



ISSN 2224-087X

# ЕЛЕКТРОНІКА ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

ELECTRONICS  
AND INFORMATION TECHNOLOGIES

**Збірник наукових праць**  
**Випуск 31**



2025

**ELECTRONICS  
AND  
INFORMATION  
TECHNOLOGIES**

**Issue 31**

Scientific journal

Published 4 issue per year

*Published since 1966*

**ЕЛЕКТРОНІКА  
ТА  
ІНФОРМАЦІЙНІ  
ТЕХНОЛОГІЇ**

**Випуск 31**

Збірник наукових праць

Виходить 4 рази на рік

*Видається з 1966 р.*

**Ivan Franko National  
University of Lviv**

**Львівський національний  
університет імені Івана Франка**

**2025**

## ЗАСНОВНИК: ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Друкується за ухвалою Вченої Ради  
Львівського національного університету  
імені Івана Франка  
протокол №21/10 від 29.10.2025 р.

У 1966–2010 рр. збірник виходив під назвою «Теоретична електротехніка»

Збірник «Електроніка та інформаційні технології» містить оригінальні результати досліджень з електронного матеріалознавства, моделювання фізичних явищ, процесів і систем електроніки, обробки сигналів і зображень, інформаційних технологій.

### Редактори

Проф. *І. Карбовник* – головний співредактор  
Проф. *І. Бордун* – головний співредактор  
Проф. *О. Крупиш* – відповідальний редактор  
С.н.с. *Я. Шмигельський* – відповідальний секретар

### Редакційна колегія

д-р техн. наук, проф. *О. Андрейків*  
д-р габіл., проф. *Б. Андрієвський*  
д-р фіз.-мат. наук, проф. *І. Болеста*  
канд. фіз.-мат. наук, доц. *С. Вельгош*  
д-р фіз.-мат. наук, проф. *П. Венгерський*  
д-р техн. наук, проф. *Р. Воробель*  
д-р фіз.-мат. наук, проф. *Р. Головчак*  
д-р техн. наук, д-р габіл., проф. *Ф. Івацішин*  
д-р фіз.-мат. наук, проф. *О. Кушнір*  
д-р фіз.-мат. наук, проф. *А. Лучечко*  
д-р техн. наук, проф. *Л. Муравський*  
д-р техн. наук, проф. *М. Назаркевич*  
д-р фіз.-мат. наук, проф. *І. Оленич*  
д-р фіз.-мат. наук, проф. *Б. Павлик*  
д-р техн. наук, доц. *Б. Павлишенко*  
д-р техн. наук, проф. *С. Рендзіняк*  
д-р фіз.-мат. наук, проф. *Б. Русин*  
д-р фіз.-мат. наук, проф. *М. Причула*  
д-р габіл., проф. *Ц. Славінські*  
канд. фіз.-мат. наук, доц. *Ю. Фургала*  
д-р габіл., проф. *Б. Ціж*  
д-р габіл., проф. *Бушита Шахраї*  
д-р фіз.-мат. наук, проф. *Г. Шинкаренко*  
канд. фіз.-мат. наук, доц. *Р. Шувар*  
д-р фіз.-мат. наук, проф. *І. Яворський*

### Адреса редакційної колегії:

Львівський національний університет імені Івана  
Франка, факультет електроніки та інформаційних  
технологій, вул. Ген. М. Тарнавського, 107,  
79017, Львів, Україна  
тел. (+38) (093) 864-01-19

е-mail: [elit@lnu.edu.ua](mailto:elit@lnu.edu.ua)

web-сайт: <http://publications.lnu.edu.ua/collections/index.php/electronics/index>

Реєстрація суб'єкта у сфері друкованих медіа:  
Рішення Національної ради України з питань  
телебачення і радіомовлення № 1877 від  
30.05.2024 р. Ідентифікатор медіа R30-04912

У 1966–2010 рр. збірник виходив під назвою «Теоретична електротехніка»

“Electronics and information technologies” journal contains original research results on electronics material science, modelling of physical phenomena, processes and electronic systems, signal and image processing and information technologies.

### Editors

Prof. *I. Karbovnyk* – Chief Co-Editor  
Prof. *I. Bordun* – Chief Co-Editor  
Prof. *O. Krupych* – Managing Editor  
Sen. Res. *Ya. Shmygelsky* – Technical Editor

### Editorial Board

*O. Andreykiv*, Dr. Sc., Prof.  
*B. Andrievsky*, Dr. Habil., Prof.  
*I. Bolesta*, Dr. Sc., Prof.  
*S. Velgosh*, PhD, Assoc. Prof.  
*P. Vengersky*, Dr. Sc., Prof.  
*R. Vorobel*, Dr. Sc., Prof.  
*R. Holovchak*, Dr. Sc., Prof.  
*F. Ivachyshyn*, Dr. Sc., Dr. Habil., Prof.  
*O. Kushnir*, Dr. Sc., Prof.  
*A. Luchechko*, Dr. Sc., Prof.  
*L. Muravsky*, Dr. Sc., Prof.  
*M. Nazarkevych*, Dr. Sc., Prof.  
*I. Olenych*, Dr. Sc., Prof.  
*B. Pavlyk*, Dr. Sc., Prof.  
*B. Pavlyshenko*, Dr. Sc., Assoc. Prof.  
*S. Rendzinyak*, Dr. Sc., Prof.  
*B. Rusyn*, Dr. Sc., Prof.  
*M. Prytula*, Dr. Sc., Prof.  
*C. Slawinski*, Dr. Habil., Prof.  
*Yu. Furgala*, PhD, Assoc. Prof.  
*B. Tsizh*, Dr. Habil., Prof.  
*Bouchta SAHRAOUI*, Dr. Habil., Prof.  
*G. Shynkarenko*, Dr. Sc., Prof.  
*R. Shuvar*, PhD, Assoc. Prof.  
*I. Yavorsky*, Dr. Sc., Dr. Habil., Prof.

### Editorial office address:

Ivan Franko National University of L'viv,  
Faculty of Electronics and Computer Technologies  
107 Tarnavsky St., UA–79017,  
Lviv, Ukraine  
tel. (+38) (093) 864-01-19

### АДРЕСА РЕДАКЦІЇ, ВИДАВЦЯ І ВИГОТОВЛЮВАЧА:

Львівський національний університет імені Івана Франка  
вул. Університетська, 1, 79000 Львів, Україна

Свідчення про внесення суб'єкта видавничої справи до Державного реєстру видавців,  
виготівників і розповсюджувачів видавничої продукції. Серія ДК № 3059 від 13.12.2007 р.

© Львівський національний університет імені Івана Франка, 2025

## CONTENTS

### INFORMATION SYSTEMS AND TECHNOLOGIES

Named entity recognition using generative transformer models with sentence-level data augmentation approaches. <i>Ihor Drozdov, Bohdan Pavlyshenko (14)</i> .....	5
Stochastic analysis of CPU scheduling in Apple M-series. <i>Bohdan Mikh, Yuriy Korchak (12)</i> .....	19
A multifunctional smartclock for voice interaction and adaptive task scheduling. <i>Dmytro Kozliuk, Halyna Klym (14)</i> .....	33
Features of designing software distributed systems architecture. <i>Ivan Rovetskii (8)</i> .....	45
Intelligent methods for data analysis in information and communication systems monitoring processes. <i>Andrii Senyk, Volodymyr Kotsun, Bohdan Penyukh, Bohdan Tsybulyak (8)</i> .....	53
Modular approach to building a hardware-software platform for smart home automation: from simple rules to intelligent scenarios. <i>Olha Shymchyshyn, Maryan Shymchyshyn, Vladyslav Serhiienko (10)</i> .....	61

### MODELING OF PROCESSES AND EFFECTS

Comparative study of feature detectors and filtering methods in image matching. <i>Andriy Fesiuk, Yuriy Furgala (18)</i> .....	71
Predicting quantitative characteristics of air pollution. <i>Volodymyr Hura, Igor Olenych, Oleh Sinkevych, Oksana Ostrovska, Roman Shuvar (16)</i> .....	89
Physics-informed neural networks for inverse tasks of one-dimensional wave propagation. <i>Igor Kolych, Roman Shuvar (10)</i> .....	105
Prospects of using the wave function collapse algorithm for improving heuristic search strategies. <i>Denys-Roman Rudyk, Oleksii Kushnir (8)</i> .....	115
Remote monitoring system for microclimate parameters in beehives. <i>Yurii Zborivskyi, Bohdan Koman (14)</i> .....	123

### MATERIALS FOR ELECTRONIC ENGINEERING

Electronic properties of Ga-doped As-Se-Te glasses. <i>Yaroslav Shpotyuk, Adam Ingram, Andriy Luchechko, Dmytro Slobodzyan, Markiyan Kushlyk, Oleh Kravets, Mykhaylo Shpotyuk, Roman Golovchak (8)</i> .....	137
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----



## ЗМІСТ

### ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Розпізнавання іменованих сутностей із використанням генеративних трансформерних моделей з аугментацією даних на рівні речень. <i>Ігор Дроздов, Богдан Павлишенко (14)</i> .....	5
Стохастичне дослідження планувальника задач Apple M-series. <i>Богдан Міх, Юрій Корчак (12)</i> .....	19
Багатофункціональний розумний годинник для голосової взаємодії та адаптивного планування завдань. <i>Дмитро Козлюк, Галина Клим (14)</i> .....	31
Особливості проектування архітектури програмних розподілених систем. <i>Іван Ровецький (12)</i> .....	45
Дослідження інтелектуальних методів аналізу даних у процесах моніторингу інформаційно-комунікаційних систем. <i>Андрій Сеник, Володимир Коцун, Богдан Пенюх, Богдан Цибуляк (8)</i> .....	53
Модульний підхід до побудови апаратно-програмної платформи автоматизації розумного будинку: від простих правил до інтелектуальних сценаріїв. <i>Ольга Шимчишин, Мар'ян Шимчишин, Владислав Сергієнко (10)</i> .....	61

### МОДЕЛЮВАННЯ ПРОЦЕСІВ ТА ЯВИЩ

Порівняльне дослідження детекторів особливих точок та методів фільтрації у зіставленні зображень. <i>Андрій Фесюк, Юрій Фургала (18)</i> .....	71
Прогнозування кількісних характеристик забруднення повітря. <i>Володимир Гура, Ігор Оленич, Олег Сінкевич, Оксана Островська, Роман Шувар (16)</i> .....	89
Фізично-інформовані нейронні мережі для оберненої задачі поширення одновимірних хвиль. <i>Ігор Колич, Роман Шувар (10)</i> .....	105
Перспективи використання алгоритму колапсу хвильової функції для вдосконалення евристичних стратегій пошуку. <i>Денис-Роман Рудик, Олексій Кушнір (16)</i> .....	115
Система дистанційного моніторингу параметрів мікроклімату у вулицях. <i>Юрій Зборівський, Богдан Коман (14)</i> .....	123

### МАТЕРІАЛИ ЕЛЕКТРОННОЇ ТЕХНІКИ

Електронні властивості скла As-Se-Te, легованого Ga. <i>Ярослав Шпотюк, Адам Інграм, Андрій Лучечко, Дмитро Слободзян, Маркіян Кушлик, Олег Кравець, Михайло Шпотюк, Роман Головчак (8)</i> .....	137
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----

UDC: 004.91

## NAMED ENTITY RECOGNITION USING GENERATIVE TRANSFORMER MODELS WITH SENTENCE-LEVEL DATA AUGMENTATION APPROACHES

Ihor Drozdov<sup>\*</sup>, Bohdan Pavlyshenko<sup>\*</sup>

Department of System Design,  
Ivan Franko National University of Lviv  
50 Drahomanova St., UA-79005 Lviv, Ukraine

Drozdov, I. V., Pavlyshenko, B.M. (2025). Named Entity Recognition Using Generative Transformer Models with Sentence-Level Data Augmentation Approaches. *Electronics and Information Technologies*, 31, 5–18. <https://doi.org/10.30970/eli.31.1>

### ABSTRACT

**Background.** Named entity recognition, as one of the key tasks of the natural language processing (NLP) field, plays a vital role in the processing and understanding of the texts. Usage of transformer-based models demonstrates exceptional performance on most NLP tasks but requires a considerable amount of information for practical model training. Building a high-quality annotated dataset for named entity recognition is resource-intensive, especially for low-resourced languages. Using data augmentation to extend the annotated dataset with synthetic data provides an opportunity to increase the efficiency of the models for named entity recognition. This study aims to use sentence-level augmentations and large language models to improve model performance on small datasets.

**Materials and Methods.** To investigate the impact of data augmentation, 5%, 10%, and 20% of training data from the CoNLL and Ontonotes5 datasets with different characteristics were taken. Three main approaches were used to construct the augmented data: summarizing sentences using the T5 model, followed by inserting named entities, paraphrasing sentences using the OpenAI Api, and several methods of replacing named entities in initial and synthetic sentences. BERT, ALBERT, DistilBERT, and RoBERTa models were used for evaluation.

**Results and Discussion.** According to the results, the effectiveness of using different augmentation methods significantly depends on the initial dataset and its quality. For small datasets with few categories for recognition, sentence-level augmentation methods through summarization or paraphrasing improve the efficiency of models by up to 10%. On the other hand, with an increase in the size of the dataset, artificially created data can lead to a deterioration in recognition results.

**Conclusion.** Using data augmentation to recognize named entities is an effective tool for small datasets and can improve model performance in resource-constrained cases like specific domains and low-resourced languages. However, synthetic data cannot fully replace a larger, better-built original dataset through context extension for existing named entities and the generation of new, synthetic entities.

**Keywords:** named entity recognition, natural language processing, data augmentation, large language models

### INTRODUCTION

Natural Language Processing (NLP) is a field of research that plays a crucial role in developing and improving existing information processing methods. As one of the fundamental tasks in NLP, Named Entity Recognition (NER) is essential for text



© 2025 Ihor Drozdov & Bohdan Pavlyshenko. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

understanding and extracting key information from textual data. Accurate recognition of entities like persons, organizations, locations, dates, and other categories is crucial for various NLP tasks like question answering, machine translation, sentiment analysis, and the construction of comprehensive knowledge bases [1, 2]. Recent advances in transformer-based architectures like BERT, RoBERTa, and others have substantially enhanced NLP tasks' performance in general and NER in particular by effectively capturing contextual and linguistic nuances within large-scale textual data [3, 4, 5]. Despite these advances, these models remain highly dependent on high-quality annotated datasets, and the lack of sufficient labeled data often impacts their effectiveness, especially in specialized domains and low-resourced languages.

