, 2004. doi: https://doi.org/10.1109/TIP.2003.819861
7. A. Mittal, A. Moorthy, A. Bovik, "Referenceless image spatial quality evaluation engine," in *45th Asilomar Conference on Signals, Systems and Computers*, vol. 38, pp. 53–54, 2011. doi: https://doi.org/10.1109/ACSSC.2011.6190099
8. A. Mittal, R. Soundararajan, A.C. Bovik, "Making a "completely blind" image quality analyser," *IEEE Signal Process. Lett.*, vol. 20, issue 3, pp. 209–212, 2013. doi: https://doi.org/10.1109/LSP.2012.2227726
9. N. Venkatanath, D. Praneeth, Bh. Maruthi Chandrasekhar, S.S. Channappayya, S.S. Medasani, "Blind image quality evaluation using perception based features," *2015 Twenty First National Conference on Communications (NCC), Mumbai, India, 2015*, pp. 1–6. doi: https://doi.org/10.1109/NCC.2015.7084843
10. R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018*, pp. 586–595. doi: https://doi.org/10.1109/CVPR.2018.00068
11. N.I. Nedashkovskaya, "Method for weights calculation based on interval multiplicative pairwise comparison matrix in decision-making models," *Radio Electronics, Computer Science, Control*, no. 3, pp. 155–167, 2022. doi: https://doi.org/10.15588/1607-3274-2022-3-15
12. N.I. Nedashkovskaya, "Estimation of the accuracy of methods for calculating interval weight vectors based on interval multiplicative preference relations," *IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC)*, 2022. doi: https://doi.org/10.1109/SAIC57818.2022.9922977
13. N.I. Nedashkovskaya, "Method for Evaluation of the Uncertainty of the Paired Comparisons Expert Judgements when Calculating the Decision Alternatives Weights," *Journal of Automation and Information Sciences*, vol. 47, issue 10, pp. 69–82, 2015. doi: https://doi.org/10.1615/JAutomatInfScien.v47.i10.70

14. N.D. Pankratova, N.I. Nedashkovskaya, “Hybrid Method of Multicriteria Evaluation of Decision Alternatives,” *Cybernetics and Systems Analysis*, vol. 50, no. 5, pp. 701–711, 2014. doi: https://doi.org/10.1007/s10559-014-9660-2
15. N.I. Nedashkovskaya, “Investigation of methods for improving consistency of a pairwise comparison matrix,” *Journal of the Operational Research Society*, vol. 69, no. 12, pp. 1947–1956, 2018. doi: https://doi.org/10.1080/01605682.2017.1415640
16. C. Ledig et al., “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 21–26 July 2017*, pp. 105–114. doi: https://doi.org/10.1109/cvpr.2017.19
17. J. Kim, J.K. Lee, K.M. Lee, “Accurate Image Super-Resolution Using Very Deep Convolutional Networks,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016*, pp. 1646–1654. doi: https://doi.org/10.1109/cvpr.2016.182
18. J. Kim, J.K. Lee, K.M. Lee, “Deeply-Recursive Convolutional Network for Image Super-Resolution,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016*, pp. 1637–1645, 2016. doi: https://doi.org/10.1109/cvpr.2016.181
19. C. Dong et al., “Image Super-Resolution Using Deep Convolutional Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016. doi: https://doi.org/10.1109/tpami.2015.243928
20. A.A. Lanko, N.I. Nedashkovskaya, “Generative models and methods of deep learning for the SISR problem,” *System sciences and informatics: collection of reports of the 3rd All-Ukrainian scientific and practical conference “System sciences and informatics”, November 25–29, 2024, Kyiv*. K.: IASA KPI, 2024, pp. 176–181. Available: http://mmsa.kpi.ua/sites/default/files/systemni_nauky_ta_informatyka_2024.pdf
21. K. He et al., “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” *2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015*, pp. 1026–1034. doi: https://doi.org/10.1109/iccv.2015.123
22. X. Glorot, Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), Sardinia, Italy, 13–15 May 2010*, PMLR, vol. 9, pp. 249–256. Available: http://proceedings.mlr.press/v9/glorot10a.html
23. A. Lugmayr et al., “SRFlow: Learning the Super-Resolution Space with Normalizing Flow,” *Computer Vision – ECCV 2020, Cham, 2020*, pp. 715–732. doi: https://doi.org/10.1007/978-3-030-58558-7_42
24. Q. Jiang et al., “Single Image Super-Resolution Quality Assessment: A Real-World Dataset, Subjective Studies, and an Objective Metric,” *IEEE Transactions on Image Processing*, vol. 31, pp. 2279–2294, 2022. doi: https://doi.org/10.1109/tip.2022.3154588
25. N.I. Nedashkovskaya, “A system approach to decision support on basis of hierarchical and network models,” *System Research and Information Technologies*, no. 1, pp. 7–18, 2018. doi: https://doi.org/10.20535/srit.2308-8893.2018.1.01
26. A. Ignatov et al., “PIRM challenge on perceptual image enhancement on smartphones: report,” *Conference on Computer Vision (ECCV) Workshops*, 2019. doi: https://doi.org/10.1007/978-3-030-11021-5_20
27. Dandan Gao, Dengwen Zhou, “A very lightweight and efficient image super-resolution network,” *Expert Systems with Applications*, vol. 213, Part A, 1, March 2023, 118898. doi: https://doi.org/10.1016/j.eswa.2022.118898
28. “GitHub - tensorlayer/SRGAN: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” *GitHub*. Available: https://github.com/tensorlayer/SRGAN
29. “GitHub - twtygqyy/pytorch-vdsr: VDSR (CVPR2016) pytorch implementation,” *GitHub*. Available: https://github.com/twtygqyy/pytorch-vdsr.

30. “GitHub - Lornatang/SRCNN-PyTorch: Pytorch framework can easily implement srcnn algorithm with excellent performance,” *GitHub*. Available: https://github.com/Lornatang/SRCNN-PyTorch
31. B. Lim, S. Son, H. Kim, S. Nah, K.M. Lee, “Enhanced deep residual networks for single image super-resolution,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017*, pp. 1132–1140. doi: https://doi.org/10.1109/CVPRW.2017.151
32. X. Wang et al., “ESRGAN: Enhanced super-resolution generative adversarial networks,” *Computer Vision – ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part V*, pp. 63–79. doi: https://doi.org/10.1007/978-3-030-11021-5_5

*Received 27.12.2024*

## INFORMATION ON THE ARTICLE

**Anna A. Lanko,** ORCID: 0009-0005-8370-5739, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine, e mail: lanko.anna@lll.kpi.ua

**Nadezhda I. Nedashkovskaya,** ORCID: 0000-0002-8277-3095, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine, e mail: nedashkovskaya.nadezhda@ lll.kpi.ua

**ОЦІНЮВАННЯ ЯКОСТІ МОДЕЛЕЙ ТА МЕТОДІВ ГЛИБОКОГО НАВЧАННЯ ДЛЯ ФОРМУВАННЯ СУПЕРРОЗДІЛЬНИХ ЗОБРАЖЕНЬ** / Н.І. Недашківська, А.А. Ланько

**Анотація.** Розглянуто метрику оцінювання результатів генерації суперрозділь-них зображень під час розв’язання задачі SISR. Дослідження включає два експерименти: власну реалізацію мережевих архітектур для SRGAN, VDSR і SRCNN, і точне налаштування попередньо навчених моделей SRGAN, VDSR і SRCNN. Запропоновано алгоритм оцінювання якості моделей і методів глибокого навчання для генерації суперроздільних зображень. Модель VDSR продемонструвала найкращі результати з точки зору піксельного, структурних і перцептивних показників, а також часу навчання та візуального підтвердження якості згенерованого зображення людиною, підкреслюючи, що залишкове навчання є більш ефективним, ніж рекурсивне навчання за умов двох проведених експериментів. Порогові значення для прийнятних і високоякісних результатів визначено шляхом візуального аналізу багатьох згенерованих зображень і відповідних показників якості, включно з тими, про які повідомляли інші дослідники.

**Ключові слова:** задача SISR, оцінювання якості, генеративні моделі, методи глибокого навчання, згорткова нейронна мережа, залишкове навчання, рекурсивне навчання, тонке налаштування попередньо навчених моделей, перцептивна метрика, LPIPS, багатокритеріальний аналіз розв’язань, набір даних DIV2K, порогові значення для прийнятних і високоякісних згенерованих зображень.

UDC 543.33 + 004.032.26
DOI: 10.20535/SRIT.2308-8893.2025.4.07

# PREDICTION OF MECHANISMS OF TOXIC ACTION OF PHENOLS BY MEANS OF PROBABILISTIC NEURAL NETWORK IN COMBINATION WITH KRUSKAL–WALLIS TEST

**Ya.M. PUSHKAROVA, G.M. ZAITSEVA**

**Abstract.** Prediction of the toxicity of chemical compounds is one of the most important steps in drug design. The use of phenolic compounds is a promising component in the pharmaceutical industry with many possible applications. The paper focuses on the application of a probabilistic neural network for classifying 232 phenols based on their mechanisms of toxic action. The Kruskal–Wallis test was also used to assess the influence of molecular descriptors on the reliable classification of phenolic compounds based on the mechanisms of their toxic action. It is shown that for the correct training of a probabilistic neural network and effective prediction of the mechanisms of toxic action of phenols, it is sufficient to use only 5 molecular descriptors.

**Keywords:** artificial neural network, classification, drug design, phenol, toxicity.

## INTRODUCTION

Assessment of the toxicity of chemical compounds is an important and necessary stage on the way to the creation of new medicines. It is known that the experimental study of only one type of toxicity is an expensive and long-term process. Phenolic compounds have a number of useful properties that make them interesting for pharmacy: antioxidant, anti-inflammatory, antimicrobial properties, anticancer activities, etc. Additionally, phenolic compounds are often found in natural sources, such as plants, which adds to their appeal for use in pharmacy [1–4].

Overall, the diverse range of beneficial properties exhibited by phenolic compounds makes them valuable compounds in pharmacy and medicine, with potential applications in the treatment and prevention of various diseases. But before using phenols in pharmacy, it is important to predict possible mechanisms of their toxic action (polar narcotics, weak acid respiratory uncouplers, pro-electrophiles and soft electrophiles). This helps to identify risks to people and to take measures to reduce the possible negative consequences, that is, to develop safe medicines [5; 6].

Chemometric methods use mathematical and statistical models to analyze complex data sets and extract meaningful information, making them valuable tools in pharmaceutical research and development. Chemometric methods, in particular artificial neural networks, are widely used for prediction and classification tasks in pharmacy. Artificial neural networks are computational models inspired by the structure and functioning of biological neural networks in the human brain. These methods can help predict various properties of pharmaceutical compounds, such as their stability, toxicity, solubility and bioavailability. They are also used for identifying different types of drugs or distinguishing between counterfeit and authentic products [7–10].

## MATERIALS AND METHODS

### Data Set

The studied dataset consists of a training, testing and validation sub-sets with a total of 232 phenolic compounds: training sub-set – 197 phenols, testing sub-set – 20 phenols, validation sub-set – 15 phenols. All phenolic compounds were characterized by seven physical-chemical descriptors: 1) distribution coefficient; 2) energy of the lowest unoccupied molecular orbital; 3) molecular weight; 4) negatively charged molecular surface area in percent's; 5) sum of absolute charges on nitrogen and oxygen atoms in a molecule; 6) largest positive charge on a hydrogen atom; 7) electrotopological state index for the hydroxyl group. Values of these descriptors and toxicity values were taken from [6].

Distribution of the studied phenolic compounds into classes according to the mechanisms of toxic action of phenolic compounds to Tetrahymena pyriformis is presented in Table 1. The most numerous class is class 1 of polar narcotics (71.6% of all studied phenolic compounds), other classes are almost the same in number of samples.