Recent developments in generative Large Language Models (LLMs), such as GPT (OpenAI), LLaMA (Meta), and Mistral, provide significant advances for most of the NLP tasks. These models leverage unstructured textual data to extract and understand sophisticated language patterns. As a result, LLMs provide prompt-based communication in a native language, excellent text understanding capabilities, text generation, and advanced reasoning. Nevertheless, the adoption of LLMs is often limited by their computational resource requirements for request processing and model fine-tuning, very high associated infrastructure costs, and request processing speed, restricting their practical usage for many applications and smaller-scale deployments [6, 7]. Aside from this, LLMs' great reasoning possibilities greatly benefit smaller models' data preparation and fine-tuning.

One of the primary challenges in effectively training transformer-based models such as BERT or RoBERTa is the creation of sufficiently large datasets with high-quality annotated data. Typically, annotation is performed manually by data labeling specialists or domain experts. This process becomes even more complex when dealing with highly specialized domains or low-resource languages. Data augmentation offers a promising alternative for expanding labeled datasets by generating synthetic data [8, 9]. Data augmentation research has become increasingly popular in recent years and often requires selecting the most effective methods for specific tasks and application domains. Most augmentation techniques can be broadly categorized according to their application level: character level, word level, sentence level, and document level [10, 11]. Some methods are more general-purpose, while others are effective for specific classes of tasks. Nevertheless, selecting appropriate augmentation techniques for a given task can be challenging. In [12, 13], augmentation methods that have demonstrated exemplary performance for text classification tasks are reviewed; however, not all these methods are effective in the context of named entity recognition tasks.

The scope of this paper is to extend our previous research [14] from word-level data augmentations to sentence-level augmentations like text summarization, entity injection, and context-dependent LLM-based augmentations, generated by OpenAI models.

## MATERIALS AND METHODS

In the scope of this research, CoNLL 2003 [15] and Ontonotes 5 [16] were used. Despite these datasets being introduced more than ten years ago, both are very popular for NER research as they provide an outstanding possibility to compare with other authors' research results. **Table 1** provides an overview of the CoNLL dataset, the number of sentences, and entities. It has four entity types: Person (PER), Location (LOC), Organization (ORG), and Miscellaneous (MISC). **Table 2** provides information about the Ontonotes5 dataset which has 18 different entity types: Person (PER), Facility, Nationalities/Religious/Political groups, GPE (Geo-political entity), Organizations (ORG), Locations (not GPE), Date (DATE), Time, etc. During this research, six subsets of the original training dataset were used, as demonstrated in **Table 3**.

**Table 1. Information about the CoNLL 2003 dataset.**

CoNLL 2003 dataset, English						
Set type	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	14,041	203,621	7,140	3,438	6,321	6,600
Validation set	3,250	51,362	1,837	922	1,341	1,842
Test set	3,453	46,435	1,668	702	1,661	1,617

**Table 2. Information about the Ontonotes5 dataset.**

Ontonotes5 dataset, English							
Set type	Sentences	Tokens	PER	ORG	GPE	DATE	Other*
Training set	59,924	1,088,442	6,292	5,363	1790	3,533	8,695
Validation set	8,528	147,718	1,071	914	506	799	1,641
Test set	8,262	152,723	977	950	462	767	1,750

**Note:** \* – the “Other” column contains the total number of entities for the remaining 14 entity types.

**Table 3. Partial training subsets used in this work. Validation and test parts are unchanged.**

Abbreviation	Sentences CoNLL	Sentences Ontonotes5	Description
S100	14,041	59,924	Contains all sentences from the original dataset without changes
S20	2,808	11,984	Contains 20% of the initial train dataset records: the first 10% of sentences and the last 10%
S10	1,404	5,992	Contains 10% of the initial train dataset records: the first 5% of sentences and the last 5%
S5	702	2,996	Contains only the first 5% of the dataset
R10	1,404	5,992	Contains 10% of the initial train dataset chosen by random
R5	702	2,996	Contains 5% of the initial train dataset chosen by random

Within the scope of this study, the main attention was paid to the creation of additional context through paraphrasing or summarization of sentences, the use of LLMs to create sentences close in context to the original ones, with the same entities from the original sentence. Additionally, two scenarios were used: replace named entities in the augmented sentence with random ones from the train dataset, and the second scenario – replace named entities in the augmented sentence with random ones generated by LLM.

For each data augmentation approach described above, the training sub-dataset term relates to the part of the dataset that was used during the training session. For example, dataset R5 (**Table 3**) has only 5% of randomly selected initial sentences from the original

dataset and is a training sub-dataset for the R5 dataset from the original one. Augmentations were applied to sentences from the training sub-dataset with the multiplier provided in **Table 4**. Produced augmentations and original sentences were added to the new augmented training sub-dataset. Thus, the following data augmentation scenarios were used:

- Replace entities for random ones from the training sub-dataset (RND\_ENT). With this augmentation type, the original positions and types of the named entities persisted. Still, each entity was replaced with a new, random one in the training sub-dataset.
- Rephrasing with OpenAI LLMs (OPENAI\_SENT) – with this augmentation type, the OpenAI API was used to build the new sentences as a rephrasing of the original one using the same named entities, which exist in the original sentence. As a result, the new sentence extends the context for the original entities.
- Rephrasing or summarization with sentence entities (SUMM\_SENT) – with this augmentation type, Text-to-Text summarization or rephrasing was used using the T5-base model [17] to build required sentences via sampling. All named entities from the original sentence were injected into random places for each newly generated sentence.
- Rephrasing with OpenAI LLMs with train sub-dataset entities (OPENAI\_SDE) – this augmentation type was based on OPENAI\_SENT augmentation. Random train sub-dataset entities were used to replace the existing ones in each.
- Rephrasing or summarization with train sub-dataset entities (SUMM\_SDE) – this augmentation type is the same as OPENAI\_SDE, but utilizes SUMM\_SENT
- Rephrasing or summarization with OpenAI entities (SUMM\_GENT) – this augmentation type was built the same way as SUMM\_SDE but using OpenAI-generated entities.
- Rephrasing with OpenAI LLMs with OpenAI entities (OPENAI\_GENT) – this augmentation type was built the same way as OPENAI\_SDE but using OpenAI-generated entities.

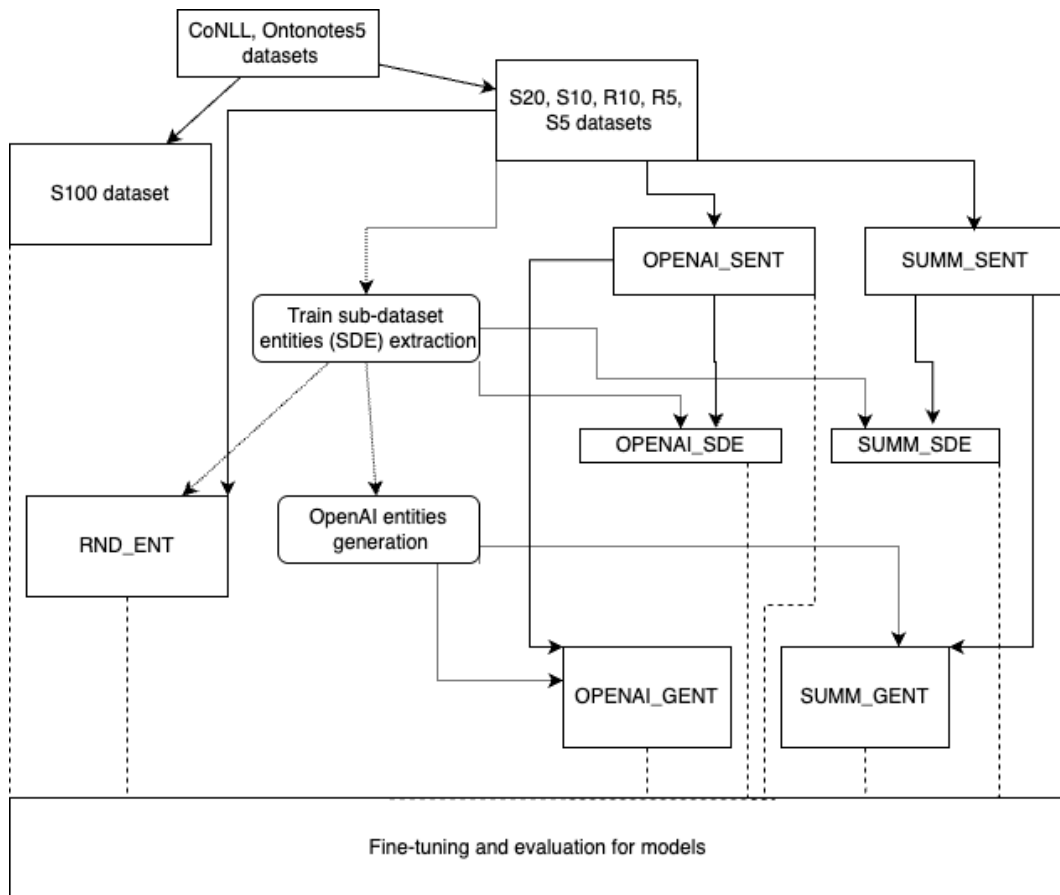
Moreover, in [14] word-level augmentations like synonyms, antonyms, and word-embeddings were applied to the CoNLL dataset. During this experiment, the same data augmentation approaches on word-level were applied for the Ontonotes5 dataset to have the possibility to compare word-level augmentations with sentence-level ones for both datasets.

At the same time, our main objective is to investigate the impact of data augmentation on the most popular transformer-based models. Thus, BERT, RoBERTa, DistilBERT, and ALBERT models were used for fine-tuning during the experiment [14]:

- BERT model – one of the first transformer-based models, demonstrated state-of-the-art in multiple NLP tasks compared to previous approaches.
- ALBER and DistilBERT models aim to optimize the initial BERT model implementation with parameter optimization, faster training time, a smaller model, and similar performance due to more effective model utilization.
- RoBERTa model – also, a BERT-based model, optimizes the initial model, introduces additional dynamic masking, and uses a significantly bigger initial training dataset.

### Experiment structure

Experiments were built and executed around the HuggingFace platform [18] as it provides valuable tools to load and manipulate datasets, prepare pipelines for dataset preparation, fine-tune process configuration, and model training and evaluation. **Fig.1** demonstrates a general scheme of the experiment flow to build the required datasets with augmentation and fine-tune models. The experiment has the following key points:



**Fig. 1.** Datasets generation and model fine-tuning scheme. Solid arrows – dataset was built from parent; Dotted arrows – relation to some additionally generated data to use; Dashed lines – flow direction to indicate which datasets were used for fine-tuning

- Dataset S100 was used only to receive the score of the models on the full dataset and compare it with the augmented ones.
- **Table 4** contains information about the number of additional augmented sentences per augmentation type, additionally generated for the augmented dataset. Original sentences were also included in the augmented dataset.
- The train sub-dataset entities (SDE) extraction process involves identifying all annotated named entities and constructing a dictionary for each type from the train sub-dataset.

**Table 4. Total augmentations count per augmented dataset, generated with different approaches.**

Scenarios	Augmentations count
RND_ENT	2, 5, 10
OPENAI_SENT, OPENAI_SDE, OPENAI_GENT	1, 3, 5
SUMM_SENT, SUMM_SDE, SUMM_GENT	3, 5

- OpenAI API [19] was used for all LLM prompt-based requests with the gpt-4.1-mini model.
- OpenAI's entity generation process uses entities from the train sub-dataset as a context and generates up to 4 additional entities per existing one with the same context.
- OPENAI\_SENT and SUMM\_SENT were built with additional sentences as described in **Table 4**. OPENAI\_SDE, OPENAI\_GENT, SUMM\_SDE, and SUMM\_GENT utilize augmented datasets from OPENAI\_SENT and SUMM\_SENT, but apply random entity replacements based on the augmentation type.
- The experiment used pre-trained models from the HuggingFace platform: bert-base-uncased, roberta-base, albert-base-v2, and distilbert-base-uncased.
- The following initial training parameters were used for fine-tuning models: batch size is 32, learning rate is  $5e^{-5}$ , weight decay is 0.01, and batch size is 16.
- For evaluation of the models F1 score was used with the seqeval library [20]
- All experiments were executed with Google Colab, a T4 GPU with High RAM runtime, or an Apple MacBook Pro M4 Max 40-core GPU.

## RESULTS AND DISCUSSION

It makes sense to split the research results into two blocks:

- Datasets preparation – preparation of the datasets with different types of augmentations depends on available resources, the size of the initial dataset, and the time required to build the dataset. Let us shed light on some key points here.
- Model evaluation – It is essential to understand the behavior of the models during fine-tuning, evaluate the models' effectiveness based on the F1 score, and compare the results of different augmentation methods and initial sub-dataset sizes.

As expected, the preparation of augmented datasets differs significantly in terms of time and implementation complexity. **Table 5** demonstrates the required time to build 100 augmented sentences for the CoNLL dataset. The time needed to generate augmented entities was evaluated over 10 runs using 100 randomly selected sentences from the original dataset. It should be noted that this measurement is somewhat approximate, as various factors significantly influence it. Since RND\_ENT utilizes only the training subset for inserting random entities, its execution is virtually instantaneous. The methods SUMM\_SENT, SUMM\_SDE, and SUMM\_GENT demonstrate execution speeds more than 10 times faster than methods relying on the OpenAI API. It is important to note that the

**Table 5. Approximate time in seconds to create augmentations for 100 sentences for the CoNLL dataset.**

Augmentations count Augmentation type	2/5/10	1	3	5
RND_ENT	1	--	--	--
OPENAI_SENT, OPENAI_SDE, OPENAI_GENT	--	3,750	9,500	16,150
SUMM_SENT, SUMM_SDE, SUMM_GENT	--	--	750	1,150



summarization-based methods were executed using local computational resources. In contrast, the generation of augmentations via OpenAI depends entirely on the external platform's performance and response time.

### Models Evaluation

In **Tables 6–9**, F1-score measurements for different models were demonstrated for all applied augmentation approaches and train sub-datasets. Additionally, for CoNLL and Ontonotes5 datasets, word-level augmentation results were added [14]. For each augmentation type, datasets were built with augmented sentence multipliers, as described in **Table 4**. In **Tables 6–9**, results were provided based on the mean and standard deviation for the augmentation type and dataset.