**Table 1.** Distribution of the studied phenolic compounds into classes according to the mechanisms of toxic action to Tetrahymena pyriformis

| Classes According to Mechanisms of Toxic Action | Number of Phenolic Compounds | | | |
|---|---|---|---|---|
| | Training sub-set | Testing sub-set | Validation sub-set | Total |
| Class 1. Polar narcotics | 138 | 16 | 12 | 166 |
| Class 2. Weak acid respiratory uncouplers | 15 | 1 | 1 | 17 |
| Class 3. Pro-electrophiles | 22 | 2 | 0 | 24 |
| Class 4. Soft electrophiles | 22 | 1 | 2 | 25 |

### Applied Methods

The software package Matlab R2023b (trial individual license 11937601) was used in the present work for realization Kruskal–Wallis test and probabilistic neural network [11].

The Kruskal–Wallis test is a non-parametric statistical test used to determine whether there are statistically significant differences between two or more groups of a dependent variable [12].

A probabilistic neural network is a type of artificial neural network, which consists of following layers: input layer, pattern layer, summation layer, and output layer. A brief overview of how probabilistic neural network works [13–15]:

- input layer receives the input pattern;
- neurons of pattern layer store the training patterns;
- summation layer computes the similarity between the input pattern and the stored patterns using Gaussian function;
- output layer produces the class probability estimates.

To classify a new input pattern, the probabilistic neural network computes the class probabilities using the summation layer and outputs the class with the highest probability.

## RESULTS AND DISCUSSION

### Definition of Informative Descriptors for Classification of Phenolic Compounds into Classes According to the Mechanisms of Toxic Action

The calculation of the Kruskal–Wallis test for 232 phenols characterized by 7 molecular descriptors and toxicity is given in Table 2.

**Table 2.** Results of the Kruskal–Wallis test calculation for 7 descriptors and toxicity

| Parameter | Toxicity | Distribution coefficient | Energy of the lowest unoccupied molecular orbita | Molecular weight | Negatively charged molecular surface area in percent's | Sum of absolute charges on nitrogen and oxygen atoms in a molecule | Largest positive charge on a hydrogen atom | Electrotopological state index for the hydroxyl group |
|---|---|---|---|---|---|---|---|---|
| $\chi^2$ | 17.80 | 54.32 | 104.90 | 35.78 | 70.24 | 31.71 | 4.34 | 18.56 |

Critical value of $\chi^2$ at the significance level of 5% with 3 degrees of freedom is 7.82 [16].

It was established some dependences between studied descriptors and classification of phenolic compounds according to the mechanisms of their toxic action:

1) descriptor largest positive charge on a hydrogen atom is not influenced on classification of phenolic compounds according to the mechanisms of toxic action, because experimental value of $\chi 2$ is less than critical value ($4.34 < 7.82$);

2) descriptor energy of the lowest unoccupied molecular orbital has the greatest influence on the phenols classification according to the mechanisms of toxic action (maximum experimental value of $\chi^2$ is established for this descriptor — 104.90);

3) the studied parameters can be conventionally divided into three groups according to their influence on the classification of phenols:

- weak influence: toxicity and electrotopological state index for the hydroxyl group;
- moderately strong influence: molecular weight and sum of absolute charges on nitrogen and oxygen atoms in a molecule;
- strong influence: distribution coefficient, energy of the lowest unoccupied molecular orbital and negatively charged molecular surface area in percent's.

### Application of Probabilistic Neural Network

In the context of the probabilistic neural network, the spread of the Gaussian function is an important parameter for its construction. Choosing the right spread parameter is crucial for the performance of the probabilistic neural network. If the spread is too small, the network may over fit to the training data and perform poorly on new data. If the spread is too large, the network may under fit and fail to capture the underlying patterns in the data [8; 13].

In the present work it was investigated the applicability of probabilistic neural network at different values of the spread of the Gaussian function: 0.1; 0.2;

0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1.0. It should be noted that the probabilistic neural network is trained with zero error at spread values from 0.1 to 1.0 for different sets of descriptors. Results of prediction of the mechanisms of toxic action of phenols for testing and validation sub-sets are also the same for spread values from 0.1 to 1.0 for different sets of descriptors.

The unreliability of the prediction was estimated as the part of incorrectly classified phenols of the testing or validation sub-sets in percent's [8]:

$$P = \frac{n}{N} \cdot 100\% ,$$

where $n$ is the number of incorrectly classified phenols in the testing or validation sub-set; $N$ is the total number of phenols in the testing or validation sub-set.

Results of prediction of the mechanisms of toxic action of phenolic compounds by means of probabilistic neural network based on a set of all 7 molecular descriptors and toxicity are shown in Table 3.

**Table 3.** Unreliability values of the prediction based on a set of all 7 molecular descriptors and toxicity

| **Sub-set** | ***P*, %** |
|---|---|
| Testing | 10.0 |
| Validation | 6.7 |

Results of prediction of the mechanisms of toxic action of phenolic compounds by means of probabilistic neural network based on a set of 5 molecular descriptors (distribution coefficient, energy of the lowest unoccupied molecular orbital, molecular weight, negatively charged molecular surface area in percent's and sum of absolute charges on nitrogen and oxygen atoms in a molecule) are shown in Table 4.

**Table 4.** Unreliability values of the prediction based on a set of 5 molecular descriptors

| **Sub-set** | ***P*, %** |
|---|---|
| Testing | 20.0 |
| Validation | 6.7 |

One can see, that results of prediction of the mechanisms of toxic action of phenolic compounds based on a set of all 7 molecular descriptors with toxicity and based on a set of 5 molecular descriptors are differed by two incorrectly classified phenols. This confirms, the verity of calculation results of the Kruskal–Wallis test: largest positive charge on a hydrogen atom, toxicity and electrotopological state index for the hydroxyl group are weakly influenced on assignment of phenols to one or another class according to mechanisms of their toxic action.

Decreasing the number of descriptors into 3 (distribution coefficient, energy of the lowest unoccupied molecular orbital and negatively charged molecular surface area in percent's) resulted in an increasing the part of incorrectly classified phenols of the testing sub-set from 20% till 40% (Table 5). It means, that molecular weight and sum of absolute charges on nitrogen and oxygen atoms in a mole-

cule are moderately strong influenced for classification of phenols according to mechanisms of their toxic action and can't be ignore.

**Table 5.** Unreliability values of the prediction based on a set of 3 molecular descriptors

| Sub-set | *P*, % |
|---|---|
| Testing | 40.0 |
| Validation | 6.7 |

Detailed information about prediction of the mechanisms of toxic action of phenolic compounds of testing and validation sub-sets are shown in Tables 6 and 7, correspondingly: 1 — polar narcotics; 2 — weak acid respiratory uncouplers; 3 — pro-electrophiles; 4 — soft electrophiles. Incorrect predictions are indicated in bold text.

**Table 6.** Results of prediction of the mechanisms of toxic action of phenols of the testing sub-set

| N | Phenol compound | Predicted mechanism of toxic action using 7 descriptors and toxicity (0.1 ≤ spread ≤ 1.0) | Predicted mechanism of toxic action using 5 descriptors (0.1 ≤ spread ≤ 1.0) | Predicted mechanism of toxic action using 3 descriptors (0.1 ≤ spread ≤ 1.0) | True mechanism of toxic action [5, 6] |
|---|---|---|---|---|---|
| 1 | 2-Fluorophenol | 1 | 1 | 1 | 1 |
| 2 | 2-Allylphenol | 1 | 1 | 1 | 1 |
| 3 | 3-Chlorophenol | 1 | 1 | 1 | 1 |
| 4 | 4,6-Dichlororesorcinol | 1 | 1 | **3** | 1 |
| 5 | 4-Benzyloxyphenol | 1 | 1 | 1 | 1 |
| 6 | 3-Iodophenol | 1 | 1 | 1 | 1 |
| 7 | 2,3-Dichlorophenol | 1 | 1 | 1 | 1 |
| 8 | 4-Phenylphenol | 1 | 1 | 1 | 1 |
| 9 | 4-Hexyloxyphenol | 1 | 1 | **3** | 1 |
| 10 | 4-Hexylresorcinol | 1 | 1 | 1 | 1 |
| 11 | 2,4,5-Trichlorophenol | 1 | 1 | 1 | 1 |
| 12 | 2,4-Diaminophenol | 3 | 3 | **1** | 3 |
| 13 | Methylhydroquinone | 3 | **1** | **1** | 3 |
| 14 | 3-Nitrophenol | 4 | 4 | **1** | 4 |
| 15 | 4-Ethoxyphenol | 1 | **3** | **3** | 1 |
| 16 | 4-Bromo-2,6-dimethylphenol | 1 | 1 | 1 | 1 |
| 17 | 4-Methoxyphenol | 1 | 1 | 1 | 1 |
| 18 | 2,6-Diiodo-4-nitrophenol | **1** | **1** | **4** | 2 |
| 19 | 2-Methyl-3-nitrophenol | **4** | **4** | **4** | 1 |
| 20 | 4-Isopropylphenol | 1 | 1 | 1 | 1 |

**Table 7.** Results of prediction of the mechanisms of toxic action of phenols of the validation sub-set

| N | Phenol compound | Predicted mechanism of toxic action using 7 descriptors and toxicity (0.1 ≤ spread ≤ 1.0) | Predicted mechanism of toxic action using 5 descriptors (0.1 ≤ spread ≤ 1.0) | Predicted mechanism of toxic action using 3 descriptors (0.1 ≤ spread ≤ 1.0) | True mechanism of toxic action [5, 6] |
|---|---|---|---|---|---|
| 1 | 4-Hydroxypropiophenone | 1 | 1 | 1 | 1 |
| 2 | 3-Hydroxybenzaldehyde | 1 | 1 | 1 | 1 |
| 3 | 4-(4-Hydroxyphenyl)-2-butanone | 1 | 1 | 1 | 1 |
| 4 | 4-Hydroxybenzaldehyde | 1 | 1 | 1 | 1 |
| 5 | 4-Isopropylphenol | 1 | 1 | 1 | 1 |
| 6 | 3-Fluoro-4-nitrophenol | 4 | 4 | 4 | 4 |
| 7 | Benzyl-4-hydroxybenzoate | 1 | 1 | 1 | 1 |
| 8 | 5-Pentylresorcinol | 1 | 1 | 1 | 1 |
| 9 | 2-Hydroxy-4-methoxyacetophenone | 1 | 1 | 1 | 1 |
| 10 | 3-Methyl-2-nitrophenol | 1 | 1 | 1 | 1 |
| 11 | 2-Ethylhexyl-4′-hydroxybenzoate | 1 | 1 | 1 | 1 |
| 12 | 2,3-Dinitrophenol | 2 | 2 | **1** | 2 |
| 13 | 2-Nitrophenol | 4 | 4 | 4 | 4 |
| 14 | 3-Methoxyphenol | 1 | 1 | 1 | 1 |
| 15 | 4-Chlororesorcinol | **3** | **3** | 1 | 1 |

## CONCLUSIONS

A set of five molecular descriptors (distribution coefficient, energy of the lowest unoccupied molecular orbital, molecular weight, negatively charged molecular surface area in percent's and sum of absolute charges on nitrogen and oxygen atoms in a molecule) is sufficient for correct classification of phenolic compounds by mechanisms their toxic effects.

The application of probabilistic neural network provides a reliable classification of phenolic compounds by mechanisms of their toxic action, as well as prediction of the mechanisms of their toxic action with high accuracy.

The proposed procedure for predicting the mechanisms of toxic action of phenolic compounds can be useful at the stage of development of medicines.

## REFERENCES

1. R.E. Mutha, A.U. Tatiya, S.J. Surana, "Flavonoids as natural phenolic compounds and their role in therapeutics: An overview," *FJPS*, vol. 7, article no. 25, pp. 1–13, 2021. doi: https://doi.org/10.1186/s43094-020-00161-8
2. A. Durazzo et al., "Polyphenols: A concise overview on the chemistry, occurrence, and human health," *Phytother. Res.*, vol. 33(9), pp. 2221–2243, 2019. doi: https://doi.org/10.1002/ptr.6419
3. M.M. Rahman et al., "Role of phenolic compounds in human disease: current knowledge and future prospects," *Molecules*, vol. 27(1), pp. 233, 2021. doi: https://doi.org/10.3390/molecules27010233
4. Y. Pushkarova, G. Zaitseva, M.Al Saker, "Prediction of Toxicity of Phenols Using Artificial Neural Networks," *IEEE — 12th International Conference on Advanced Computer Information Technologies (ACIT), Spisska Kapitula, Slovakia, 26–28 September 2022*, pp. 493–496. doi: https://doi.org/10.1109/ACIT54803.2022.9913174

5. N. Aptula et al., "Multivariate discrimination between modes of toxic action of phenols," *QSAR*, vol. 21(1), pp. 12–22, 2002. doi: https://doi.org/10.1002/1521-3838(200205)21:1<12::AID-QSAR12>3.0.CO;2-M
6. M.T. Cronin et al., "Comparative assessment of methods to develop QSARs for the prediction of the toxicity of phenols to Tetrahymena pyriformis," *Chemosphere*, vol. 49(10), pp. 1201–1221, 2002. doi: https://doi.org/10.1016/s0045-6535(02)00508-8
7. Y. Pushkarova, V. Panchenko, Y. Kholin, "Application an Artificial Neural Network for Prediction of Substances Solubility," *IEEE EUROCON 2021 − 19th International Conference on Smart Technologies, Lviv, Ukraine, 6–8 July 2021*, pp. 82–87. doi: https://doi.org/10.1109/EUROCON52738.2021.9535593
8. Y. Pushkarova, G. Zaitseva, A. Kaliuzhenko, "Classification of Residual Solvents by Risk Assessment Using Chemometric Methods," *IEEE — 13 th International Conference on Advanced Computer Information Technologies* (*ACIT*)*, Wroclaw, Poland, 21–22 September 2023*, pp. 562–565. doi: https://doi.org/10.1109/ACIT58437. 2023.10275405
9. E. Havránková, E.M. Pena-Mendez, J. Csöllei, J. Havel, "Prediction of biological activity of compounds containing a 1, 3, 5-triazinyl sulfonamide scaffold by artificial neural networks using simple molecular descriptors," *Bioorg. Chem.*, vol. 107, pp. 104565, 2021. doi: https://doi.org/10.1016/j.bioorg.2020.104565
10. S. Gummadi, P.K. Chandaka, "Chemometrics approach to drug analysis – An overview," *Am. J. Pharm. Tech. Res.*, vol. 9, pp. 1–13, 2019. doi: https://doi.org/10.46624/ajptr.2019.v9.i1.001
11. "MATLAB and Simulink for Engineered Systems," *mathworks.com*. Available: https://www.mathworks.com/
12. E. Ostertagova, O. Ostertag, J. Kováč, "Methodology and application of the Kruskal–Wallis test," *Appl. Mech. Mater.*, vol. 611, pp. 115–120, 2014. doi: https://doi.org/10.4028/www.scientific.net/AMM.611.115
13. D.F. Specht, "Probabilistic Neural Networks," *Neural Netw.*, vol. 3, pp. 109–118, 1990.
14. B. Mohebali, A. Tahmassebi, A. Meyer-Baese, A.H. Gandomi, "Probabilistic neural networks: a brief overview of theory, implementation, and application," *Handbook of probabilistic models*, pp. 347–367, 2020. doi: https://doi.org/10.1016/B978-0-12-816514-0.00014-X
15. Y. Zeinali, B.A. Story, "Competitive probabilistic neural network," *ICAE*, vol. 24(2), pp. 105–118, 2017. doi: https://doi.org/10.3233/ICA-170540
16. J. Miller, J.C. Miller, *Statistics and chemometrics for analytical chemistry*. Pearson education, 2018.

*Received 15.06.2024*

## INFORMATION ON THE ARTICLE

**Yaroslava M. Pushkarova**, ORCID: 0000-0001-9856-7846, Bogomolets National Medical University, Ukraine, e-mail: yaroslava.pushkarova@gmail.com

**Galina M. Zaitseva**, ORCID: 0000-0003-3138-6324, Bogomolets National Medical University, Ukraine, e-mail: galinazaitseva777@gmail.com

## ПРОГНОЗУВАННЯ МЕХАНІЗМІВ ТОКСИЧНОЇ ДІЇ ФЕНОЛІВ ЗА ДОПОМОГОЮ ЙМОВІРНІСНОЇ НЕЙРОННОЇ МЕРЕЖІ В ПОЄДНАННІ З ТЕСТОМ КРАСКЕЛА–УОЛЛІСА /

Я.М. Пушкарьова, Г.М. Зайцева

**Анотація.** Прогнозування токсичності хімічних сполук є одним із найважливіших етапів розроблення лікарських засобів. Використання фенольних сполук є перспективним компонентом у фармацевтичній промисловості з багатьма можливими застосуваннями. Працю присвячено застосуванню ймовірнісної нейронної мережі для класифікації 232 фенолів за механізмами їх токсичної дії. Для встановлення впливу молекулярних дескрипторів на достовірну класифікацію фенольних сполук за механізмами їх токсичної дії використали тест Краскела–Уолліса. Показано, що для коректного навчання ймовірнісної нейронної мережі та ефективного прогнозування механізмів токсичної дії фенолів достатньо використовувати лише 5 молекулярних дескрипторів.

**Ключові слова:** штучна нейронна мережа, класифікація, дизайн ліків, фенол, токсичність.

UDC 519.254+001.2
DOI: 10.20535/SRIT.2308-8893.2025.4.08

# ALGORITHMS FOR ASSIGNMENT OF EXTERNAL REVIEWERS FOR PHD-THESIS DEFENSE

**SERHIY SHTOVBA, MYKOLA PETRYCHKO**

**Abstract**. We propose an approach to assigning external reviewers. In the proposed approach, only the semantic similarity between applications and reviewers is taken into account; the similarity indices are assessed, and the necessary number of reviewers is assigned to ensure the maximum suitability level of the reviewers with the application, according to some criteria. We also perform a comparative analysis of various optimization algorithms using the criterion of "assignment quality–optimization time". Experiments on the dataset showed that a reasonable balance between the "assignment quality" and "optimization time" criteria for the assignment of external reviewers can be achieved using a greedy algorithm without elitism or brute-force search on a truncated set of candidates. An application of the proposed algorithms improves the average quality of PhD committees by 13–34% across the entire dataset, depending on the algorithm used.

**Keywords**: external reviewers, reviewer assignment problem, categorization, optimization, brute force algorithm, greedy algorithm, assignment in isolation, PhD-thesis, Dimensions, ANZSRC 2020, research group.

## INTRODUCTION

External reviewers are persons from outside an institution who are invited to provide an independent evaluation or assessment of a particular project, document, research paper, or system. They are often selected for their expertise in a relevant field and are expected to offer objective, unbiased feedback. In academia, external reviewers are used in the peer-reviewing to evaluate the quality, relevance, and originality of academic papers before publication. They may also be used for reviewing PhD-thesis.

In Ukraine, a PhD thesis is defended in front of a committee. A PhD-committee consists of 5 scientists with expertise in the thesis subject. The chairman and 1 or 2 reviewers are from the PhD-student's institution, and 2 or 3 external reviewers are invited from other institutions. The members of the PhD-committee are assigned manually, which has several disadvantages. First of all, there are corruption risks when the committee is formed exclusively from friendly persons who a priori give only favorable reviews regardless of the results of the thesis. Second, a lot of time is spent on manual search and analysis of candidates for the committee. Third, the combining competence of the committee may not fully correspond to the thesis topic due to the fact that some of the good

candidates were missed during the manual search. Therefore, there is an interest in automating the assignment of reviewers to eliminate the specified risks of the human factor influence.

The general task of assigning the reviewers consists of three stages [1]: 1) forming of a pool of potential reviewers and subsequently choosing a method of data representation for reviewers and applications; 2) assessing the similarities between the application and the reviewers; 3) assignment of applications to reviewers to maximize combined similarity across all the subjects with some constraints. Typical constraints include balancing reviewer workloads, taking into account their preferences, and preventing conflicts of interest. In this work, it is assumed that the pool of potential reviewers is available.

Automatic assignment of reviewers assumes that some initial information about reviewers and applications is available. A structured set of such information is called a reviewer profile and an application profile. The following information about reviewer's publications is used usually to build a reviewer's profile: title, abstract, keywords, full text, list of references, and list of citations [2]. Abstract, full text, keywords and title are most often used to create an application profile [2].

Applications' profiles and reviewers' profiles are built using various natural language processing methods based on bag of words [2; 3; 4], hidden semantic analysis [5; 6], topic modeling [7; 8], static language models with deep learning [9; 10; 11] and contextual models with deep learning [12]. Approaches to solving the problem of automatic assignment of reviewers in most cases require a fairly large amount of initial information about the reviewers' publications, their interaction with other scientists, and similar information about the authors of applications. Analyzing this information is costly and will not be expedient if thousands of candidates are to be analyzed in detail for each team of reviewers.

Our paper is dedicated to the assignment of external reviewers for PhD thesis defense. A candidate list of available internal reviewers is usually too short; hence it makes no sense to optimize it. We focus on the task of express assignment of external reviewers, where a long initial list of candidates is to be reduced drastically. The subsequent short list can be analyzed manually, or a fine assignment procedure can be activated, which is resource-intensive and requires a much larger volume of initial information than is required for express assignment. During express assignment, only the semantic similarity between applications and reviewers is taken into account, which provide the maximal level of collective competence of the committee. In this paper we perform comparative analysis of various optimization algorithms by using the criteria of *"assignment quality – optimization time"* in order to better understand the tradeoffs when choosing "assignment quality" over "optimization time" or vice versa.

## DATA REPRESENTATION

At the first stage of assigning the reviewers, it is necessary to choose the source data for decision-making, as well as the method of its representation in vector form. In the case of an application, a list of its keywords is used, and in the case of a reviewer, a list of keywords obtained from available data is used. In general, this list of keywords can be from the candidate's recent publications, from his CV or from a profile from some register of scientists. In the second case, keywords or

research interests are formed by the candidate at his own discretion, that is, they are presented in an arbitrary form without reference to any rubric or classifier.

The source data is usually processed using statistical models, topic models and embedding models. Some of them analyze the frequency of occurrence of words in the text, others form representation vectors based on the co-occurrence of words. Usually, the resulting vector representations are difficult to interpret. In addition, obtaining such representations requires a large amount of data. We suggest using the approach from [13], according to which a set of keywords is categorized as a vector in the space of research groups from the Australian and New Zealand Standard Research Classification — ANZSRC 2020. ANZSRC 2020 includes 171 research groups from 22 divisions. Therefore, the final representation of the application and reviewer profiles looks like a distribution over the 171 research groups from ANZSRC 2020.