**Table 6. F1-scores for RoBERTa model evaluation and different CoNLL and Ontonotes5 datasets augmentation approaches.**

	S100	S20	S10	R10	S5	R5
<b>CoNLL Dataset</b>						
OPENAI_GENT		91.4±0.6	87.0±0.6	90.3±0.7	84.0±0.4	88.0±1.4
OPENAI_SDE		92.2±0.3	87.5±0.7	92.0±0.5	84.7±0.3	88.3±1.8
OPENAI_SENT		91.6±1.3	87.1±1.2	90.6±1.4	85.7±0.4	88.3±0.9
RND_ENT		92.8±0.2	87.9±0.2	92.5±0.5	86.0±0.3	90.0±2.1
SUMM_GENT		90.1±0.2	83.0±0.0	87.5±0.0	77.8±0.6	84.3±1.9
SUMM_SDE		90.2±1.0	82.3±0.3	88.6±1.1	76.4±1.4	82.6±3.3
SUMM_SENT		91.2±0.3	85.7±0.4	89.5±0.1	81.6±1.5	86.5±2.3
Word-level		92.7±0.3	88.6±0.3	92.0±0.4	86.1±0.8	89.4±0.5
Without aug.	95.77	91.61	87.46	90.22	82.92	83.16
<b>Ontonotes5 Dataset</b>						
OPENAI_GENT		80.3±1.1	77.7±1.2	82.0±1.4	77.5±1.1	79.9±1.1
OPENAI_SDE		80.4±1.2	78.9±1.3	82.9±0.8	79.6±0.9	82.0±0.4
OPENAI_SENT		78.0±3.3	76.2±2.6	80.2±3.6	76.7±2.1	77.7±3.2
RND_ENT		81.1±1.3	79.7±1.0	83.5±0.4	80.4±0.1	82.6±0.1
SUMM_GENT		81.0±0.3	79.0±0.4	83.5±0.4	78.2±0.1	80.0±0.0
SUMM_SDE		80.1±0.1	78.1±0.1	83.5±0.6	78.3±0.6	81.1±0.2
SUMM_SENT		82.6±0.1	80.6±0.2	84.1±0.6	80.1±0.6	81.5±0.8
Word-level		82.7±0.2	81.0±0.4	85.3±0.5	80.9±0.4	83.2±0.2
Without aug.	88.62	83.5	81.4	84.71	79.83	82.67



**Table 7. F1-scores for BERT model evaluation and different augmentation approaches for CoNLL and Ontonotes5 datasets.**

	S100	S20	S10	R10	S5	R5
<b>CoNLL Dataset</b>						
OPENAI_GENT		89.4±0.9	83.3±1.1	87.7±0.6	80.8±0.4	86.3±1.5
OPENAI_SDE		89.9±0.4	84.3±0.8	89.5±0.6	80.4±0.9	86.8±1.5
OPENAI_SENT		89.4±1.9	85.0±2.1	89.0±1.5	81.0±0.3	86.0±1.0
SUMM_GENT		87.5±0.5	80.8±0.0	84.2±1.2	73.8±0.1	80.6±1.2
SUMM_SDE		86.9±0.2	80.1±0.7	85.3±0.8	71.3±1.8	79.9±0.6
SUMM_SENT		88.0±0.3	81.2±0.7	86.0±0.6	74.4±0.1	81.3±1.2
RND_ENT		90.6±0.2	84.2±0.5	90.2±0.4	81.5±0.7	87.0±0.7
Word-level		90.7±0.3	85.2±0.7	90.0±0.2	82.3±1.0	87.5±0.7
Without aug.	94.4	90	85.5	88.9	77.2	77.6
<b>Ontonotes5 Dataset</b>						
OPENAI_GENT		75.8±1.2	73.3±1.4	77.3±1.8	73.6±0.8	75.1±0.8
OPENAI_SDE		76.1±1.8	74.1±1.3	78.9±1.1	74.8±0.4	77.3±0.7
OPENAI_SENT		72.6±3.8	71.1±2.9	75.9±3.8	71.8±2.5	73.1±2.6
SUMM_GENT		76.7±0.1	74.0±0.0	77.8±0.3	72.9±0.3	74.9±0.3
SUMM_SDE		74.6±0.6	72.7±0.1	79.1±0.7	73.4±0.1	75.9±0.0
SUMM_SENT		76.6±0.3	75.1±0.5	79.3±1.0	74.2±0.0	76.6±0.5
RND_ENT		77.2±0.6	76.1±0.8	79.7±0.4	76.6±0.1	78.3±0.6
Word-level		78.3±0.4	76.1±0.8	81.5±0.3	75.9±0.7	79.0±0.4
Without aug.	85.4	80	76.8	81.4	75	76.8

**Table 8. F1-scores for ALBERT model evaluation and different augmentation approaches for CoNLL and Ontonotes5 datasets.**

	S100	S20	S10	R10	S5	R5
<b>CoNLL Dataset</b>						
OPENAI_GENT		87.0±1.2	82.7±0.5	86.3±1.4	80.8±0.4	83.8±0.8
OPENAI_SDE		88.0±1.1	82.5±0.4	88.4±0.2	80.9±1.0	84.6±1.6
OPENAI_SENT		88.2±2.1	83.6±2.6	87.2±1.6	81.1±1.8	83.5±0.9
SUMM_GENT		86.3±0.4	78.7±1.0	82.7±0.6	76.2±0.2	79.6±0.2
SUMM_SDE		85.2±0.3	78.6±0.1	83.7±0.1	74.0±0.1	80.0±1.0
SUMM_SENT		87.2±0.9	82.2±0.1	85.6±0.3	77.6±0.1	81.5±0.2

	S100	S20	S10	R10	S5	R5
RND_ENT		88.1±0.4	83.7±1.4	88.4±0.1	81.1±0.6	84.6±0.2
Word-level		88.6±0.7	83.3±0.6	88.6±0.7	82.6±0.6	85.1±0.9
Without aug.	93.4	89.7	85.7	87.5	81.9	83.9
<b>Ontonotes5 Dataset</b>						
OPENAI_GENT		72.8±2.0	71.5±1.7	74.8±1.6	71.6±2.5	73.0±1.0
OPENAI_SDE		73.4±2.2	71.5±2.0	76.5±1.2	72.8±1.9	75.4±0.8
OPENAI_SENT		71.9±3.8	70.4±2.8	73.7±3.8	70.0±3.1	71.7±3.1
SUMM_GENT		75.1±0.6	73.0±0.3	76.8±0.2	68.5±5.4	73.9±0.6
SUMM_SDE		73.8±0.2	71.5±0.9	77.9±0.7	70.9±1.0	74.2±0.7
SUMM_SENT		76.4±0.4	74.7±0.1	78.8±0.2	74.3±0.1	75.8±0.1
RND_ENT		74.4±1.8	72.0±2.3	77.7±1.3	73.5±1.9	76.8±1.0
Word-level		75.2±0.5	72.9±0.8	79.2±0.3	73.8±0.5	76.7±1.2
Without aug.	84.5	78.2	76.1	80	75.5	76.5

**Table 9. F1-scores for DistilBERT model evaluation and different augmentation approaches for CoNLL and Ontonotes5 datasets.**

	S100	S20	S10	R10	S5	R5
<b>CoNLL Dataset</b>						
OPENAI_GENT		88.2±0.2	81.8±0.3	86.2±0.3	78.2±2.7	83.7±1.9
OPENAI_SDE		89.2±0.1	83.4±0.2	88.3±0.8	78.5±2.6	84.3±1.6
OPENAI_SENT		88.8±1.2	83.1±1.7	87.8±0.7	79.9±1.7	83.3±1.3
SUMM_GENT		85.2±0.7	76.1±0.9	80.7±0.7	70.1±1.7	75.2±2.6
SUMM_SDE		84.5±0.7	76.7±0.3	81.8±0.0	68.3±1.2	75.5±2.0
SUMM_SENT		85.5±0.2	78.2±0.6	83.6±0.7	73.1±0.4	78.3±0.7
RND_ENT		89.7±0.4	84.0±0.4	89.0±0.5	81.5±0.7	85.0±1.3
Word-level		89.7±0.3	84.2±0.6	89.1±0.3	81.4±0.7	85.9±0.3
Without aug.	94.2	89.9	83.9	87.7	73.3	77.7
<b>Ontonotes5 Dataset</b>						
OPENAI_GENT		74.1±1.3	71.7±1.6	75.9±1.6	71.1±0.2	72.9±0.6
OPENAI_SDE		74.6±1.5	72.3±1.2	77.7±0.6	72.7±0.3	75.6±0.5
OPENAI_SENT		71.9±3.8	70.3±2.3	74.8±3.2	70.4±1.9	72.0±2.2
SUMM_GENT		74.6±0.6	71.0±0.5	75.6±0.1	69.8±0.3	71.2±0.5

	S100	S20	S10	R10	S5	R5
SUMM_SDE		72.9±0.7	69.7±1.3	76.9±0.4	70.0±0.5	72.9±0.0
SUMM_SENT		76.4±0.4	73.0±0.1	77.9±0.3	72.1±0.3	74.9±0.4
RND_ENT		75.8±0.6	74.1±1.2	78.7±0.7	74.4±0.2	76.9±0.3
Word-level		77.3±0.6	75.2±0.6	80.4±0.3	74.4±0.5	77.6±0.2
Without aug.	85.3	78.4	76	79.6	74.1	75.3

Based on **Tables 6–9**, the following key points can be highlighted:

- The performance of the models on the S100 dataset exceeds the best result of all sub-datasets, even with augmentations, by 3–4%. This is an expected outcome, as the full dataset contains more diverse named entities in various contexts.
- For S20, S10, R10 patterns, results for the CoNLL dataset demonstrated a positive impact for some augmentation methods and a negative impact for others. In most cases, word-level augmentations and augmentations based on random entity replacement show minor improvement in 1–3% for RoBERTa and BERT models. At the same time, ALBERT and DistilBERT even have a negative impact. All models, except RoBERTa, show an adverse effect of the augmentation on dataset quality and the received results.
- For the Ontonotes5 dataset, all models demonstrated a negative impact on the performance of the models. The Ontonotes5 dataset is bigger than CoNLL, has more distributed entities by dataset, and has 18 categories. Augmented data with the same entities or injected out-of-context entities improves some popular categories but negatively impacts the rest.
- For the S5 pattern on the CoNLL dataset, most methods demonstrated performance improvement up to 5%. It is not the expected result that augmentation methods, based on summarization, demonstrated negative impact and performance degradation up to 5–7% in some cases. From the group of those methods, summarization with random entity injection demonstrates better results with negligible performance degradation. Also, the ALBERT model hurts this dataset pattern in all cases, except word-level.
- For the R5 pattern on the CoNLL dataset, most of the augmentations demonstrated a significant increase in performance up to 7–8% in almost all cases, except the summarization group of methods. The summarization group of methods demonstrated minor improvement for RoBERTa and BERT models and performance degradation for the rest of the models. Some of the augmentation methods achieved performance results on the level of S10, R10, S20 datasets, or close to it.
- It is interesting that the context in which the named entity is used has a bigger impact on the model's possibilities to recognize some entity than the named entity itself. Sentences rephrasing using the same entities in the original sentence with OpenAI demonstrated better results in almost all experiments than summarization methods with entity injection. Moreover, for both approaches, entity replacement without random ones from the dataset or OpenAI-generated ones doesn't make any performance improvements and has an adverse effect.
- It is not an expected finding that summarization and OpenAI-based methods do not provide performance improvement on entity recognition compared with simpler ones, like word-level.

## CONCLUSION

This article investigates the impact on the effectiveness of sentence-level data augmentation models, the generation of augmented data using LLMs using OpenAI models, and the substitution of entities in sentences with random ones from a set of entities. To investigate how the size of the initial dataset affects the efficiency of augmentation, the CoNLL 2003 and Ontonotes5 datasets were used, and six different approaches of the initial dataset splittings were used for each of them: a complete dataset for reference comparison, 20% of the initial training dataset, two split approaches by 10% and two 5% partitions. For the 10% and 5% datasets, two different methods were used: all sentences from the initial dataset in a row, as this allows taking related sentences into one dataset, and sentences taken randomly. The study used the RoBERTa, DistilBERT, ALBERT, and BERT models, with the estimate based on the F1 score.

Creating additional context for named entities using summarization or paraphrasing techniques of the initial sentences, while preserving the initial named entities, should increase the efficiency of the models. According to the results obtained, in most cases, word-level augmentations showed better results or were on a par with sentence-level methods. It is worth noting that augmentations based on the substitution of entities in the initial sentences showed results similar to methods at the word level. However, in most cases, sentence-level methods negatively impacted model performance more than a reference result without augmentations for a comparable data set. A significant increase in the context of applying specific named entities leads to an oversaturation of the initial dataset with certain entities but reduces the model's generalization ability.

Most methods showed a significant improvement in results for the CoNLL dataset when 5% of the initial dataset was split. Almost all augmentation methods showed an increase of 5-9%, for the RoBERTa model using augmentations based on random entity substitution, 92.26% was achieved compared to a non-augmentation option of 83.16%. Nevertheless, the results of using augmentations for the Ontonotes5 dataset did not improve the result, and in many cases, worsened it. Due to the larger size of the initial dataset, the relatively small number of named entities in the text, and the large number of categories, augmentation is inefficient. It leads to a decrease in data quality for the Ontonotes5 dataset.

To summarize further research directions, context generation looks promising to create synthetic data, but requires more careful planning. Named entities distribution and initial dataset quality could significantly impact the result. Using LLMs as a supporting tool for training dataset generation for smaller models could provide excellent results, especially with fine-tuning models like LLaMa and Mistral for domain analysis, existing dataset analysis, and extending the weakest parts.

## ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, authorship, and/or publication of this article.

## COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

## AUTHOR CONTRIBUTIONS

Conceptualization, [I.D.]; methodology, [I.D.]; validation, [B.P., I.D.]; formal analysis, [I.D.]; investigation, [I.D.]; resources, [I.D.]; data curation, [I.D.]; writing – original draft preparation, [I.D.]; writing – review and editing, [B.P.]; visualization, [I.D.].

All authors have read and agreed to the published version of the manuscript.

## REFERENCES

- [1] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *Proceedings of NAACL-HLT 2016*, 260–270. <https://doi.org/10.18653/v1/N16-1030>
- [2] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-1423>
- [3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- [4] Yamada, I., Asai, A., Shindo, H., Takeda, H., & Matsumoto, Y. (2020). LUKE: Deep contextualized entity representations with entity-aware self-attention. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 6442–6454. <https://doi.org/10.18653/v1/2020.emnlp-main.523>
- [5] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*. <https://doi.org/10.48550/arXiv.1907.11692>
- [6] Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33 (2020): 1877–1901.
- [7] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*. <https://doi.org/10.48550/arXiv.2302.13971>
- [8] Chen, J., Tam, D., Raffel, C., Bansal, M., & Yang, D. (2023). An empirical survey of data augmentation for limited data learning in NLP. *Transactions of the Association for Computational Linguistics*, 11, 191–211. [https://doi.org/10.1162/tacl\\_a\\_00542](https://doi.org/10.1162/tacl_a_00542)
- [9] Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.findings-acl.84>
- [10] Chen, S., Aguilar, G., Neves, L., & Solorio, T. (2021). Data augmentation for cross-domain named entity recognition. *arXiv preprint arXiv:2109.01758*. <https://arxiv.org/abs/2109.01758>
- [11] Dai, X., & Adel, H. (2020). An Analysis of Simple Data Augmentation for Named Entity Recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, 3861–3867. International Committee on Computational Linguistics. <https://doi.org/10.18653/v1/2020.coling-main.343>
- [12] Pavlyshenko, B., & Stasiuk, M. (2023). Augmentation in a binary text classification task. In *2023 IEEE 13th International Conference on Electronics and Information Technologies (ELIT)* (pp. 177–180). IEEE. <https://doi.org/10.1109/ELIT57602.2023.10151742>
- [13] Pavlyshenko, B., & Stasiuk, M. (2024). Data augmentation in text classification with multiple categories. *Electronics and Information Technologies*, 25, 67–80. <http://dx.doi.org/10.30970/eli.25.6>
- [14] Pavlyshenko, B., & Drozdov, I. (2024). Influence of data augmentation on named entity recognition using transformer-based models. *Electronics and Information Technologies*, 28, 61–72. <http://dx.doi.org/10.30970/eli.28.6>

- [15] Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*. <https://arxiv.org/abs/cs/0306050>
  - [16] Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers* (pp. 57–60). <https://aclanthology.org/N06-2015>
  - [17] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1–67. <http://jmlr.org/papers/v21/20-074.html>
  - [18] HuggingFace. (n.d.). HuggingFace [Computer software]. Retrieved June 2025, from <https://huggingface.co/>
  - [19] OpenAI. (n.d.). OpenAI API [Computer software]. Retrieved June 2025, from <https://platform.openai.com/docs/api-reference>
  - [20] Seqeval library repository. (n.d.). Retrieved June 2025, from <https://github.com/chakki-works/seqeval>
- 

## РОЗПІЗНАВАННЯ ІМЕНОВАНИХ СУТНОСТЕЙ ІЗ ВИКОРИСТАННЯМ ГЕНЕРАТИВНИХ ТРАНСФОРМЕРНИХ МОДЕЛЕЙ З АУГМЕНТАЦІЄЮ ДАНИХ НА РІВНІ РЕЧЕНЬ

**Ігор Дроздов, Богдан Павлишенко**

*Кафедра системного проектування,  
Львівський національний університет імені Івана Франка,  
вул. Драгоманова, 50, м. Львів, 79005, Україна*

### АНОТАЦІЯ

**Вступ.** Розпізнавання іменованих сутностей як одне з ключових завдань галузі обробки природної мови відіграє важливу роль в опрацюванні та розумінні текстів. Використання моделей на основі трансформерів демонструє виняткову продуктивність у більшості задач в області опрацювання природної мови, але вимагає значного обсягу інформації для практичного навчання моделі. Створення високоякісного анотованого набору даних для розпізнавання іменованих сутностей потребує значних ресурсів, особливо для малопоширених мов. Використання аугментації даних для розширення анотованого набору даних синтетичними дає змогу підвищити ефективність моделей для розпізнавання іменованих сутностей. Метою цього дослідження є використання аугментації на рівні речень та великих мовних моделей для підвищення продуктивності моделей на малих наборах даних.

**Матеріали та методи.** Для дослідження впливу аугментації даних було взято 5%, 10% і 20% відсотків тренувальних даних із датасетів CoNLL та Ontonotes5 з різними характеристиками. Для побудови наборів аугментованих даних було використано три основні підходи: узагальнення речень за допомогою моделі T5 із подальшою вставкою іменованих сутностей, перефразування речень за допомогою OpenAI Api, а також кілька методів заміни іменованих сутностей у початкових і синтетичних реченнях. Для оцінювання використовували моделі BERT, ALBERT, DistilBERT і RoBERTa.

**Результати.** Ефективність використання різних методів аугментації суттєво залежить від початкового набору даних і його якості. Для невеликих наборів даних із невеликою кількістю категорій для розпізнавання використання методу аугментації на рівні речень за допомогою узагальнення або перефразування дають підвищення

ефективності моделей до 10%. З іншого боку, у випадку штучно створених даних збільшення обсягу набору даних може призвести до погіршення результатів розпізнавання.

**Висновки.** Використання аугментації даних для розпізнавання іменованих сутностей є ефективним інструментом для невеликих наборів даних і може підвищити продуктивність моделі у випадках з обмеженими ресурсами. Однак синтетичні дані не можуть повністю замінити більший, краще побудований вихідний набір даних за допомогою розширення контексту для існуючих іменованих сутностей і генерування нових синтетичних сутностей.

**Ключові слова:** розпізнавання іменованих сутностей, обробка природної мови, аугментація даних, великі мовні моделі



UDC 004.4`2:004.054

## STOCHASTIC ANALYSIS OF CPU SCHEDULING IN APPLE M-SERIES

Bohdan Mikh , Yuriy Korchak\* 

Ivan Franko National University of Lviv  
107 Tarnavsky St., Lviv 79017, Ukraine

Mikh, B., Korchak, Yu. (2025). Stochastic Analysis of CPU Scheduling in Apple M-Series. *Electronics and Information Technologies*, 31, 19–30. <https://doi.org/10.30970/eli.31.2>

### ABSTRACT

**Background.** Efficient task scheduling in heterogeneous CPU architectures is critical for maintaining system responsiveness and optimal resource utilization under fluctuating workloads. Apple M-Series processors, based on ARM architecture, integrate high-performance (P-cores) and energy-efficient (E-cores), allowing adaptive distribution of tasks depending on their computational complexity and latency sensitivity. This architectural design presents new challenges and opportunities for analyzing task dispatching mechanisms at the operating system level, particularly within macOS scheduling subsystems.

**Materials and Methods.** The study employed low-level telemetry data collected from macOS-based systems operating under high-load production-like CI/CD scenarios, simulating real-world parallel task execution. The collected data sets were analyzed using stochastic time series modeling, construction of confidence intervals, approximation of waiting times with exponential and log-normal distributions, autocorrelation function analysis, Pearson correlation metrics, and evaluation of context switching frequency across multiple QoS (Quality of Service) classes.

**Results and Discussion.** The analysis revealed a clear architectural specialization between core types. P-cores demonstrated consistently higher processing intensity, reduced queuing delays, and superior responsiveness for delay-sensitive tasks, whereas E-cores ensured stable handling of background workloads. Statistical modeling identified a significant inverse correlation between P-core utilization share and overall task latency, confirming that increasing P-core allocation directly improves execution time for critical workloads. Derived autocorrelation and distribution parameters allow the formulation of quantitative models describing resource allocation behavior in Apple Silicon's heterogeneous environment.

**Conclusion.** The obtained results provide a statistically grounded basis for improving task dispatching strategies in macOS on Apple Silicon platforms. The findings contribute to better latency predictability, efficient resource balancing, and a deeper understanding of kernel-level scheduling dynamics under highly parallelized workload scenarios.

**Keywords:** macOS, Apple Silicon, CPU scheduling, stochastic modeling, time series analysis, ARM architecture

### INTRODUCTION

In the modern era of heterogeneous architectures, having a powerful processor does not guarantee the optimal distribution of threads and tasks between cores; this is handled by context switching. Apple Silicon M-Series introduced two types of cores: high-performance P-cores and energy-efficient E-cores. Unlike the traditional big-LITTLE (standard ARM architecture, where each core is similar to the other, achieving



© 2025 Bohdan Mikh & Yuriy Korchak. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and information technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.



heterogeneity in the architecture), Apple has patented its model of load distribution and power management between core clusters [1].

To study the behavior of the XNU scheduler, detailed telemetry was collected, including interval CPU metrics for P-cores and E-cores via Instruments (Time Profiler), taskpolicy logs to record Quality of Service (QoS) classes and dynamic priority changes, and vmstat data on context switching and queue lengths.

The article presents an empirical stochastic analysis: CPU load is modelled as a floating time series with 95% confidence intervals, queueing time is approximated by exponential and lognormal distributions, context switching frequency is measured for each core type, and the correlation between the proportion of time on P-cores and thread latency is quantified.

## MATERIALS AND METHODS

### M-Series architecture

Apple M-Series SoCs integrate two classes of CPU cores with distinct roles and characteristics on a single chip (Fig. 1). P-cores (Firestorm) have extended instruction sets and high clock speeds, ensuring minimal latency when performing single or latency-sensitive tasks. E-cores (Icestorm) are the opposite in design: they are simpler, operate in the 1-2 GHz range, and consume much less energy per clock cycle [2].

This allows Apple to achieve high energy efficiency and reduce heat dissipation in cases where there is no need to keep all cores at maximum frequencies. Both types of cores operate through a single system-level cache (SLC) subsystem [3], which acts as a buffer between the cores and memory controllers. This architecture reduces the latency of access to shared data and allows P- and E-cores to allocate resources (time, power, cache) more efficiently [4].

When load is increased or thermal/power constraints are reached, the system can automatically reduce the clock speed of the P-cores or transfer some of the threads to the E-cores using dynamic throttling.

High-priority tasks with low latency are maintained on the P-cores. At the same time, background or less critical processes migrate to the E-cores, striking a balance between performance and energy efficiency.

Only an understanding of the physical topology of P- and E-cores, their cache subsystems, and data buses allows for the correct interpretation of load graphs, context

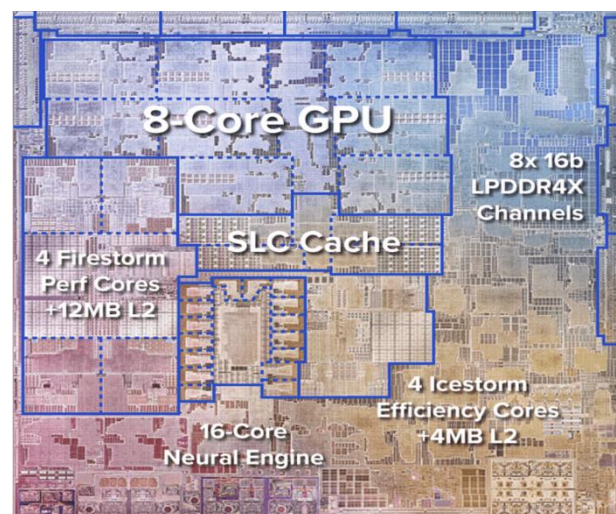


Fig. 1. Floor plan of Apple M1 SoC with highlighted P-cores and E-cores [2].

switch frequencies, and task wait times, as well as the formulation of recommendations for optimal scheduling policies under various load scenarios.

### Context switching

Context switching is the process of removing a thread from execution and loading another thread (Fig. 2). In the XNU scheduler [5], the frequency and cost of these switches depend on quantization: with Round-Robin, each thread receives a small amount of time (usually 1-4 ms), after which the execution queue is checked, and a context switch may occur. In the XNU scheduler (MacOS kernel), thread dispatching is based on a precise classification by quality of service (QoS) classes, which affects the order and frequency of context switching.

There are four main QoS levels [6]:

- User Interactive – tasks with the highest priority (UI, animation, user input);
- User Initiated – long-running but critical user operations (opening a document, compiling);
- Utility – medium-importance background services (indexing, caching);
- Background – low-priority or batch calculations (system updates, synchronization).

For each QoS level, it maintains its queue of tasks (processes) on each logical processor, which prevents thread blocking.

When time becomes available to execute a task (also sometimes referred to as a 'slot'), the task scheduler checks the queues from highest to lowest priority and selects the first available thread. Threads with higher QoS can interrupt lower-priority threads without waiting for their quantum to complete (preemptive scheduling). The XNU scheduler, like Apple's task scheduler, attempts to resume the thread on the same physical core, but under heavy load, it redirects threads between P-cores and E-cores. When thermal or energy

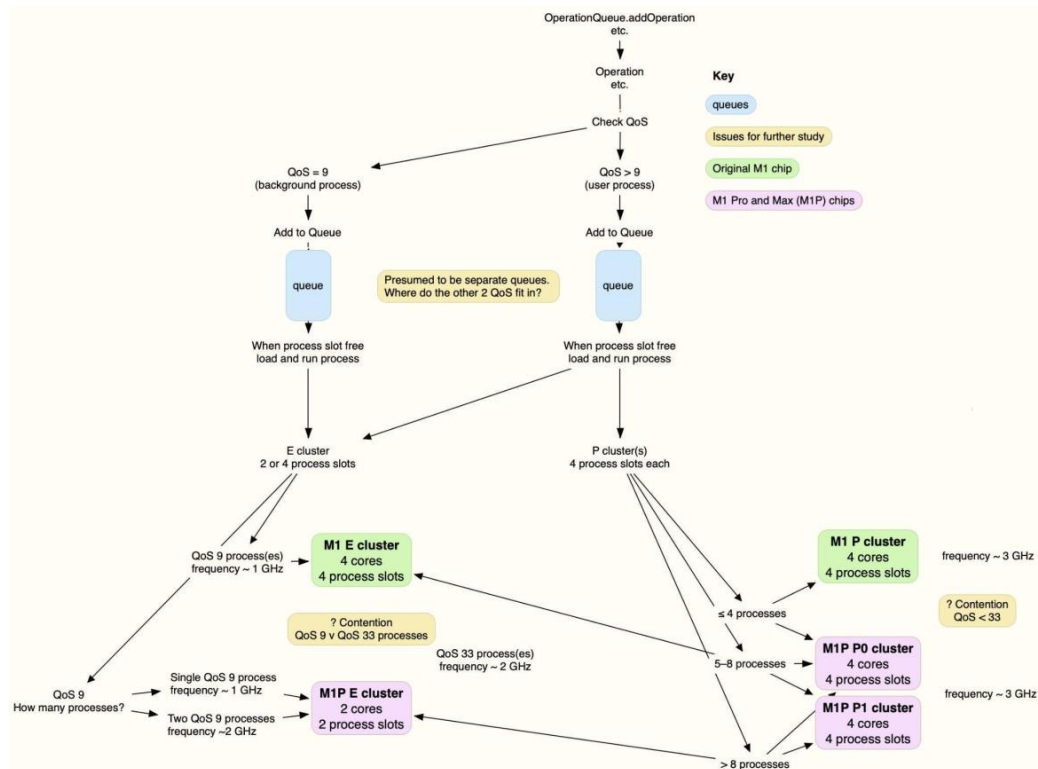


Fig. 2. Load distribution and task scheduling scheme on M-series processors [7].

limits are reached, the scheduler reduces clock frequencies or moves contexts from P-cores to E-cores, which also affects the context switching frequency. As a result, the system simultaneously provides low latency for essential streams and a stable throughput channel for background tasks.