In order to carry out a categorization, it is necessary to have a corpus of marked articles that are assigned to one or more research groups, and a machine learning model that, based on keywords, assigns the analyzed profile to certain research groups. We use the information resources of the Dimensions, in which more than 100M publications are already categorized according to ANZSRC 2020. For a search query in the form of a keyword, Dimensions produces an output that indicates how many publications with that keyword are assigned to each of the research group. This procedure is shown schematically in Fig. 1. It also shows that in the collection of marked documents an article can be categorized into several research groups, for example, *Article 1* is assigned to *Research Group 1* and *Research Group 2*. Based on this output, the distribution of a keyword's occurrence in the context of various research groups can be built. For example, for the keyword from Fig. 1 distribution looks like this: *Research Group 1* — 3 appearances, *Research Group 2* — 2 appearances, *Research Group 3* — 2 appearances, and *Research Group K* — 1 appearance. On the basis of this distribution, the keyword *"some keyword"* is further categorized within the framework of the research classification system. To categorize a set of keywords, the algorithm from [13] is applied, which is based on the resources and services of Dimensions. This algorithm takes into account both the occurrence of isolated keywords from a profile, as well as the co-occurrence of keyword pairs. The algorithm allows to filter the information noise caused by both stop words and rare keywords that have low reliability of the conclusions.

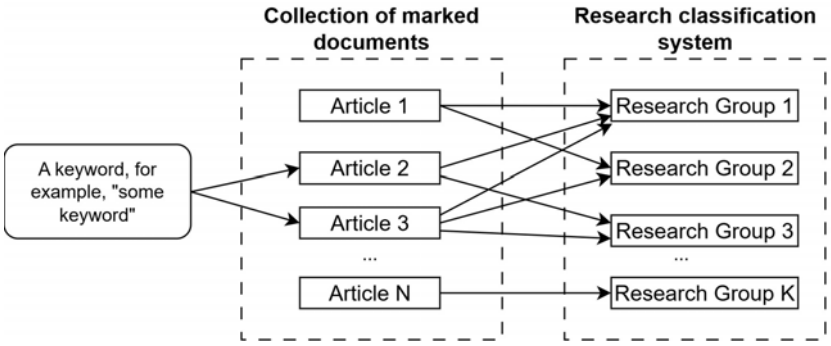



*Fig. 1.* Keyword categorization schema

The categorization algorithm consists of 3 stages. For a set of two keywords the procedure of categorization is schematically shown on Fig. 2. In the first stage the set *E* of search queries is created using the initial keywords and their pairwise

combinations. At the second stage the membership degrees of queries to research groups are computed. For this the overall distribution of the number of publications over research groups using Dimensions API is found. Then the same is done for each search query with subsequent stop-words detection and noise filtering. Having done this, the relative frequencies of search queries based on the overall distribution is found and the noise reduction using cumulative contribution of research groups is done. On the third stage all the queries distributions are averaged that produces one-dimensional vector. We further perform truncation to at most *RG_max* research groups with non-zero membership degree. A reviewer by the proposed algorithm can be categorized to at most *T_max* research groups, and the smallest membership degree is restricted to be at least *RG_min_degree*. The truncation is done in the last step of the third stage by removing research groups with low membership degree.

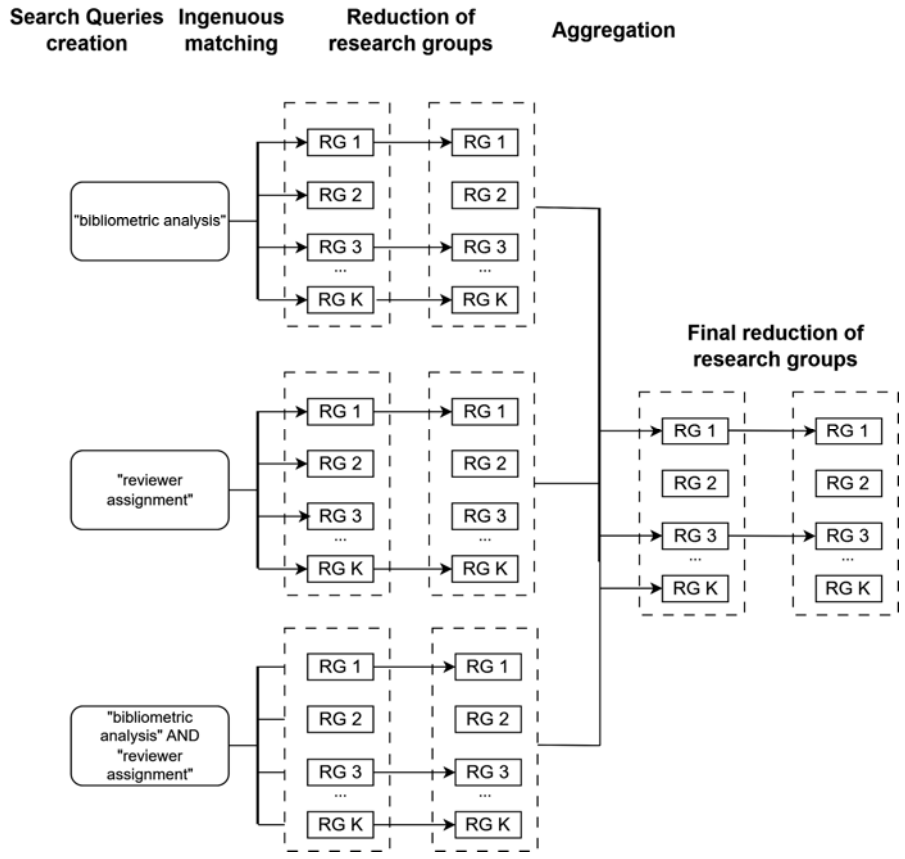



*Fig. 2.* Keywords detailed categorization schema

The MATLAB-style pseudocode of the categorization algorithm is as follows:

```
%STAGE #1 – creating the set E of search queries from the key-
% words w
E=w
for i=1:length(w)-1
     for j=i:length(w)
       E={E; [w(i) 'AND' w(j)] }
     end
end
%STAGE #2 – compute membership degrees to research groups by
% each query
```

```
< Find the total number of publications in each research groups
  N=[N(1), N(2), …, N(m)], m=171 >
Counter=0  % the counter of successful query responses
for i=1:length(E)
     < Find Q — the total number of publications in Dimensions,
       that contain E{i} >
     If Q>Threshold_StopWord continue   % ignoring the stop-
     words
     end
     If Q<Threshold_Noise continue;   % ignoring the rare key-
     words
     end
     < Find t(1), t(2), …, t(m) — the number of publications in
     each
       research group for query E{i} >
     %Ignoring the research group with a tiny number of publica
     % tions:
     index=find(t<Threshold_topic)
     t(index)=0
     if max(t)==0 continue
     end
     r=t./N   %frequency of E{i}'s occurrence in research groups
     %Normalizing the frequency distributions:
     Gamma=r./sum(r)
     < Choosing the most popular research groups that have cumu-
     lative
       contribution in Gamma >= Tail. ID-numbers of the remain-
     ing research groups
       are put in vector Rejected >
     %Ignoring the research groups with contribution lower than
     % Tail:
     Gamma(Rejected)=0
     Gamma=Gamma./sum(Gamma)   %normalizing again
     Counter=Counter+1
     Mu(Counter)=Gamma
end
If Counter==0 return ('Unsuccessful')
end
%STAGE #3 — compute membership degrees using all queries
Mu_mean=mean(Mu)   % averaging all successful queries
%Computing the current number of the selected research groups:
Current_N_RGs=sum(Mu_mean>0)
[Mu, RG_ID, Current_N_RGs]=Top_RG(Mu_mean, Source_RG_ID, RG_max)
% Top_RG — forms RG_ID as a selection of RG_max research groups
% with
% highest membership degree from Source_RG_ID. RG_ID is descend
% ing order
% list of research groups according to their membership degrees
% Mu.
% Vector Mu is normalized in [0; 1].
%Finish truncation based on kinship of research groups:
while (true)
       if (Current_N_RGs<=Tmax AND Mu(end)>RG_min_degree) break
       end
```

```
        if (Current_N_RGs<=1) break
        end
        < Drop the minor groups and redistribute its contribution
        to
          others based on their kinship >
        for target=1:Current_N_RGs-1
            akin_factor=Jaccard(RG_ID(target),
        RG_ID(Current_N_RGs))
            Mu(target)=Mu(target)+Mu(Current_N_RGs)*akin_factor
        end
[Mu, RG_ID, Current_N_RGs]=Top_RG(Mu, RG_ID, Current_N_RGs-1);
end
Return(Mu, RG_ID)
```

At the last stage of the algorithm when dropping a minor research group its contribution is redistributed to other research groups based on their kinship. The additional value is proportional to the kinship level between the target research group and the research group being removed. The kinship level is assessed using Jaccard index, where the size of the intersection is the number of publications categorized to belong to both research groups, and the size of the union is the number of publications categorized to either of research groups [14]. We formed the matrix of Jaccard indices for research groups using Dimensions API for the data period of 2019–2023. The intuition behind this step lays in the fact that we want to increase the influence of the subset of research groups that are more akin than others.

For example, a researcher is categorized tentatively to research groups *4410 Sociology*, *4611 Machine Learning*, *3508 Tourism*, and *3504 Commercial Services* as follows: $\left(\frac{0.4}{4410}, \frac{0.25}{4611}, \frac{0.2}{3508}, \frac{0.15}{3504}\right)$. Let us drop the minor research group *3504*. For this, we first compute Jaccard indices between *4609* and other research groups using the method from [14]. For the data of 2019–2023 they are:

$J(4410, 3504) = 0.044$;

$J(4611, 3504) = 0$;

$J(3508, 3504) = 0.478$.

By taking into account the kinships, the contribution of the research group *4609* is redistributed in the following way:

$$\left(\frac{0.4 + 0.044 \cdot 0.15}{4410}, \frac{0.25 + 0 \cdot 0.15}{4611}, \frac{0.2 + 0.478 \cdot 0.15}{3508}\right).$$

As a result, we get: $\left(\frac{0.466}{4410}, \frac{0.25}{4611}, \frac{0.271}{3508}\right)$. After norming: $\left(\frac{0.472}{4410}, \frac{0.275}{3508}, \frac{0.253}{4611}\right)$. As a result, research group *3508 Tourism* has been strongly reinforced. This research group is closely related to *3504 Commercial Services*, which has been eliminated. If we simply discard the minor research group, then after normalization we get $\left(\frac{0.47}{4410}, \frac{0.29}{4611}, \frac{0.24}{3508}\right)$. In this case, there was no additional reinforcement of the *3508* research group.

Let's present a step-by-step example of how the proposed algorithm works. For this, *Susan Dumais* is considered as a potential reviewer. The reviewer's information is taken from her Google Scholar profile that contains a set of research interests. Those interests may be interpreted as a set of initial keywords. For this reviewer the keywords are: *"Information Retrieval"*, *"Human-Computer Interaction"*. Interests often complement each other thus making the research topics more focused. To take this into account, additional keywords are synthesized as pairs of initial interests. Interests in a pair are combined by a logical operation AND as follows: *"Information Retrieval"* AND *"Human-Computer Interaction"*. Fig. 3 shows the initial distribution of membership degrees to research groups for the research interests of *Susan Dumais*. For each of the reviewer's interest and conjunction of her interests the distribution to research groups from Dimensions is found. Then the research groups with cumulative contribution less than *Tail* is dropped to reduce the noise (Fig. 4). *Tail* is set to be 0.93. The next step is to average over all interests' distribution (Fig. 5) and further restrict the max number of non-zero membership degrees to be at most *RG_max*. *RG_max* is set to be 12. The noise reduction steps and the restriction on the max number of non-zero membership degrees are based on the assumption that researchers usually are proficient only in a few research fields at once. In the end in case of $T_{\max} = 4$ *Susan Dumais* is represented by the following research groups:

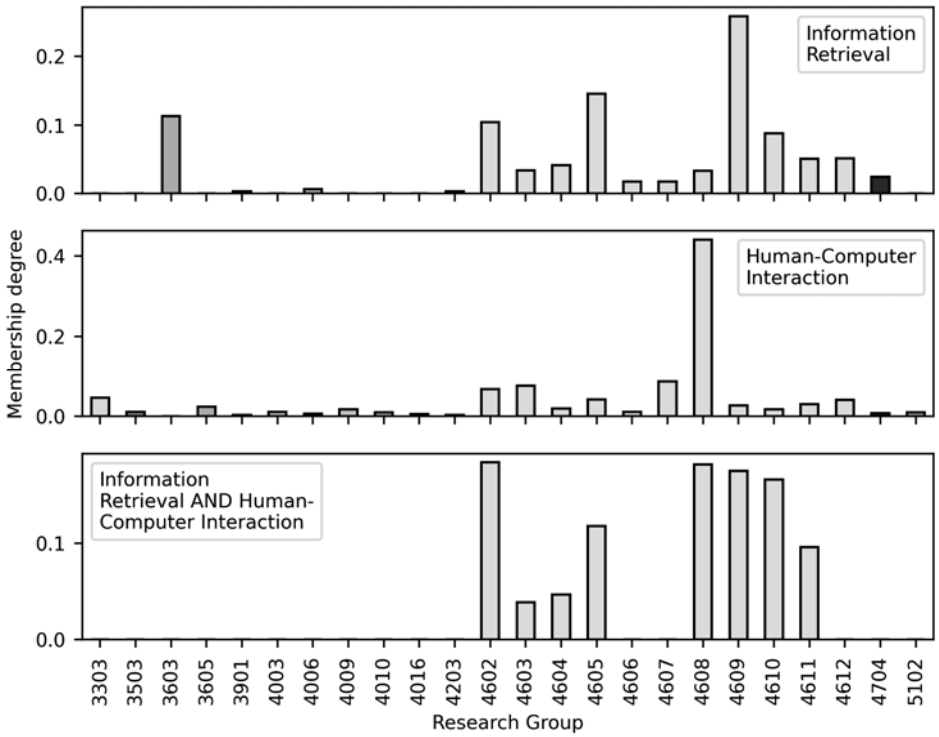



*Fig. 3.* The initial interests' distributions for *Susan Dumais*

| | |
|---|---|
| *4608 Human-Centred Computing* | with degree 0.35; |
| *4609 Information Systems* | with degree 0.25; |
| *4602 Artificial Intelligence* | with degree 0.21; |
| *4605 Data Management and Data Science* | with degree 0.19. |

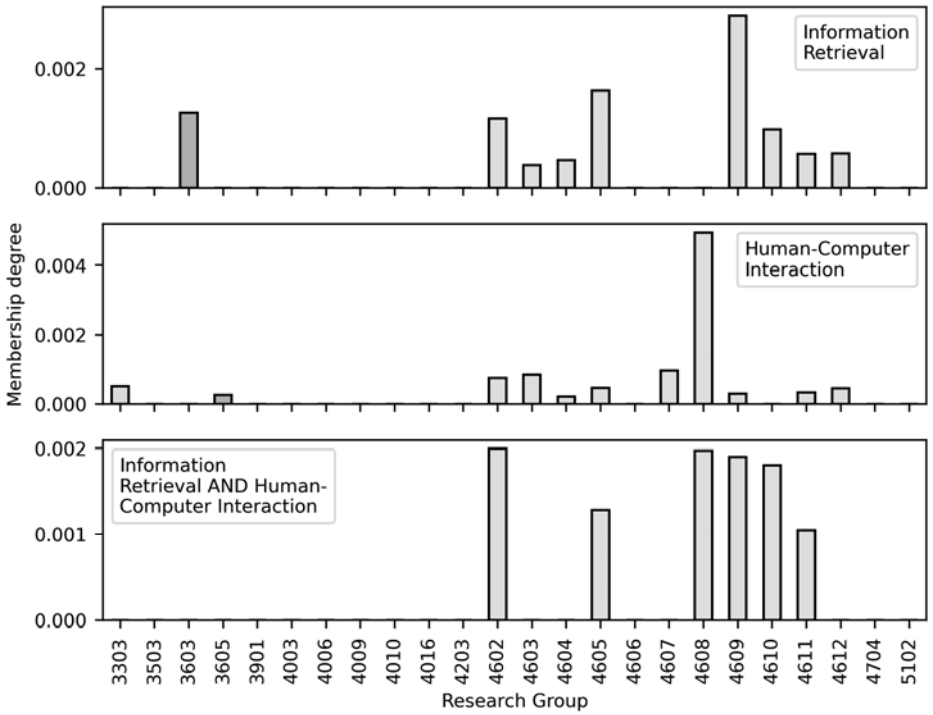



*Fig. 4.* Interests' distributions after filtering by *Tail*

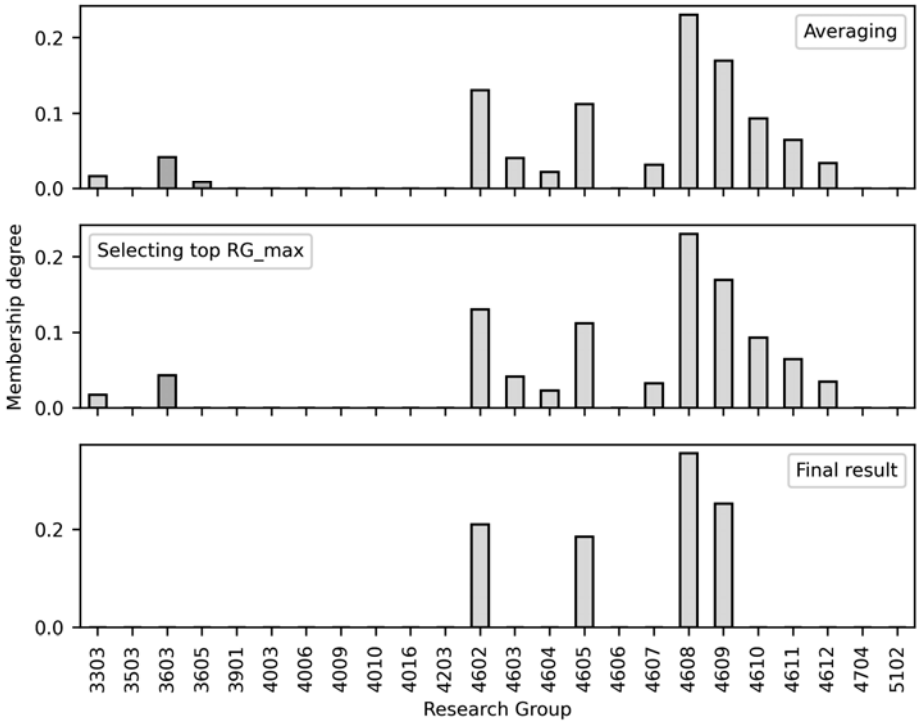



*Fig. 5.* Reviewer's distribution after averaging over all insterests' distributions and final result

As the result of categorization, an application profile, defined as a set of keywords $A_w = \{w_1, w_2, ..., w_n\}$, is transformed into a profile defined as a cate-

gorical distribution over research groups $A_t = \{\mu_{t_1}(A), \mu_{t_2}(A), ..., \mu_{t_m}(A)\}$, where $\mu_{t_i}(A) \in [0;1]$ denotes membership degree of application $A$ to research group $t_i$, $i = \overline{1, m}$. Similarly, a reviewer's profile, defined as a set of keywords or research interests $R_w = \{w_1, w_2, ..., w_n\}$, is transformed into a profile defined as a categorical distribution over research groups $R_t = \{\mu_{t_1}(R), \mu_{t_2}(R), ..., \mu_{t_m}(R)\}$.

## SIMILARITY ASSESMENT

To match reviewers and applications, a similarity metric between 2 categorical distributions, the reviewer keywords' research groups distribution and the application keywords' research groups distribution, has to be defined. For this, the metric from [15] is used. The metric calculates the similarity of two objects $X$ and $Y$ with the following categorical distributions $(\mu_1(X), \mu_2(X), ..., \mu_m(X))$ and $(\mu_1(Y), \mu_2(Y), ..., \mu_m(Y))$, where $m$ denotes the number of categories, that are research groups in our case, $\mu_i(X)$ denotes membership degree of object $X$ to $i$-th category, $\mu_i(Y)$ denotes membership degree of object $Y$ to $i$-th category, $i = \overline{1, m}$. Distributions are normalized and satisfy the following conditions:

$$\mu_i(X) \in [0;1], \quad \mu_i(Y) \in [0;1], \quad i = \overline{1, m};$$
$$\sum_{i=\overline{1, m}} \mu_i(X) = 1; \quad \sum_{i=\overline{1, m}} \mu_i(Y) = 1.$$

The categorical distributions of objects $X$ and $Y$ look like two fuzzy sets on universal sets of all categories. Therefore, to calculate the similarity of objects $X$ and $Y$, it is proposed to use an intersection of the corresponding fuzzy sets. This is reflected in the metric [15], according to which the similarity of objects $X$ and $Y$ is defined as follows:

$$Fit(X, Y) = \sum_{i=\overline{1, m}} \min(\mu_i(X), \mu_i(Y)) + \Delta F(X, Y), \quad (1)$$

where $\sum_{i=\overline{1, m}} \min(\mu_i(X), \mu_i(Y))$ is an addend that evaluates the direct similarity of objects $X$ and $Y$; $\Delta F(X, Y)$ is an addend that evaluates the similarity of objects $X$ and $Y$ through akin categories (akin research groups in our case). Across the all research groups, kinship is conveniently represented by a binary fuzzy relationship in the form of an $m \times m$ matrix. Each element of the matrix corresponds to the kinship level of two corresponding research groups. An identification of this kinship matrix is easily performed by the method [14], which uses the Jaccard index on data from Dimensions.

## TASK STATEMENT OF ASSIGNMENT OPTIMIZATION

Consider the task of assigning a team of reviewers, who are collectively the best suited for reviewing an application. For this task, 2 cases are possible: forming a team from scratch and supplementing the team with new members.

Given: an application profile $A_t = \{\mu_{t_1}(A), \mu_{t_2}(A), ..., \mu_{t_m}(A)\}$ and profiles of *k*-th potential reviewers $R_{tj} = \{\mu_{t_1}(R_j), \mu_{t_2}(R_j), ..., \mu_{t_m}(R_j)\}$, $j = \overline{1,k}$ in the space of *m* research groups. The entire set of reviewers is denoted as $\mathbf{R} = \{R_1, R_2, .., R_k\}$.

Find out: subset of reviewers $S \subset \mathbf{R}$ with the highest overall suitability level to all the topics of the application:

$$Fit(A, Agg(S)) \to \max,$$

where $Agg(S)$ denotes aggregation function of categorical distributions of the assigned reviewers set.

Aggregation of categorical distributions by reviewer profiles $R_{tj}$, $j = \overline{1,k}$ in the space of research groups from ANZSRC 2020 is implemented using the third stage of the above described categorization algorithm.

The number of reviewers for an application is denoted by $c = |S|$. This quantity is constant; usually it is from 2 to 5 people. The level of suitability between the application and the team of reviewers is calculated by formula (1).

**REVIEWER ASSIGNMENT ALGORITHMS**

The task of assigning reviewers from a mathematical point of view is to find a subset of fixed cardinality. To solve such problems in practice, mostly approximate algorithms are used. Among the set of possible algorithms, it is necessary to choose the one that provides a balance between assignment quality and efforts for solution finding. The following algorithms are proposed to be used.

*Brute force*. The best solution can be found by trivial brute force. For application *A*, among all possible teams of size *c* from the reviewers set **R,** a team with the maximum level of suitability has to be found. The complexity of brute force grows exponentially. The number of operations is proportional to the binomial coefficient: $\frac{n!}{(n-c)!c!}$. So even for medium-sized problems, it is unrealistic to walk through all possible options and adhere to some time constraints. Moreover, the number of options depends very much on the *c*.

*Brute force on a truncated set of candidates*. In practice, candidates with a low level of similarity are unlikely to be assigned as reviewers. Therefore, the rational step would be to ignore potential reviewers with very low similarity. By rejecting candidates with low similarity to the application, for example, at the level of 0.1 or 0.2, the search time can be significantly reduced. The number of operations is still proportional to the binomial coefficient but on a much smaller set of reviewers: $n \cdot p(r > truncation\_level)$, where $p(r > truncation\_level)$ is the probability that a reviewer *r* will have at least $truncation\_level$ similarity level with the application. The more we thin out the initial list of candidates, the shorter the duration of optimization will be, but the risks of deviating far from the optimum increase.

*Pure greedy algorithm*. The reviewers are assigned iteratively to ensure at each step the maximum suitability of the current fragment of the team to the application. The algorithm is performed in *c* iterations. At each iteration, one new

member is added to the team of reviewers, who at this iteration maximizes the level of combined suitability of the current composition with the application. In the first iteration, we find the candidate with the highest similarity to the application. In the second iteration, we choose the candidate who, together with the already selected member of the team, has the highest suitability level to the application. The number of operations with this approach is significantly reduced and is proportional to $n^c$, but the solution may turn out to be suboptimal.

*Greedy algorithm with elitism.* The candidate with the highest value of suitability to the application is added first. At the same time, the level of combined suitability of updated reviewer team to the application is not taken into account. Other reviewers are assigned according to the pure greedy algorithm, that is, candidates are assigned who, in the current iteration, maximize the team's suitability level to the application. The greedy algorithm with elitism significantly shortens the duration of the optimization but still is proportional to $n^{c-1}$.

*Assignment in isolation*. The easiest way to assign reviewers is to choose those who are the most similar to the application. The combined suitability of the team is not taken into account. It is assumed that the stronger each of the candidates corresponds to the application, the better the team will be. Roughly speaking, the combined suitability level of the team is considered to be the sum of the similarity levels of each member. Algorithmically, assignment in isolation is implemented by sorting the candidates in descending order of similarity to the application and selecting the first $c$ candidates. The number of operations is proportional to $n \cdot c$ in the best case. This is a very fast algorithm, but with a small chance of getting to the optimum.

## DATASET FOR ASSIGNING EXTERNAL REVIEWERS

For experiments on the assignment of external reviewers, a dataset of PhD-thesis was collected [16]. For this, the information system of Ukrainian National Agency for Higher Education Quality Assurance was used. The collected theses belong to various research fields (Fig. 6) with the predominance of *Information Technologies*.

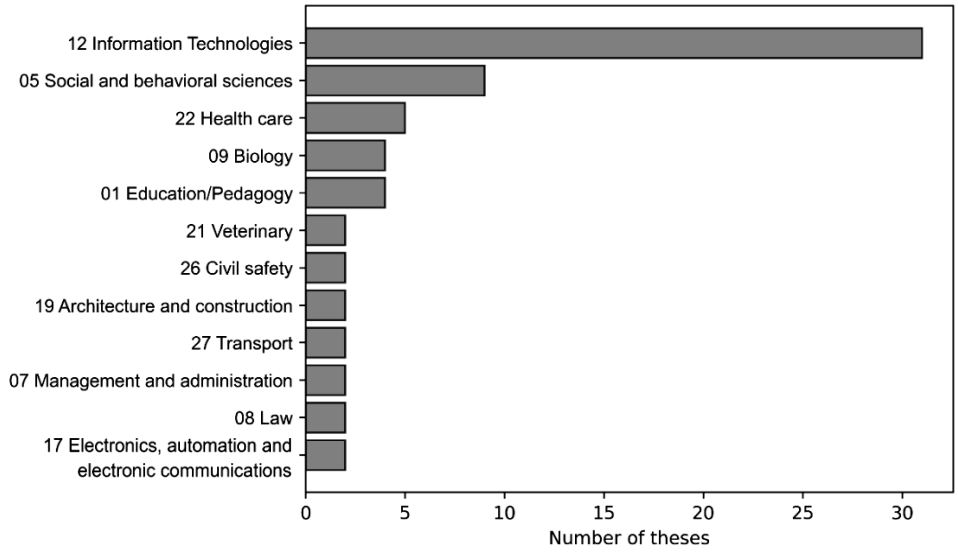



*Fig. 6.* PhD-theses distribution over research fields

**EXPERIMENTS ON ASSIGNING EXTERNAL REVIEWERS**

Experiments on external reviewers' assignment are conducted on the formed dataset of theses. At first, a thesis's keywords are categorized according to the keyword categorization algorithm within the research groups from ANZSRC 2020. Next, in a similar way, the keywords of the articles of the committees' members are categorized. Pairs of keywords are combined into additional queries only within one article. For each committee, the external reviewers are removed and new ones are assigned from other committees to maximize combined suitability. After removing the external reviewers, we get a set of fragments of committees, containing the chairman and two or one internal reviewers. The task is to find external reviewers whose addition to the fragments of committee ensures their maximum of combined suitability level to the topic of the theses.

The results of the reviewers' assignment are compared with the version of the committee, which is formed by the institution. The effect is estimated by an average level of change in the suitability level of committees:

$$E(F^{new}, F^{current}) = \frac{\sum\limits_{i=\overline{1,N}} (F_i^{new} - F_i^{current})}{\sum\limits_{i=\overline{1,N}} F_i^{current}} \cdot 100\%,$$

where $N$ denotes number of theses; $F_i^{new}$ denotes suitability level of the committee for *i*-th thesis after optimization, $i = \overline{1,N}$; $F_i^{current}$ denotes suitability level of the committee for *i*-th thesis before optimization, $i = \overline{1,N}$.

Fig. 7 presents the results of optimization using various assignment algorithms. Most of the committees from institutions have the suitability level

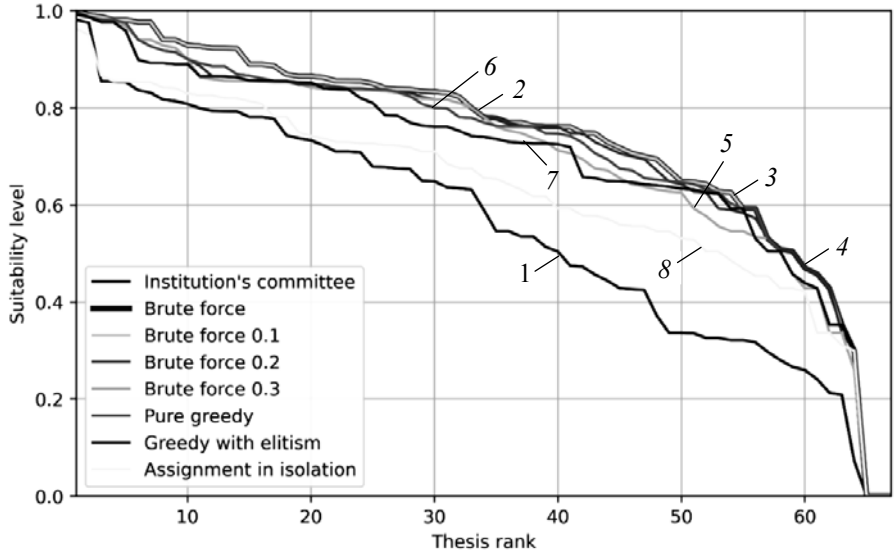



*Fig. 7.* Distribution of committees' suitability level depending on the algorithm used

above 0.2. The interquartile range is approximately equal to [0.4; 0.8]. With brute force there is a significant improvement in the suitability levels for the majority of committees. Some committees are not improved or the improvement level is low. This is due primarily to the fact that the distribution of theses by fields in the dataset is uneven and the dataset has a relatively small size. In almost all cases,

committees from institutions have a lower suitability level to thesis than found by any assignment algorithm. By manually creating committees with limited opportunities for choosing committee's members, we get an average level of suitability to the thesis. On the other hand, with the automatic assignment of committee's members and a sufficiently large pool of candidates, we get a significant improvement of the committees only by changing external reviewers.

Fig. 8 compares suitability levels of committees' found by brute force with the committees found by other algorithms including brute force on a truncated set of candidates. Brute force on a truncated set of candidates with similarity threshold 0.1 performs almost identically as regular brute force, but the optimization time is reduced (Fig. 9). Brute force on a truncated set of candidates with similarity thresholds 0.2 and 0.3 performs very similar to the regular brute force, but there are a few suboptimal committees in both cases. Committees found by pure greedy algorithm are also suboptimal. Its performance is very close to the brute force 0.2 and is somewhat better than the brute force 0.3, but the time of optimization is significantly better (Fig. 9). Greedy algorithm with elitism performs slightly worse than pure greedy algorithm, there are slightly more suboptimal committees, but it is close to the brute force 0.3 with the optimization time reduced (Fig. 9). Under the assignment in isolation, most of the committees are suboptimal but it is the fastest among the algorithms (Fig. 9). This is due to the fact that the high similarity of a candidate with a thesis does not mean that the team formed by assignment in isolation covers the entire research groups' distribution of the thesis.

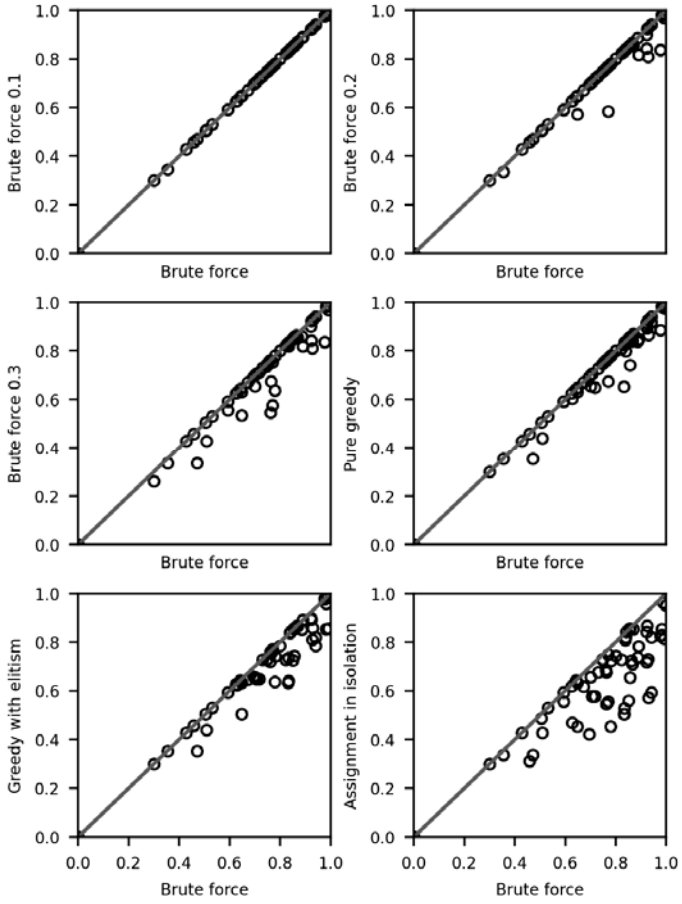



*Fig. 8.* Comparison of committees found by brute force with the committees found by faster algorithms

Fig. 9 compares the results of committees' assignments according to various optimization algorithms. Optimizing the truncated set of candidates with the similarity threshold of 0.3 is clearly unsuccessful. All others form a Pareto set. Therefore, when choosing an algorithm, it is necessary to take into account priorities, what is needed — a quick result or a high-quality one. From Fig. 9, it can be seen that the level of change due to the skip from pure greedy algorithm to brute force algorithms grows slowly. But the optimization time increases significantly. Therefore, the pure greedy assignment algorithm can be considered the most balanced. An alternative to it can be the brute force on truncated set of candidates with the similarity threshold in the vicinity of 0.25. These conclusions are based on experiments on a small dataset. With real databases of large volume, the optimization time by brute force algorithms can increase drastically.