### Data collection

The study utilized data collected directly from a real, highly loaded CI/CD environment in real time. The active user-generated load requests are running on a deployed Apple Mac mini environment with M1/M2 and M4 series processors (from the 2024 series).

Standard Mac OS operating system tools were used to collect data, which allowed for sufficiently accurate recording of low-level dispatcher performance characteristics. Time Profiler (via Instruments) [8] was used with a resolution of 1 ms to record the load of each core and its type (P or E). This made it possible to build a time series of CPU utilization separately for each core class. At the same time, task policy was used to collect information about the QoS classes assigned to threads and the dynamic changes in these classes during execution.

Additionally, vmstat provided aggregated information about the number of context switches, the length of the ready queue, and the idle time at 500 ms intervals. The waiting time in the queue was recorded separately: for each step of the CI/CD process, the time between its placement in the scheduler and the actual start of execution was recorded. Thus, the data obtained from the real environment made it possible to study the peculiarities of the XNU scheduler on Apple Silicon without any artificial load or synthetic tests, which in turn should help to understand the system's real operational capacity and separate it from 'artificial' analysis.

## RESULTS AND DISCUSSION

To visually confirm the nature of the load distribution between cores during live operation, CPU History fragments were recorded from the macOS system monitor during the active computation phase. It should be emphasized that these screenshots show relative per-core activity rather than absolute percentages of utilization; the corresponding absolute values are provided separately in Fig. 3. Fig. 4 presents usage histograms for each of the 12 logical cores (6 E-cores and 6 P-cores). E-cores (Cores 1–6) demonstrate denser average activity, reflecting background and I/O-bound processes, which the XNU dispatcher assigns to less productive cores according to QoS priorities. P-cores (Cores 7–12) exhibit shorter but more pronounced bursts, typical for latency-sensitive jobs requiring elevated QoS. This behavior is consistent with the earlier ACF and correlation analysis ( $r \approx -0.74$  between P-core utilization and task latency), as well as the rhythmic activity patterns observed in the autocorrelation function. Fig. 5 illustrates a second fragment captured during a sustained high CI/CD workload. Unlike synthetic stress tests that produce uniform 100% utilization across all cores, real workloads generate high but heterogeneous load characterized by bursts and idle intervals. Green areas correspond to user processes, while red segments represent system threads.

Taken together, this confirms the hypothesis stated in the study about a significant difference in the behavior of different types of cores, depending on QoS, scheduler class, and load type [9].

Fig. 3 illustrates the time distribution of processor usage percentages across different types of cores. The blue line corresponds to the load of productive cores (P-cores), while the orange line corresponds to energy-efficient cores (E-cores). The horizontal axis represents the conditional time (in units of discrete measurements), while the vertical axis represents the load level in percentage.

As can be seen from the graph (Fig. 3), P-cores systematically demonstrate higher-than-average utilization (utilization per time) during most periods, remaining in the range of 60–85%. Meanwhile, E-cores operate in a more scattered mode, with an average load of

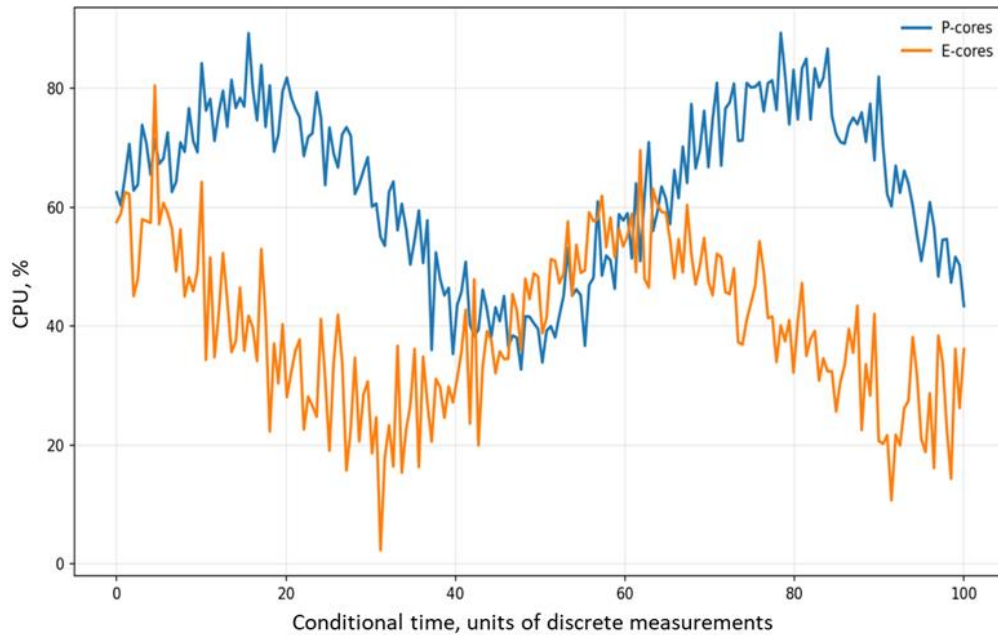


Fig. 3. M-series processor load on P- and E-cores.

30–55%. Periods of decline on P-cores (for example, between  $t \approx 35$  and  $t \approx 50$ ) are often accompanied by slight increases on E-cores, indicating a partial shift in load during phase changes and load type for computation.

This confirms the assertion that the XNU dispatcher, adhering to the priority scheduling model, assigns compute-bound tasks with high quality of service (QoS) to the most productive cores. The behavior corresponds to the architectural [10] role of the cores: P-cores are for latency-sensitive threads, while E-cores are for background or lower-priority threads.

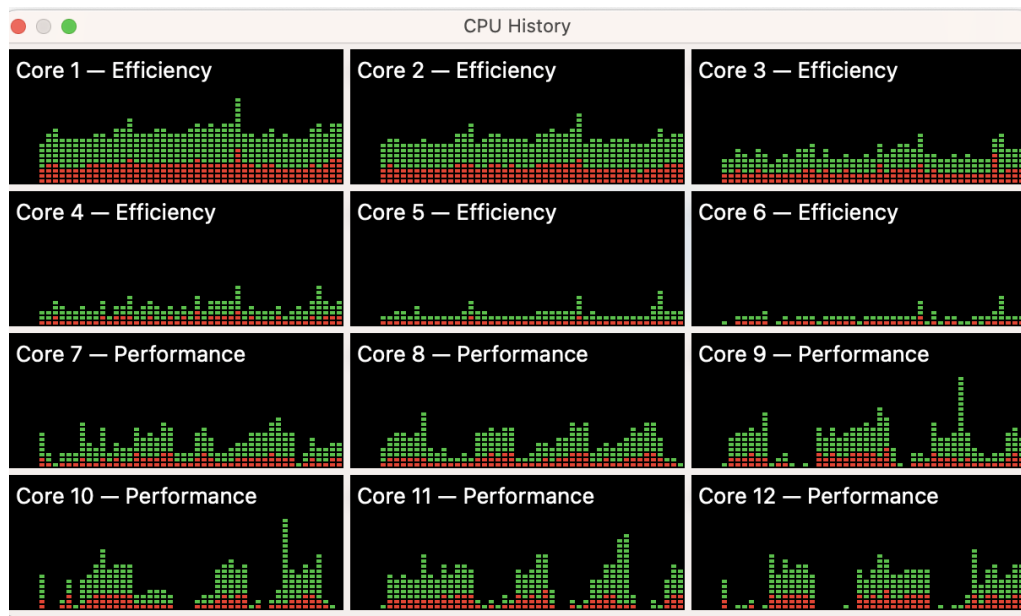


Fig. 4. Histograms of CPU resource usage by P- and E-cores in a scaled state.

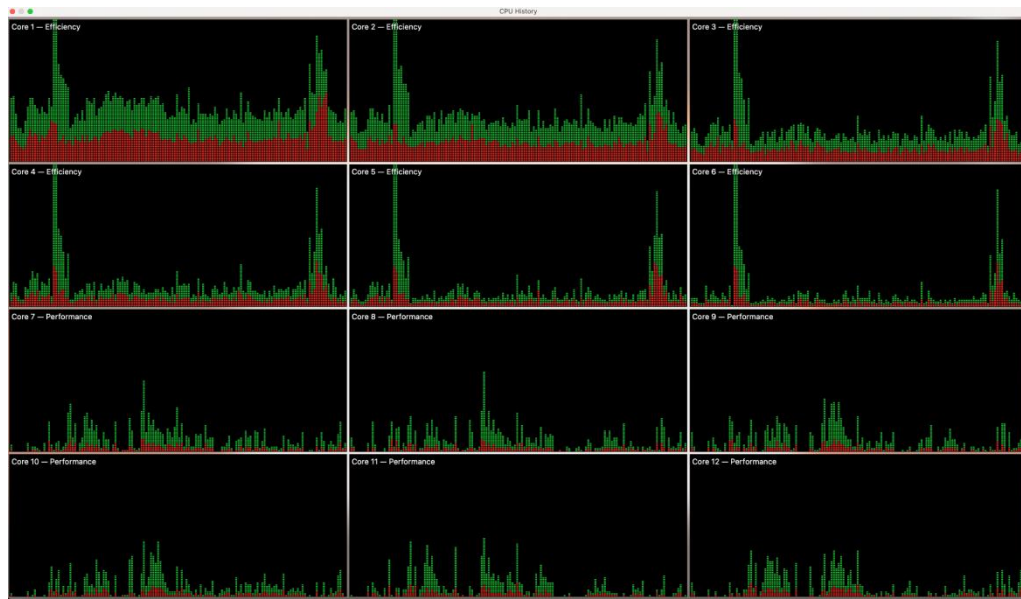


Fig. 5. Comparison of P- and E-cores activity under stress load.

Fig. 6 shows the empirical distribution of thread waiting times in the queue until the first dispatching, constructed separately for productive (P) and energy-efficient (E) cores.

Based on the data, density histograms were constructed, and exponential approximation curves were superimposed for each subsampled class using the following formula:

$$f(t) = \lambda e^{-\lambda t}, \quad (1)$$

where  $\lambda$  is the parameter inverse to the average waiting time,  $t$  is time.

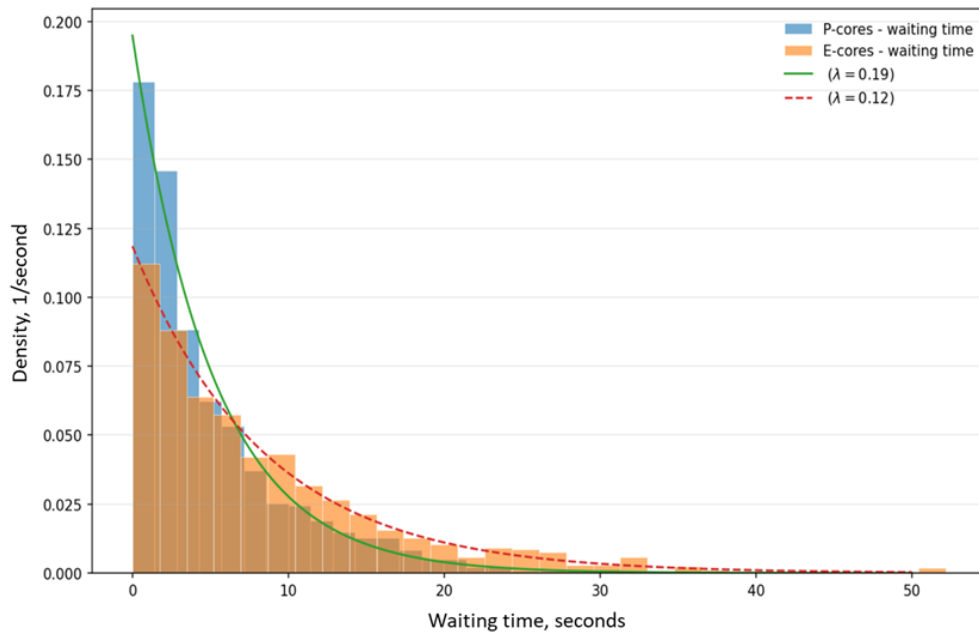


Fig. 6. Empirical distribution of the waiting time of flows in the queue until the moment of launch with exponential superposition.



The blue histogram represents the waiting time of flows that were eventually dispatched to P-cores, and the orange histogram represents the waiting time of flows that were sent to E-cores.

The green line ( $\lambda = 0.20$ ) shows the exponential approximation for P-cores, and the red dotted line ( $\lambda = 0.13$ ) shows the exponential approximation for E-cores.

The value of  $\lambda$  indicates a significantly higher service speed on P-clusters. The distributions of both classes demonstrate the 'fading' property; most flows are executed with a slight delay, but there is a long 'tail' (heavy tail), especially on E-cores [11].

This is typical for heterogeneous systems, where background tasks can accumulate as the priority load increases. These results are consistent with the XNU QoS scheduler policy: the most delay-sensitive threads are assigned to P-cores. At the same time, E-cores can accumulate a queue of lower-priority tasks, leading to an overall increase in latency.

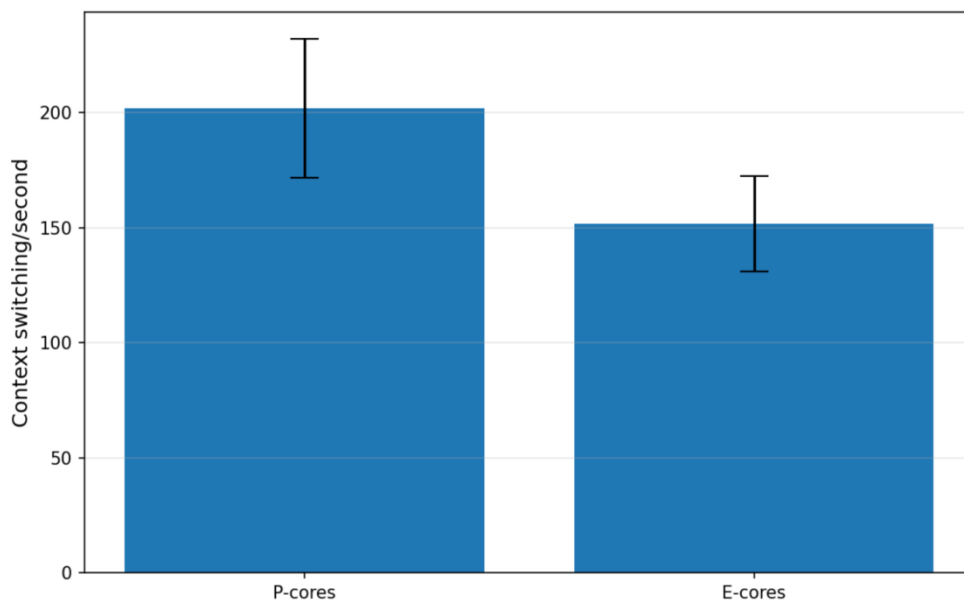
**Fig. 7** shows the average context switch rate (context switches per second) for threads executed on P- and E-cores.

The bars represent the average value, and the black lines represent the standard deviation ( $\pm 1\sigma$ ) calculated based on 500 ms samples from vmstat. As shown in the diagram, P-cores exhibit a significantly higher switch rate, averaging over 200 context switches per second.

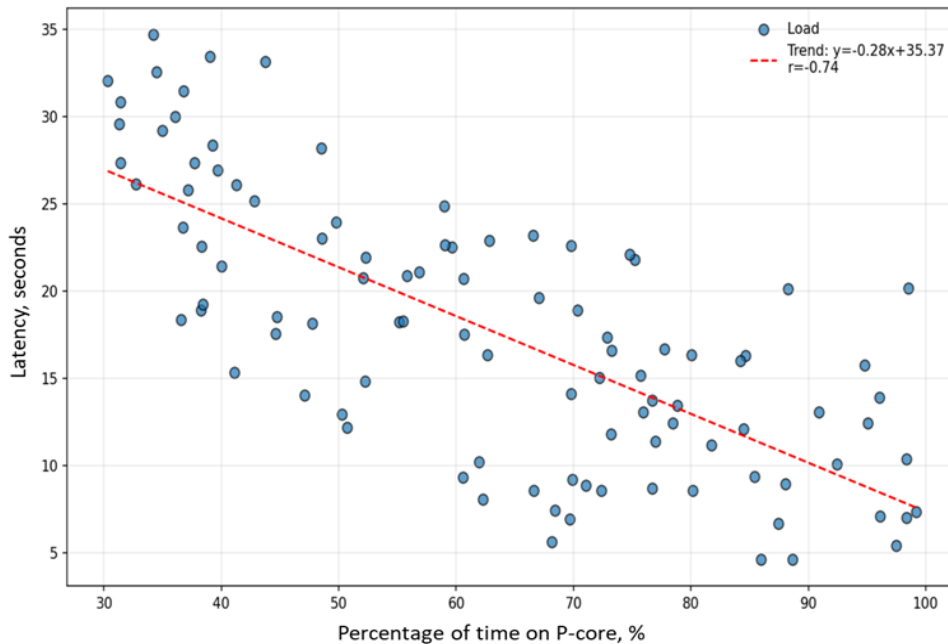
This correlates with the intensive servicing of short-lived threads that require frequent reallocation of CPU resources. Meanwhile, E-cores, which mainly serve background or less critical tasks, have a lower and more stable frequency, approximately 150 switches per second.

The higher level of context switching on P-cores can also be explained by a more aggressive policy of preempting flows in high-priority QoS classes. This dynamism suggests that XNU schedules flow on productive cores to minimize latency. At the same time, higher switching volumes can lead to additional overhead if not controlled by appropriate throttling mechanisms or time quantum optimization.

In **Fig. 8**, each point on the graph corresponds to a separate task in processor time, where the X-axis plots the percentage of time on P-cores and the Y-axis plots the total latency in seconds from entering the queue to completion.



**Fig. 7.** Average number of context switches per second for threads on P- and E-cores.



**Fig. 8.** Correlation between the percentage of time spent on P-cores and task execution delay.

There is a clear negative correlation between these two metrics, which is emphasized by the trend line (dotted red) with the equation  $y = -0.28x + 35.37$ . The calculated Pearson coefficient  $r = -0.74$  confirms a strong inverse relationship: the more time the stream spends on productive cores, the lower its total delay.

This relationship highlights the crucial role of productive cores in determining the scheduler's time characteristics under high computational load intensity. Such a correlation demonstrates the potential for adaptive transfer of critical tasks to P-clusters, thereby minimizing execution latency. The presence of a stable negative gradient also indicates the limited compensatory capabilities of E-cores as the complexity of task processing increases.

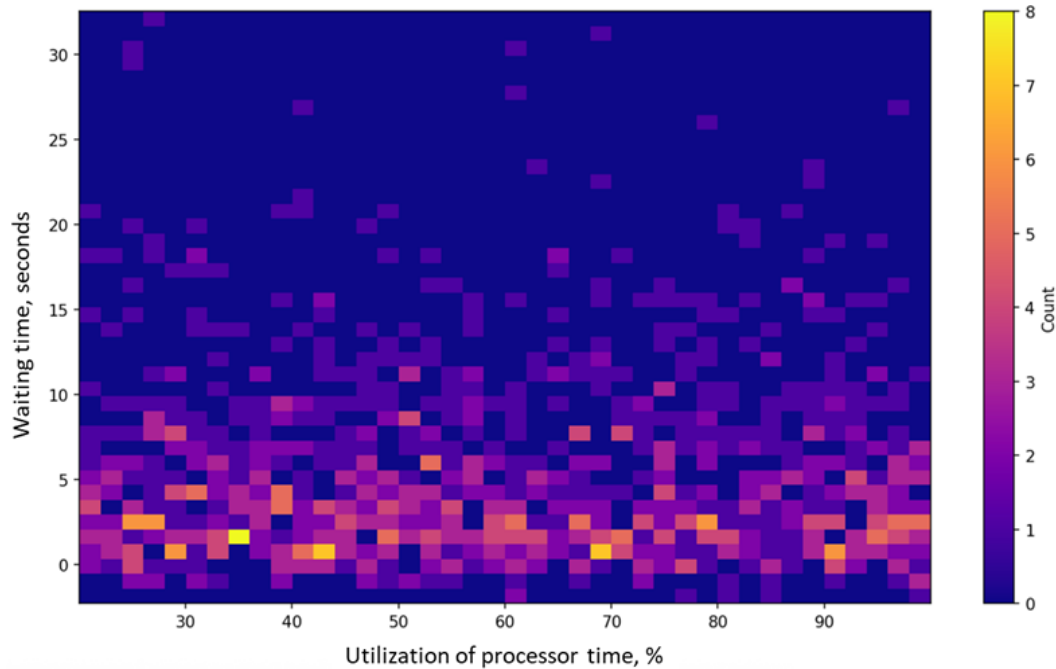
This indicates the practical advantage of assigning critical threads to P-cores in terms of both faster instruction processing and dispatching within XNU [12]. This approach is particularly relevant for tasks that are sensitive to pipeline delays (e.g., compilation or testing), where even a few seconds of delay can be critical in a large task queue.

The analysis also confirms that the amount of time spent by the flow on P-cores is a key predictor of task execution delays. This highlights the importance of considering processor architecture features when developing task scheduling policies to optimize response times in highly parallel systems.

Given the patterns observed, it is advisable to further detail the interaction between QoS classes, load types, and flow distribution dynamics in heterogeneous environments. Such research can provide additional practical guidelines for improving scheduling strategies, taking into account the architectural specifics of modern multi-core processors.

In **Fig. 9**, the total processor time usage (in percent) is plotted on the X-axis, and the task waiting time in the execution queue is plotted on the Y-axis. The color intensity corresponds to the number of observations in the corresponding bin (cluster). The image allows you to intuitively identify the 'hot spots' of the planning load.

The highest concentration of events (shades of yellow and pink) is observed in the range from 35% to 75% CPU load, and with a waiting time of 0 to 8 seconds. This indicates a typical area of activity for streams under real CI/CD load, where most of the job steps pass through the scheduler. At the same time, in areas with extremely low load (less than



**Fig. 9.** Correlation between total CPU load (%) and task waiting time (seconds).

30%) or, conversely, with high CPU usage (90%+), there is a noticeable increase in the variability of waiting time, which may be the result of either insufficient thread saturation (in the first case) or existing throttling (i.e., temporary, forced limitation of processor frequency or system resources to avoid overheating or exceeding energy limits), reduction of quanta, or excessive competition for schedule slots (in the second case).

The heat map provides empirical confirmation that even under high load, the XNU system maintains overall scheduling stability; however, exceeding the optimal zone (by up to ~70%) is accompanied by an increase in delay dispersion.

To quantitatively describe the patterns in load and performance, several aggregate indicators were utilized, including the autocorrelation function (ACF), Pearson's correlation coefficient, exponential distribution parameters, and the frequency of throttling events.

To assess the dependencies between the values of a random variable in a time series, the autocorrelation function (ACF) was used, which is calculated taking into account the shift to a certain delay  $k$ :

$$\rho(k) = \frac{Cov(X_t, X_{t-k})}{\sigma^2}, \quad (2)$$

where  $X_t$  – the value of the time series at a given point in time  $t$ ,  $X_{t-k}$  – values of the same series, but  $k$  time units previously,  $Cov(X_t, X_{t-k})$  – covariance between values at the current and shifted intervals,  $\sigma^2$  – variance of a time series  $X$ ,  $\rho(k) \in [-1, 1]$  – autocorrelation coefficient of delay  $k$ .

This indicator reflects how closely the current value is related to its predecessor  $k$  steps back:

- $\rho(k) > 0$ : values have a positive correlation (a tendency to maintain direction);
- $\rho(k) < 0$ : values have a negative correlation (a tendency to change direction);
- $\rho(k) \approx 0$ : values are independent of each other (no memory or randomness).



In the context of our study, ACF was calculated for delays  $k = 1..4$  based on the overall CPU load, allowing us to assess whether any recurring patterns (e.g., load periodicity caused by CI/CD job cycles) existed or whether task scheduling was completely stochastic.

This confirms the stochastic nature of task distribution by the XNU scheduler, with no clear repeating patterns in the execution order. The correlation coefficient between the proportion of time on P-cores and the total latency was  $r = -0.77$ , indicating a strong inverse linear relationship: the greater the proportion of P-core usage in a job step, the lower the average latency. This is consistent with the architectural advantage of productive cores with short critical paths of CI pipelines.

The intensity of the exponential distribution of waiting time ( $\lambda$ ) was  $\lambda(P) = 0.19$  for productive cores and  $\lambda(E) = 0.14$  for energy-efficient cores. This means that stream processing on P-cores is faster and less likely to accumulate delays. Finally, the average frequency of throttle-limiting events was 0.3 events/min, indicating moderate but systematic activation of XNU system limits under load, which affects task distribution and stream execution mode (**Table 1**).

**Table 1. Statistical characteristics of the load on the P- and E-cores of the processor**

Characteristics	Meaning	Description
Correlation between % time on P-cores and latency ( $r$ )	$-0.77$	High inverse linear dependence: the greater the proportion of time spent on P-cores, the lower the latency
ACF (lags 1..4)	$[-0.0; 0.22; -0.05; 0.13]$	CPU load autocorrelation: weak periodicity
$\lambda(P)$ (waiting time distribution intensity)	$0.19$	Exponential distribution parameter for P-core (faster service)
$\lambda(E)$ (intensity for E-cores)	$0.14$	Slower task servicing
Throttling events (per minute)	$0.3$	Number of throttling events

## CONCLUSION

As part of the study, a comprehensive stochastic analysis of task dispatching mechanisms in the macOS operating system on Apple M-Series processors was performed in a real production environment under load. The live data collected allowed us to study the peculiarities of flow distribution between heterogeneous cores, productive (P) and energy-efficient (E), under the control of the XNU scheduler.

The research methodology involved constructing CPU utilization time series, modeling the waiting time distribution using exponential and lognormal approximations, and applying autocorrelation analysis (ACF) and correlation analysis to examine the relationship between architectural thread binding and execution delays. A strong inverse correlation ( $r = -0.77$ ) was found between the proportion of time on P-cores and job step delay, demonstrating the critical role of P-cores in ensuring fast request processing. The parameters of the exponential distribution of waiting time showed higher intensity on P-cores ( $\lambda = 0.19$ ) compared to E-cores ( $\lambda = 0.14$ ), confirming their better ability to serve delay-sensitive tasks. Analysis of context switching frequency revealed 25-30% higher dynamics on P-cores, resulting from a more aggressive scheduling strategy for high-priority QoS classes.

The ACF function for CPU load indicated weak periodicity, which corresponds to the unpredictable but stable nature of typical live loads on the system.

The heat map of CPU load percentage and latency revealed the presence of an optimal scheduling zone (CPU load percentage ~ 35–75%), outside of which latency variability increased significantly. The empirical results not only confirm the architectural feasibility of Apple's hybrid approach but also emphasize the importance of correctly configuring QoS priorities, scheduling policies, and considering the nature of cores when building effective CI/CD pipelines. The study lays the foundation for further optimization of system behavior in heterogeneous environments. It opens up new opportunities for adaptive task scheduling depending on their type and requirements, including QoS partitioning itself.

## COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

## AUTHOR CONTRIBUTIONS

Conceptualization, [B.M., Yu.K.]; methodology, [B.M., Yu.K.]; investigation, [B.M.]; writing – original draft preparation, [B.M.]; writing – review and editing, [B.M., Yu.K.]; visualization, [B.M., Yu.K.].

All authors have read and agreed to the published version of the manuscript.