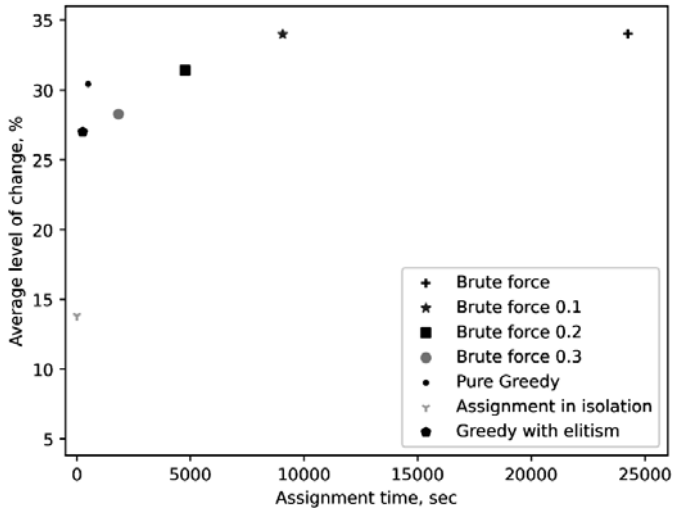



*Fig. 9.* Comparison of assignment algorithms according to the "duration — quality" criteria

**AN EXAMPLE OF ASSIGNING A COMMITTEE**

Let's consider an example of assigning a committee for the following thesis: "Models and methods of data processing of the system of remote monitoring of the condition of patients with diabetes". The thesis identifier in National Agency for Higher Education Quality Assurance is 4756.

The thesis's keywords are: edge devices; IoT; diagnostics; diseases; intelligent data analysis; information technologies; medical information systems; modeling; monitoring; data processing; patient; forecasting; software component model; system design; diabetes. After categorizing these keywords, we get the following result:

*4605 Data Management and Data Science* — 0.382;
*4606 Distributed Computing and Systems Software* — 0.255;
*4609 Information Systems* — 0.205;
*4203 Health Services and Systems* — 0.158.

The thesis is represented by the following vector:

$$A_t = \left( \frac{0.382}{4605}, \frac{0.255}{4606}, \frac{0.205}{4609}, \frac{0.158}{4203} \right)$$

In National Agency for Higher Education Quality Assurance, the research topics of each committee member are represented by the keywords of 3 or 4 of his/her papers. To categorize them, the principle of a bag of keywords is applied. Categorization of a member takes place as follows: 1) for each set of keywords of one paper, their paired combinations is created; 2) the received sets of keywords of different papers are combined into into one bag; 3) categorize the received set of keywords according to the algorithm [13]. The result of the committee categorization is as follows.

Research groups of the chairman are:

*4609 Information Systems* — 0.381;
*4203 Health Services and Systems* — 0.225;
*4606 Distributed Computing and Systems Software* — 0.214;
*4601 Applied Computing* — 0.180.

Suitability level of the chairman is:

$$Fit\left(\left(\frac{0.382}{4605},\frac{0.255}{4606},\frac{0.205}{4609},\frac{0.158}{4203}\right),\left(\frac{0.381}{4609},\frac{0.225}{4203},\frac{0.214}{4606},\frac{0.180}{4601}\right)\right)=0.577.$$

Research groups of the first inner reviewer are:

*4606 Distributed Computing and Systems Software* — 0.337;
*4605 Data Management and Data Science* — 0.256;
*4003 Biomedical Engineering* — 0.244;
*3208 Medical Physiology* — 0.162.

Suitability level of the first inner reviewer is:

$$Fit\left(\left(\frac{0.382}{4605},\frac{0.255}{4606},\frac{0.205}{4609},\frac{0.158}{4203}\right),\left(\frac{0.337}{4606},\frac{0.256}{4605},\frac{0.244}{4003},\frac{0.162}{3208}\right)\right)=0.564.$$

Research groups of the second inner reviewer are:

*4606 Distributed Computing and Systems Software* — 0.426;
*4605 Data Management and Data Science* — 0.299;
*4003 Biomedical Engineering* — 0.138;
*4604 Cybersecurity and Privacy* — 0.135.

Suitability level of the second inner reviewer is:

$$Fit\left(\left(\frac{0.382}{4605},\frac{0.255}{4606},\frac{0.205}{4609},\frac{0.158}{4203}\right),\left(\frac{0.426}{4606},\frac{0.299}{4605},\frac{0.138}{4003},\frac{0.135}{4604}\right)\right)=0.521.$$

Research groups of the first external reviewer are:

*3201 Cardiovascular Medicine and Haematology* — 0.387;
*3203 Dentistry* — 0.215;
*4605 Data Management and Data Science* — 0.205;
*4602 Artificial Intelligence* — 0.192.

Suitability level of the first external reviewer is:

$$Fit\left(\left(\frac{0.382}{4605},\frac{0.255}{4606},\frac{0.205}{4609},\frac{0.158}{4203}\right),\left(\frac{0.387}{3201},\frac{0.215}{3203},\frac{0.205}{4605},\frac{0.192}{4602}\right)\right)=0.239.$$

Research groups of the second external reviewer are:

*4602 Artificial Intelligence* — 0.435;
*4611 Machine Learning* — 0.357;
*4605 Data Management and Data Science* — 0.208.

Suitability level of the second external reviewer is:

$$Fit\left(\left(\frac{0.382}{4605},\frac{0.255}{4606},\frac{0.205}{4609},\frac{0.158}{4203}\right),\left(\frac{0.435}{4602},\frac{0.357}{4611},\frac{0.208}{4605}\right)\right)=0.227.$$

The result of the committee aggregation is as follows:

$$Agg\left(\begin{array}{l}\left(\frac{0.381}{4609},\frac{0.225}{4203},\frac{0.214}{4606},\frac{0.180}{4601}\right)\\ \left(\frac{0.337}{4606},\frac{0.256}{4605},\frac{0.244}{4003},\frac{0.162}{3208}\right)\\ \left(\frac{0.426}{4606},\frac{0.299}{4605},\frac{0.138}{4003},\frac{0.135}{4604}\right)\\ \left(\frac{0.387}{3201},\frac{0.215}{3203},\frac{0.205}{4605},\frac{0.192}{4602}\right)\\ \left(\frac{0.435}{4602},\frac{0.357}{4611},\frac{0.208}{4605}\right)\end{array}\right)=\left(\frac{0.389}{4606},\frac{0.374}{4605},\frac{0.236}{4602}\right).$$

The combined suitability level of the committee to the thesis is $Fit\left(\left(\frac{0.382}{4605},\frac{0.255}{4606},\frac{0.205}{4609},\frac{0.158}{4203}\right),\left(\frac{0.389}{4606},\frac{0.374}{4605},\frac{0.236}{4602}\right)\right)=0.631$. This is a relatively good suitability level, which is mainly due to the strong overlap in two of the four research groups.

Let's try to choose the best external reviewers to increase the combined suitability level. The members of all other committees of the dataset are used as candidates. As the result of brute force, the two new external reviewers are found. Their profiles are as follows: $\left(\frac{0.274}{3210},\frac{0.261}{4203},\frac{0.251}{4202},\frac{0.214}{3205}\right)$ with suitability level 0.158, and $\left(\frac{0.555}{4605},\frac{0.306}{4611},\frac{0.139}{4609}\right)$ with suitability level 0.542. After aggregating all members of the new committee we get the following categorization:

$$Agg\left(\begin{array}{l}\left(\frac{0.281}{4611},\frac{0.269}{4605},\frac{0.228}{4602},\frac{0.222}{4608}\right)\\ \left(\frac{0.556}{4612},\frac{0.302}{4602},\frac{0.142}{4007}\right)\\ \left(\frac{0.457}{4611},\frac{0.196}{4603},\frac{0.183}{4605},\frac{0.163}{4609}\right)\\ \left(\frac{0.274}{3210},\frac{0.261}{4203},\frac{0.251}{4202},\frac{0.214}{3205}\right)\\ \left(\frac{0.555}{4605},\frac{0.306}{4611},\frac{0.139}{4609}\right)\end{array}\right)=\left(\frac{0.361}{4605},\frac{0.335}{4606},\frac{0.156}{4609},\frac{0.148}{4203}\right).$$

The combined suitability level of the new committee to the thesis is

$$Fit\left(\left(\frac{0.382}{4605},\frac{0.255}{4606},\frac{0.205}{4609},\frac{0.158}{4203}\right),\left(\frac{0.361}{4605},\frac{0.335}{4606},\frac{0.156}{4609},\frac{0.148}{4203}\right)\right)=0.923.$$

Comparing with the initial committee, a significant improvement in the level of suitability is observed, the new committee has the same research groups as the thesis. The improvement is about 46%.

From the given example, it can be seen that although the individual similarity of an individual member of a committee may be mediocre, the overall suitability level of the committee may turn out to be high. This is due to the fact that the new external reviewers cover the so-called minor part of the thesis topic, which is outside the field of expertise of other committee members. This is clearly visible on Fig. 10 where the difference between the distributions of thesis, institution's committee and proposed committee is shown. The thesis and proposed committee intersect in all their research groups. The institution's committee lacks the research groups *4609 Information Systems* and *4203 Health Services and Systems,* which makes it less similar to the thesis's research field.

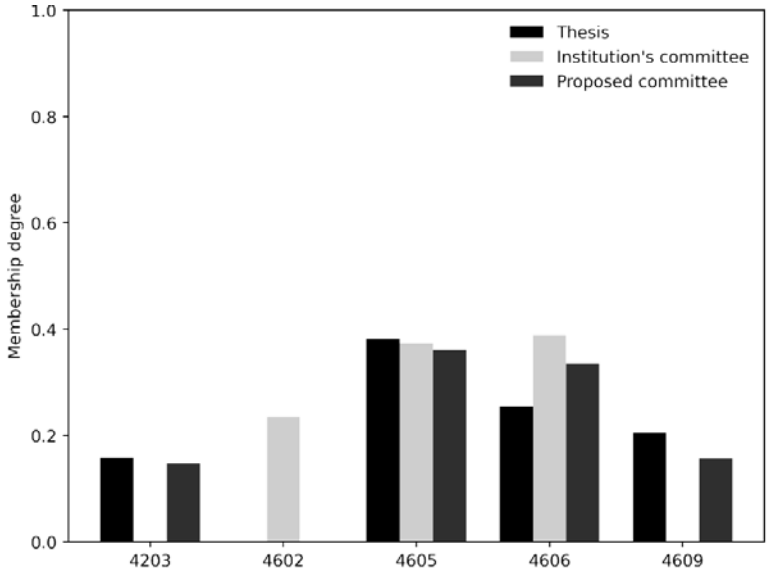



*Fig. 10.* Comparison of initial committee and proposed committee

## CONCLUSIONS

The paper proposes an express method of assigning the external reviewers for PhD defense committee. On the first stage of assignment, the application and potential reviewers are categorized by presenting their profiles as vectors in the space of research groups from ANZSRC 2020. At the second stage, the suitability levels of potential reviewers to the application topic are calculated, taking into account the kinship of research groups. At the third stage, a team of reviewers is assigned, which corresponds to the topic of the application to the maximum possible extent. To implement the third stage, the various optimization algorithms are proposed: brute force, brute force on a truncated set of candidates, greedy algorithm without elitism and with elitism, and on assignment in isolation. Experiments on the dataset of 67 PhD theses showed that the best balance in terms of assignment quality criteria and team searching duration provides greedy algorithm without elitism and brute force on a truncated set of candidates. As a result of the optimization, it was possible to improve the combined quality of committees by an average of 13–34% over all the dataset, depending on the type of algorithm used. Optimizing the truncated set of candidates with the similarity threshold of

0.3 is clearly unsuccessful. All others form a Pareto set. Therefore, when choosing an algorithm, it is necessary to take into account priorities, what is needed — a quick result or a high-quality one.