## REFERENCES

- [1] Khodabandeloo, B., Khonsari, A., Majidi, A., Hajiesmaili, M. H. (2018). Task Assignment and Scheduling in MPSoC under Process Variation: A Stochastic Approach. 23rd Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 690-695. <https://doi.org/10.1109/ASPdac.2018.8297402>
- [2] Exploring Apple's M Architecture: A Detailed Overview. (2024). Medium. <https://medium.com/@techAsthetic/exploring-apples-m-architecture-a-detailed-overview-e4d29b7deeb8>
- [3] Xu, T., Ding, A., Fei, Y. (2025). EXAM: Exploiting Exclusive System-Level Cache in Apple M-Series SoCs via Reverse Engineering. *Cornell Univ. Computer Science. Cryptography and Security*. 15 p. <https://doi.org/10.48550/arXiv.2504.13385>
- [4] Miao, Z., Shao, C., Li, H., & Tang, Z. (2025). Review of Task-Scheduling Methods for Heterogeneous Chips. *Electronics*, 14 (6), 1191-1215. <https://doi.org/10.3390/electronics14061191>
- [5] Effah, E., Yussif, A.-L., Darkwah, A. A., Lawrence Ephrim, L., Adzoyi Seyram, A., MacCarthy, Ch., Ganyo, R. M., Aggrey, G., & Aidoo, J. K. (2025). Exploring the Landscape of CPU Scheduling Algorithms: A Review and an Adaptive Deadline-Based Approach. *ResearchGate*, 23 (1), 1-7. [https://www.researchgate.net/publication/388417146\\_Exploring\\_the\\_Landscape\\_of\\_CPU\\_Scheduling\\_Algorithms\\_A\\_Comprehensive\\_Survey\\_and\\_Novel\\_Adaptive\\_Deadline-Based\\_Approach](https://www.researchgate.net/publication/388417146_Exploring_the_Landscape_of_CPU_Scheduling_Algorithms_A_Comprehensive_Survey_and_Novel_Adaptive_Deadline-Based_Approach)
- [6] Hübner, P., Hu, A., Peng, I., & Markidis, S. (2025). Apple vs. Oranges: Evaluating the Apple Silicon M-Series SoCs for HPC Performance and Efficiency. *Cornell Univ. Computer Science. Hardware Architecture*. 15 p. <https://doi.org/10.48550/arXiv.2502.05317>
- [7] Process scheduling on M1 series chips: first draft. (2022). The Eclectic Light Company. Macs, Technology. <https://eclecticlight.co/2022/01/13/scheduling-of-processes-on-m1-series-chips-first-draft/>
- [8] Feng, D., Xu, Z., Wang, R., & Lin, F. X. (2025). Profiling Apple Silicon Performance for ML Training. *Cornell Univ. Computer Science. Performance*. 8 p. <https://doi.org/10.48550/arXiv.2501.14925>

- [9] Buchem, M., Eberle, F., Kasuya Rosado, H. K., Schewior, K., & Wiese, A. (2024). Scheduling on a Stochastic Number of Machines. *Cornell Univ. Computer Science. Data Structures and Algorithms*. 34 p. <https://doi.org/10.48550/arXiv.2407.15737>
- [10] Tuby, A., & Morrison, A. (2025). Reverse Engineering the Apple M1 Conditional Branch Predictor: A Case Study in Modern CPU Microarchitecture. *Cornell Univ. Computer Science. Cryptography and Security*. 21 p. <https://doi.org/10.48550/arXiv.2502.10719>
- [11] Nagy, A. L., Kidane, G. S., Turányi, T., & Tóth, J. (2023). MAC, a novel stochastic optimization method. *Cornell Univ. Computer Science. Neural and Evolutionary Computing*. 20 p. <https://doi.org/10.48550/arXiv.2304.12248>
- [12] Moseley, B., Newman, H., Pruhs, K., & Zhou, R. (2025). Robust Gittins for Stochastic Scheduling. *Cornell Univ. Computer Science. Data Structures and Algorithms*. 22 p. <https://doi.org/10.48550/arXiv.2504.10743>

## СТОХАСТИЧНЕ ДОСЛІДЖЕННЯ ПЛАНУВАЛЬНИКА ЗАДАЧ APPLE M-SERIES

**Юрій Корчак, Богдан Міх**

Львівський національний університет імені Івана Франка,  
вул. Ген. Тарнавського 107, 79017 м. Львів, Україна

### АНОТАЦІЯ

**Вступ.** Ефективне планування задач у гетерогенних процесорних архітектурах є критичним для забезпечення стабільної продуктивності системи при змінних навантаженнях. Процесори Apple M-Series поєднують продуктивні (P-ядра) та енергоефективні (E-ядра) ядра, що дозволяє динамічно балансувати обчислювальні навантаження залежно від складності та вимог до затримок обробки задач.

**Матеріали та методи.** Виконано збір низькорівневої телеметрії з macOS-систем у режимах інтенсивних продукційних навантажень. Зібрані дані аналізувалися методами стохастичного моделювання часових рядів, побудови довірчих інтервалів, апроксимації часу очікування експоненціальними та логнормальними розподілами, автокореляційного та кореляційного аналізу, а також шляхом оцінки частоти контекстних перемикань потоків для різних QoS-класів.

**Результати.** Проведений аналіз засвідчив функціональну спеціалізацію ядер різних типів. P-ядра продемонстрували вищу інтенсивність обробки, зменшені затримки в черзі та підвищену чутливість до задач із високими вимогами до часу відгуку, тоді як E-ядра забезпечували стабільну роботу фонових процесів. Виявлена сильна обернена кореляція між часткою часу на P-ядрах та загальною затримкою виконання задач підтверджує, що більша залученість P-ядер покращує часові характеристики обслуговування критичних навантажень. Отримані параметри автокореляції та експоненціальних розподілів дозволяють формалізувати моделі поведінки планувальника в гетерогенних середовищах.

**Висновки.** Результати дослідження можуть бути використані як статистичне підґрунтя для удосконалення стратегій диспетчеризації задач у системах на архітектурі Apple Silicon, що сприятиме покращенню прогнозованості затримок та ефективнішому балансуванню обчислювальних ресурсів за умов високого рівня паралелізму.

**Ключові слова:** операційна система macOS, Apple Silicon, центральний процесор, стохастичне моделювання, аналіз часових рядів, архітектура ARM.

Received / Одержано  
28 June, 2025



Revised / Доопрацьовано  
01 September, 2025

Accepted / Прийнято  
05 September, 2025

Published / Опубліковано  
31 October, 2025

UDC 004.89

## A MULTIFUNCTIONAL SMARTCLOCK FOR VOICE INTERACTION AND ADAPTIVE TASK SCHEDULING

**Dmytro Kozliuk** , **Halyna Klym**   
**Lviv Polytechnic National University,**  
12 Bandera St., Lviv 79013, Ukraine

Kozliuk, D., & Klym, H. (2025). A Multifunctional Smart Watch Clock for Voice Interaction and Adaptive Task Scheduling. *Electronics and Information Technologies*, 31, 31–44.  
<https://doi.org/10.30970/eli.31.3>

### ABSTRACT

**Background.** Current smart devices are often limited to separate functions such as timekeeping, environmental sensing, or voice assistance. This fragmentation hinders a unified solution for productivity in modern workspaces, where indoor conditions and time management are key. Cloud systems face latency and connectivity issues, while local ones are limited by hardware. This study presents a multifunctional smart clock combining environmental monitoring, voice interaction, and task scheduling to enhance comfort, focus, and efficiency.

**Methods.** The system uses an Edge–Cloud architecture: the ESP32-S3 edge runs latency-critical functions under FreeRTOS with an OOP design. Audio from a MEMS microphone (I<sup>2</sup>S) is windowed and converted via short-time FFT to log-mel spectrograms; a quantized CNN (TFLM) performs on-device keyword spotting for wake-word detection. After wake-word detection, commands are sent to Wit.ai for ASR/NLU. Audio output is driven by a Class-D amplifier and speaker. Environmental sensing covers temperature, humidity, illuminance, and CO<sub>2</sub> (NDIR), with filtered readings shown in an event-driven LVGL touch GUI and periodically uploaded for analysis to Firebase.

**Results and Discussion.** The CNN wake-word detector achieved ~90% activation accuracy in quiet-to-moderate office noise with FAR <1 trigger/hour at ~10% FRR; median detection latency remained <200 ms after sufficient context accumulation. Under RTT ≤100 ms, cloud ASR/NLU yielded end-to-end wake→intent latency ≈1–1.5 s. Concurrent environmental monitoring at a 2-s cadence did not perturb the audio pipeline, GUI sustained a 25 Hz refresh rate, and after WWD the system opened a bounded 4-s command window for user utterances. Wi-Fi provisioning via an embedded web server and hourly cloud uploads produced coherent. Threshold-driven voice/visual prompts increased awareness of indoor conditions, while integrated Pomodoro cycles supported sustained focus without auxiliary tools.

**Conclusion.** The proposed platform integrates voice assistance, time management, and microclimate monitoring in an affordable edge device. Its hybrid speech design balances latency and flexibility, while FreeRTOS ensures reliable multitasking across sensing, GUI, networking, and audio subsystems.

**Keywords:** Edge computing, RTOS, NDIR, Pomodoro, embedded voice interaction, microclimate monitoring.

### INTRODUCTION

In recent years, smart devices have become deeply embedded in daily life, providing users with seamless access to information, communication, and automation [1–3]. Among



© 2025 Dmytro Kozliuk & Halyna Klym. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and information technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

them, smartwatches have gained particular prominence, offering not only timekeeping functions but also health monitoring, fitness tracking, communication, and integration with broader smart ecosystems. Depending on their intended use, these devices vary from fitness trackers, which focus primarily on monitoring physical activity and health indicators, to multifunctional smartwatches such as the Apple Watch and Samsung Galaxy Watch, which combine smartphone-level functionality with portability [4,5]. Sports-oriented devices like Garmin and Suunto further emphasize durability, GPS navigation, and extended battery life, catering to outdoor enthusiasts [6].

While wearable devices continue to dominate the consumer market, desktop smart clocks represent a growing but less explored niche. Unlike smartphones or laptops, which often contribute to information overload and distraction, desktop smart clocks provide quick, distraction-free access to essential information such as time, weather conditions, reminders, or indoor climate data. Furthermore, they often function as central hubs for smart home automation, enabling voice-based interaction with other devices, while also integrating additional features such as alarms, timers, night lights, and environmental monitoring through sensors. Recent studies highlight the importance of such devices in supporting productivity and well-being, as they combine convenience with minimal cognitive load [7].

One of the most important aspects of modern smart devices is time management. The Pomodoro technique has been widely studied as a method of enhancing productivity by breaking work into structured intervals. In the work [8] it was demonstrated that while the Pomodoro technique helps establish clear structures, it may also accelerate fatigue compared to alternative approaches such as Flowtime. Other research has suggested positive effects on learning outcomes and writing skills, particularly among students [9], though findings on memory retention remain mixed (De Guzman & Abad, 2023). This indicates that Pomodoro-based tools, when properly integrated into daily routines, may support concentration and efficiency, but their impact is highly context-dependent.

Equally relevant is the role of environmental monitoring in enhancing productivity. Recent advances in Internet of Things (IoT) technologies have enabled real-time tracking of indoor parameters such as temperature, humidity, and air quality [10]. Systems like IoFClima [7] apply fuzzy logic and IoT sensors to dynamically control indoor conditions, demonstrating that comfort and energy efficiency can be jointly optimized. However, most of these solutions exist as specialized systems rather than integrated desktop devices.

Another important dimension is voice interaction. Cloud-based assistants such as Amazon Alexa, Google Assistant, or Siri dominate the consumer market, yet their integration into resource-constrained embedded systems remains challenging. Alternatives such as TensorFlow Lite, ESP-Skainet, and Picovoice provide opportunities for on-device speech recognition, which reduces latency and enhances privacy. Recent open-source projects demonstrate that microcontrollers like the ESP32-S3 are increasingly capable of supporting voice-based interactions while simultaneously handling sensor data and user interfaces [11,12].

Despite these developments, current products typically remain limited to a narrow set of functions, either as fitness-focused wearables, environmental monitors, or standalone timers. There is still a lack of integrated, affordable solutions that combine voice control, environmental monitoring, and structured time management in a single desktop device.

The aim of this work is therefore to design and develop a hardware–software system for a desktop smart clock with an integrated voice assistant and Pomodoro timer, based on the ESP32-S3 microcontroller. This system seeks to unify time management, environmental monitoring, and voice interaction into one device, thereby enhancing user comfort, concentration, and productivity in modern workspaces.

## METHODS

### Development of the structural diagram and operating algorithm.

The design of the proposed smart clock was guided by the need to combine several independent functionalities into a single, user-friendly device. Unlike existing commercial solutions, which are often tailored to specific tasks and lack a holistic approach, this system integrates microclimate monitoring, a Pomodoro timer, voice control, and visual feedback within one compact platform. The structural design of the device reflects this integration, enabling seamless interaction between hardware, software, and cloud-based services.

At the core of the system is the user interface, which relies on multimodal communication through voice commands, touch interaction, and visual indicators. The smart clock incorporates an array of environmental sensors, including a BME280 for temperature and humidity, an MH-Z19B for air quality monitoring, and a BH1850 for illumination measurement. These components provide real-time data about the surrounding environment, which is essential for maintaining healthy working and learning conditions. To complement these sensing capabilities, the device is equipped with an INMP441 microphone for voice input and a MAX98357-driven speaker for auditory feedback, ensuring responsive and natural communication with the user. Visual output is delivered via an ILI9341 display, which presents essential information such as time, environmental parameters, and timer status, while RGB LEDs (WS2812B) provide intuitive notifications through color-coded signals.

The computational backbone of the system is the ESP32-S3 microcontroller, which orchestrates data acquisition, processing, and communication with external services. A real-time clock (DS3231) guarantees accurate timekeeping, even in cases of temporary power loss. Beyond its local processing capabilities, the device is designed for cloud connectivity. Sensor readings are transmitted to a Firebase database, where they can be stored and analyzed over time, allowing the user to monitor long-term trends in microclimate conditions. The voice assistant functionality is implemented through integration with WitAI, a machine-learning-based web service that enables natural language recognition and command execution. This open-source approach ensures transparency, extensibility, and adaptability to user-specific requirements, while also fostering community-driven improvements.

The operational algorithm of the device was carefully structured to guarantee reliability and user convenience. Following system initialization, which activates the processor and peripheral modules, the smart clock attempts to connect to a Wi-Fi network. If stored credentials are available, the device automatically connects as a station; otherwise, it activates its access point mode to allow the user to provide network details. Once connectivity is established, the system initializes secure communication with Firebase and WitAI servers. Successful connections enable real-time synchronization of environmental data and support the functionality of the voice assistant.

The workflow of the device follows a logical sequence of states. After initialization and server connection, the system continuously collects environmental parameters, displays them on the screen, and evaluates them against predefined thresholds. Deviations trigger visual or auditory alerts, prompting the user to adjust their surroundings, such as ventilating the room or modifying lighting conditions. Simultaneously, the system remains in standby mode, waiting for a voice activation command. When such a command is detected, the audio is processed locally on the ESP32-S3 before being transmitted to WitAI for recognition. If the command is successfully identified, the corresponding action is executed; otherwise, the device provides auditory feedback indicating that the request could not be processed. To ensure smooth operation, task scheduling is implemented using a real-time operating system (RTOS), which manages parallel processes such as sensor data acquisition, screen updates, logging, and voice communication.