The proposed method can be used to improve the efficiency of managing the processes of assigning reviewer teams in various fields, for example, for evaluation of grant applications. The method can also be used for auditing to quickly check the correctness of the assigned committees with subsequent thorough resource-intensive examination of suspicious cases.

Further research may include: studying whether using Large Language Models is a better choice for modeling the keywords representation than the proposed method; using the proposed method of express assignment in more time-consuming and iterative procedures for assigning a team of reviewers, when it is necessary to take into account not only the relevance of the topic of the application, but also the absence of a conflict of interests, the balance of the load on the reviewers, and other possible limitations. It is advisable to take into account not only the relevance of the subject of the reviewers and the application, but also the qualification level of the experts during the assignment.

**Acknoledgment.** The authors are grateful to Digital Science & Research Solutions Inc. for the provision of access to Dimensions as part of the DIM-371 project.

## REFERENCES

1. F. Wang, N. Shi, B. Chen, "A comprehensive survey of the reviewer assignment problem," *International Journal of Information Technology and Decision Making*, 9(4), pp. 645–668, 2010. doi: https://doi.org/10.1142/S0219622010003993
2. M. Aksoy, S. Yanik, M.F. Amasyali, "Reviewer assignment problem: A systematic review of the literature," *Journal of Artificial Intelligence Research*, vol. 76, 2023. doi: https://doi.org/10.1613/JAIR.1.14318
3. S. Tan, Z. Duan, S. Zhao, J. Chen, Y. Zhang, "Improved reviewer assignment based on both word and semantic features," *Information Retrieval Journal*, 24(3), pp. 175–204, 2021. doi: https://doi.org/10.1007/s10791-021-09390-8
4. D. Yarowsky, R. Florian, "Taking the load off the conference chairs: Towards a digital paper-routing assistant," *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP 1999*, pp. 220–230.
5. M. Karimzadehgan, C.X. Zhai, G. Belford, "Multi-aspect expertise matching for review assignment," *Proceedings of International Conference on Information and Knowledge Management*, pp. 1113–1122, 2008. doi: https://doi.org/10.1145/ 1458082.1458230
6. M. Mirzaei, J. Sander, E. Stroulia, "Multi-aspect review-team assignment using latent research areas," *Information Processing and Management*, 56(3), pp. 858–878, 2019. doi: https://doi.org/10.1016/j.ipm.2019.01.007
7. E. Ekinci, S.I. Omurca, "NET-LDA: A novel topic modeling method based on semantic document similarity," *Turkish Journal of Electrical Engineering and Computer Sciences*, 28(4), pp. 2244–2260, 2020. doi: https://doi.org/10.3906/ELK-1912-62
8. O. Anjum, H. Gong, S. Bhat, J. Xiong, W.M. Hwu, "Pare: A paper-reviewer matching approach using a common topic space," *EMNLP-IJCNLP 2019 – 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 518–528. doi: https://doi.org/10.18653/v1/d19-1049
9. C. Sun, K.T.J. Ng, P. Henville, R. Marchant, "Hierarchical word mover distance for collaboration recommender system," *Communications in Computer and Information Science*, vol. 996, pp. 289–302. Springer Verlag, 2019. doi: https://doi.org/10.1007/978-981-13-6661-1_23
10. X. Kong, H. Jiang, Z. Yang, Z. Xu, F. Xia, A. Tolba, "Exploiting publication contents and collaboration networks for collaborator recommendation," *PLOS One*, 11(2): e0148492, 2016. doi: https://doi.org/10.1371/journal.pone.0148492
11. B. Bhaisare, R. Bharati, "Advancing Peer Review Integrity: Automated Reviewer Assignment Techniques with a Focus on Deep Learning Applications," in *A.K. Bairwa, V. Tiwari, S.K. Vishwakarma, M. Tuba, T. Ganokratanaa, (eds) Computation of Artificial Intelligence and*

*Machine Learning. ICCAIML 2024. Communications in Computer and Information Science*, vol 2184. Springer, Cham, 2024. doi: https://doi.org/10.1007/978-3-031-71481-8_25

12. Y. Zhao, J. Tang, Z. Du, "EFCNN: A restricted convolutional neural network for expert finding," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11440 LNAI, pp. 96–107. Springer Verlag, 2019. doi: https://doi.org/10.1007/978-3-030-16145-3_8
13. S. Shtovba, M. Petrychko, "Topic modeling of researchers based on their interests from Google Scholar," *System Research and Information Technologies*, no. 2, pp. 113–129, 2021. doi: https://doi.org/10.20535/SRIT.2308-8893.2021.2.09
14. S. Shtovba, M. Petrychko, "Jaccard index-based assessing the similarity of research fields in dimensions," *CEUR Workshop Proceedings*, vol. 2533, pp. 117–128, 2019.
15. S. Shtovba, M. Petrychko, O. Shtovba, "Similarity metric of categorical distributions for topic modeling problems with akin categories," *CEUR Workshop Proceedings*, vol. 3392 "The Sixth International Workshop on Computer Modeling and Intelligent Systems", pp. 76–85, 2023. doi: https://doi.org/10.32782/cmis/3392-7
16. M. Petrychko, S. Shtovba, "Dataset for PhD theses reviewers' assignments," *ResearchGate*, 2024. doi: http://dx.doi.org/10.13140/RG.2.2.23147.35362

*Received 25.11.2024*

## INFORMATION ON THE ARTICLE

**Serhiy D. Shtovba,** ORCID: 0000-0003-1302-4899, Vasyl' Stus Donetsk National University, Vinnytsia National Technical Universiry, Ukraine, e-mail: s.shtovba@donnu.edu.ua

**Mykola V. Petrychko,** ORCID: 0000-0001-6836-7843, Vinnytsia National Technical Universiry, Ukraine, e-mail: mpetrychko@vntu.edu.ua

**АЛГОРИТМИ ПРИЗНАЧЕННЯ ЗОВНІШНІХ РЕЦЕНЗЕНТІВ ДЛЯ ЗАХИСТУ PHD-ДИСЕРТАЦІЙ** / С.Д. Штовба, М.В. Петричко

**Анотація.** Запропоновано підхід до призначення зовнішніх рецензентів. У ньому враховується лише семантична схожість між заявками та рецензентами, оцінюються індекси схожості та призначається необхідна кількість таких рецензентів, за яких забезпечується максимальний рівень відповідності рецензентів заявці за деякими критеріями. Виконано порівняльний аналіз різних алгоритмів оптимізації за критерієм «якість призначення – тривалість оптимізації». Експерименти на тестовому датасеті показали, що прийнятний баланс за критеріями «якість призначення» та «тривалість оптимізації» для призначення зовнішніх рецензентів забезпечує жадібний алгоритм без елітизму та за повного перебору на проріджений множині кандидатів. Застосування запропонованих алгоритмів покращує якість роботи докторських рад в середньому на 13–34% за усього набору даних, залежно від типу використовуваного алгоритму.

**Ключові слова:** зовнішні рецензенти, задача призначення рецензентів, категоризація, оптимізація, повний перебір, жадібний алгоритм, ізольоване призначення, PhD-дисертація, Dimensions, ANZSRC 2020, галузь досліджень.

## ВІДОМОСТІ ПРО АВТОРІВ

**Беляк Євген В'ячеславович,**
старший науковий співробітник, кандидат технічних наук, старший науковий співробітник Інституту проблем реєстрації інформації НАН України, Київ

**Бідюк Петро Іванович,**
професор, доктор технічних наук, професор кафедри математичних методів системного аналізу НН ІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

**Заварзіна Валентина Володимирівна,**
асистент кафедри інформаційно-аналітичної діяльності та інформаційної безпеки Національного транспортного університету, Україна, Київ

**Зайцева Галина Миколаївна,**
доцент, кандидат хімічних наук, завідувачка кафедри аналітичної, фізичної та колоїдної хімії Національного медичного університету імені О.О. Богомольця, Україна, Київ

**Іщенко Руслан Миколайович,**
доцент, кандидат фізико-математичних наук, доцент кафедри інформаційно-аналітичної діяльності та інформаційної безпеки Національного транспортного університету, Україна, Київ

**Крючин Андрій Андрійович,**
професор, доктор технічних наук, заступник директора з наукової роботи Інституту проблем реєстрації інформації НАН України, Київ

**Ланько Анна Анатоліївна,**
магістр за освітньо-професійною програмою «Системний аналіз фінансового ринку» спеціальності 124 «Системний аналіз», НН ІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

**Манько Дмитро Юрійович,**
старший дослідник, кандидат фізико-математичних наук, старший науковий співробітник Інституту проблем реєстрації інформації НАН України, Київ

**Медяков Олександр Олександрович,**
аспірант кафедри інформаційних систем і мереж Національного університету «Львівська політехніка», Україна, Львів

**Мороз Володимир Володимирович,**
доцент, кандидат технічних наук, професор кафедри оптимального керування та економічної кібернетики факультету математики, фізики та інформаційних технологій Одеського національного університету імені І.І. Мечникова, Україна, Одеса

**Недашківська Надія Іванівна,**
доцент, доктор технічних наук, професор кафедри математичних методів системного аналізу НН ІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

**Панібратов Роман Сергійович,**
аспірант кафедри штучного інтелекту НН ІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

**Петричко Микола Володимирович,**
доктор філософії, старший викладач кафедри комп'ютерних систем управління Вінницького національного технічного університету, Україна, Вінниця

**Попов Олександр Олександрович,**
полковник, командир Військової частини А1108, Україна, Дрогобич

**Пушкарьова Ярослава Миколаївна**
доцент, кандидат хімічних наук, доцент кафедри аналітичної, фізичної та колоїдної хімії Національного медичного університету імені О.О. Богомольця, Україна, Київ

**Спекторський Ігор Якович,**
доцент, кандидат фізико-математичних наук, доцент кафедри математичних методів системного аналізу НН ІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

**Статкевич Віталій Михайлович,**
кандидат фізико-математичних наук, науковий співробітник відділу прикладного нелінійного аналізу ННК «ІПСА» КПІ ім. Ігоря Сікорського, Україна, Київ

**Стусь Олександр Вікторович,**
кандидат фізико-математичних наук, доцент кафедри математичних методів системного аналізу НН ІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

**Тимчук Володимир Юрійович,**
старший науковий співробітник, кандидат технічних наук, старший науковий співробітник науково-дослідного відділу розвитку автоматизації Сухопутних військ науково-дослідного управління розвитку озброєння та військової техніки наукового центру Сухопутних військ Національної академії Сухопутних військ, Україна, Львів

**Триснюк Тарас Васильович,**
кандидат технічних наук, старший науковий співробітник Інституту телекомунікацій і глобального інформаційного простору НАН України, Київ

**Цибуля Сергій Анатолійович,**
кандидат технічних наук, старший дослідник, начальник науково-дослідного відділу проблем супроводження експлуатації інформаційних систем науково-дослідного управління проблем розвитку інформаційних технологій та впровадження проєктів інформатизації Збройних Сил України Центру воєнно-стратегічних досліджень Національного університету оборони України, Україна, Київ

**Швандт Максим Альбертович,**
аспірант кафедри оптимального керування та економічної кібернетики факультету математики, фізики та інформаційних технологій Одеського національного університету імені І.І. Мечникова, Україна, Одеса

**Штовба Сергій Дмитрович,**
професор, доктор технічних наук, професор кафедри інформаційних технологій Донецького національного університету імені Василя Стуса та професор кафедри комп'ютерних систем управління Вінницького національного технічного університету, Україна, Вінниця

Зміст журналу
**«Системні дослідження та інформаційні технології»**
за 2025 р.

**ЗМІСТ № 1**



**ЗМІСТ № 2**

**ЗМІСТ № 3**



**ЗМІСТ № 4**

## АВТОРИ СТАТЕЙ ЗА 2025 РІК