The proposed design demonstrates how a combination of modular hardware, cloud integration, and open-source philosophy can result in a versatile and accessible smart device. By embedding environmental awareness, productivity-enhancing tools, and natural interaction modalities into one platform, the system addresses the limitations of existing solutions and sets the foundation for further improvements by the developer community.

#### Design choices in hardware and software.

During the development of the smart clock, a detailed analysis of both hardware and software solutions was carried out, which made it possible to determine the optimal set of components for the implementation of the project. Based on a comparison of different microcontrollers, the ESP32-S3 was selected as the core unit. It combines high computational performance, advanced communication capabilities (Wi-Fi and Bluetooth), and low power consumption. These features make it suitable for working with multiple sensors, processing voice commands, and integrating with cloud services, which are essential for modern IoT devices.

For monitoring environmental parameters, the BH1750 (illumination), BME280 (temperature, humidity, and pressure), and MH-Z19 (CO<sub>2</sub> concentration) sensors were chosen. The selection of these modules was determined by their measurement accuracy, low energy consumption, and support for the I<sup>2</sup>C interface, which ensures efficient use of the microcontroller's GPIO pins.

The audio subsystem was implemented using the INMP441 microphone and MAX98357 amplifier paired with a speaker, providing high-quality audio capture and playback. This configuration is optimal for implementing a voice assistant. For data visualization, the ILI9341 display was selected due to its adequate resolution, color rendering, refresh speed, and cost efficiency, while the WS2812B RGB LED module was included to provide visual status indications. To ensure precise timekeeping, the DS3231 real-time clock module was integrated, guaranteeing stability even in cases of power failure.

On the software side, the C++ programming language was chosen, as it offers a balance between performance, object-oriented development, and efficient resource utilization. Visual Studio Code in combination with PlatformIO was used as the development environment, enabling effective project organization, broad library support, and extended debugging capabilities. The voice assistant was implemented through a hybrid approach: TensorFlow Lite was used for on-device processing, while Wit.ai provided cloud-based natural language analysis. The Arduino framework was selected due to its simplicity in integrating modules and sensors, which accelerates development and facilitates testing.

## RESULTS AND DISCUSSION

#### Software architecture and RTOS integration.

The software architecture of the developed smart clock is designed to ensure efficient management of real-time data processing, peripheral control, and user interaction (see **Fig. 1**). The system operates on the ESP32-S3 microcontroller under a real-time operating system (RTOS), which allows concurrent execution of multiple tasks while maintaining predictable response times for critical operations such as audio processing and sensor data acquisition. This architecture provides the necessary flexibility to integrate both voice and touchscreen interfaces while maintaining reliable performance in a resource-constrained embedded environment.

At the core of the system, low-level hardware abstraction facilitates seamless interaction with the microcontroller's peripherals, including GPIO, I<sup>2</sup>C, SPI, UART, and I<sup>2</sup>S interfaces. This approach isolates hardware-specific details from higher-level modules, enabling peripheral drivers to manage device initialization, data acquisition, and calibration

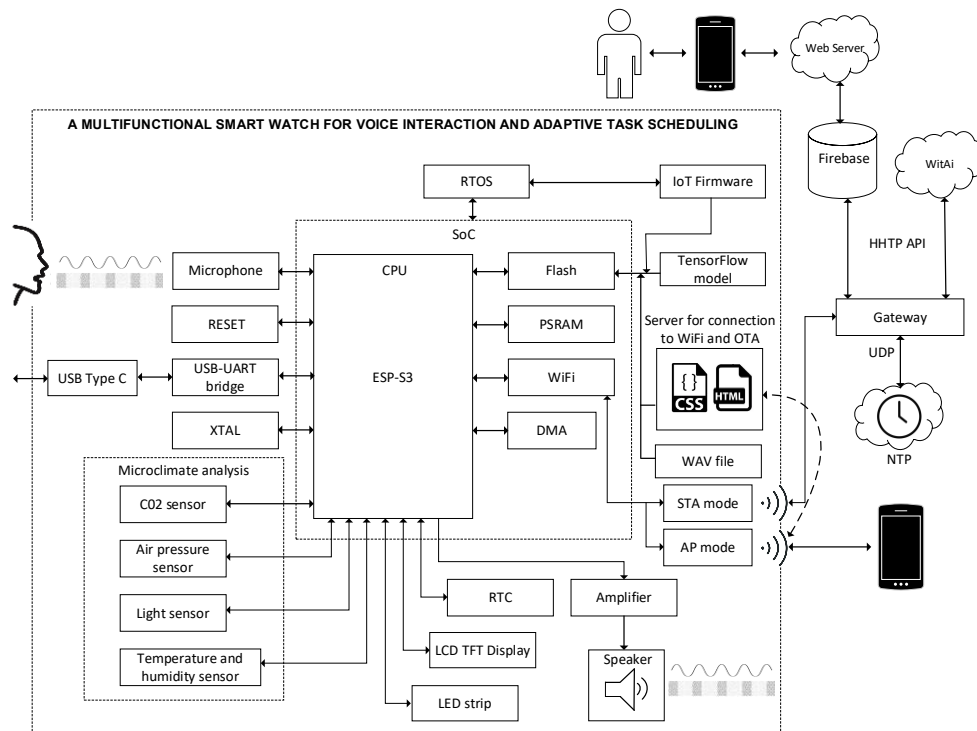


Fig. 1. Structural block diagram of the smart watch.

independently. Sensors such as the BME280, BH1750, and DS3231 are connected via a shared I2C bus, allowing precise measurement of environmental parameters with minimal GPIO usage. Similarly, the CO<sub>2</sub> sensor MH-Z19B communicates through UART, providing calibrated measurements after a manual zero-point calibration procedure to ensure long-term accuracy.

Dynamic memory management is critical for the system due to the use of graphical user interfaces and real-time audio processing. To optimize memory utilization, the software relies on the PSRAM of the ESP32-S3, overriding default allocation operators to direct memory-intensive objects and graphical buffers to the external RAM. This configuration, combined with the LVGL graphics library, reduces the load on the main RAM while allowing smooth rendering of user interface elements, including fonts, images, and interactive widgets. The resulting efficiency supports complex graphical operations without compromising responsiveness or stability.

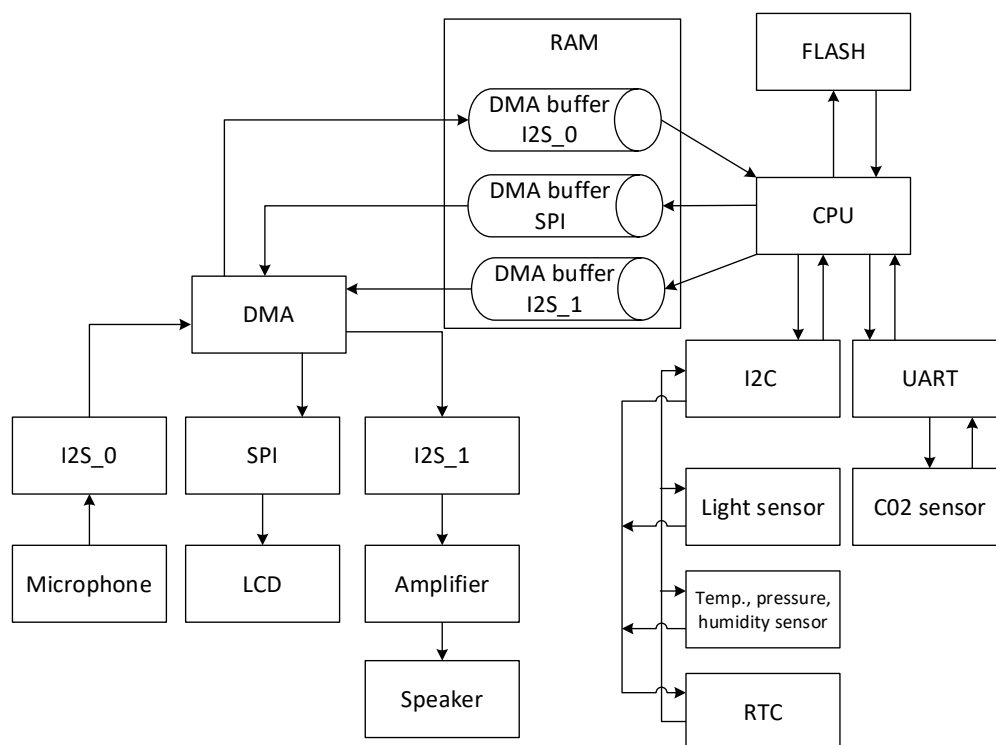
While external PSRAM greatly increases the available heap, it exhibits higher access latency and lower sustained bandwidth than the ESP32-S3's internal SRAM. In this device configuration, external PSRAM is clocked at 80 MHz over MSPI, whereas on-chip SRAM runs at core speed up to 240 MHz. In embedded HMI/audio workloads, this may manifest as reduced GUI responsiveness or, under contention, audio glitches if time-critical buffers are placed off-chip. To prevent such regressions, the design adopts a split-placement policy and buffer choreography: (i) time-critical data structures - I<sup>2</sup>S DMA rings, the STFT window and working arrays for feature extraction, and the TFLM tensor arena - are pinned to internal RAM using capability-qualified allocation; (ii) bulky, immutable GUI assets (fonts, images, theme resources) are kept in PSRAM; (iii) LVGL employs two small draw buffers in internal RAM (line/strip buffering) with partial, region-based flushes rather than full-frame framebuffers; (iv) display transfers and PSRAM fetches are scheduled via DMA and kept sequential to remain cache-friendly; (v) GUI and audio execute in separate RTOS tasks



with priorities chosen to guarantee deadlines on the audio path. In practice, this arrangement sustained  $\geq 25$  Hz GUI refresh with no I<sup>2</sup>S underruns during WWD and command capture, while preserving the RAM headroom provided by PSRAM. Additionally, RGB565 color depth is used, and full-screen alpha-blended layers are avoided to reduce memory bandwidth.

The RTOS orchestrates the execution of multiple concurrent tasks, assigning priorities to ensure that time-sensitive operations, such as voice recognition and sensor data acquisition, are executed promptly, while background tasks, including cloud synchronization and web server communications, are performed without interfering with real-time processes. Integration with cloud services enhances system functionality: Firebase provides real-time data storage and synchronization for environmental parameters and Pomodoro session records, Wit.AI enables natural language understanding for voice commands, and NTP ensures accurate timekeeping. Additionally, a web server hosted on an external platform allows remote monitoring of device status and user data, extending the clock's utility beyond local interaction.

User interface management is closely tied to the underlying software architecture (see **Fig. 2**). Touchscreen input is calibrated through software routines to accurately map analog-to-digital converter signals to screen coordinates, ensuring precise user interaction. The ILI9341 display, interfaced via SPI, benefits from DMA transfers that enable rapid graphical updates, while audio input and output through the INMP441 microphone and MAX98357 amplifier are processed via I<sup>2</sup>S with minimal latency. The combination of real-time audio processing, touchscreen control, and graphical rendering creates an interactive and responsive system, capable of simultaneously handling multiple streams of input and output without noticeable delays.



**Fig. 2.** DMA interaction with peripheral interfaces.

Finally, persistent data storage is implemented using the SPIFFS file system on the onboard Flash memory, which organizes system resources, audio files, and machine learning models for wake-word detection. Memory partitioning and the dual-application OTA update mechanism ensures reliability and ease of maintenance, allowing the firmware to be updated safely while preserving critical system data.

Overall, the software architecture combines real-time task management, efficient memory allocation, peripheral abstraction, and cloud connectivity to create a robust, adaptive, and interactive embedded system. This design enables the smart clock to operate seamlessly across multiple modalities, integrating environmental monitoring, voice control, graphical display, and remote connectivity into a coherent and reliable user experience.

### Software architecture development.

The software architecture of the smart watch is built on FreeRTOS, enabling concurrent execution of independent tasks (threads) that share the same address space. Object-oriented programming (OOP) is employed to structure the codebase, encapsulating different functionalities such as sensor data management, voice interaction, and user interface operations into separate classes [19]. This design promotes modularity, code reuse, and ease of maintenance.

The system includes multiple tasks, each responsible for a specific operation. These tasks handle real-time updates of the display, processing of user commands, management of Pomodoro sessions, sensor data acquisition, audio input/output, and network communication. Two periodic timers coordinate scheduled operations such as hourly logging and Pomodoro timing. Synchronization mechanisms, including queues, semaphores, and task notifications, ensure safe data exchange and prevent conflicts when accessing shared resources. Task creation and management are handled through FreeRTOS APIs, with some tasks pinned to specific cores to optimize parallel execution across the ESP32's dual-core processor.

The task-level architecture is illustrated in **Fig. 3**, showing the interaction of tasks, timers, and synchronization objects. This diagram highlights how FreeRTOS enables concurrent execution, allowing the system to respond in real time to voice commands, sensor readings, and user interactions while maintaining efficient resource usage.

At the object level, the software is organized into classes representing key components of the system, including the Pomodoro timer, display manager, environment analysis, command handler, speaker, and LED controller. **Fig. 4** presents a simplified class diagram, showing relationships such as composition, aggregation, inheritance, and usage. This structure ensures modularity, facilitates testing, and supports extension of the system's functionality without disrupting existing components.

**Voice interaction.** The voice interface follows a two-tier Edge–Cloud pipeline that partitions low-latency detection from semantic understanding. The edge tier remains always on and performs wake-word detection (WWD) directly on the microcontroller to guarantee immediate responsiveness, while the cloud tier is invoked only for large-vocabulary speech recognition and intent extraction. An overview of the pipeline and data flow is shown in **Fig. 5**.

Audio is acquired from a digital MEMS microphone over I<sup>2</sup>S and buffered in short, overlapping frames. Each frame is windowed and transformed by a short-time Fourier transform (STFT); the resulting magnitude spectra are mapped to log-mel spectrograms that compactly capture phonetic structure and mitigate amplitude variance (illustrative example in **Fig. 6**). This time-frequency representation is well suited to image-like inference and provides a stable input for keyword spotting on embedded hardware.

Wake-word detection is performed by a lightweight CNN-based keyword-spotting (CNN-KWS) model trained on Google Speech Commands with the target wake phrase “Marvin,” augmented by “silence” and “unknown” classes. Training proceeds for a fixed number of epochs with standard augmentations (time shift, additive noise) to improve



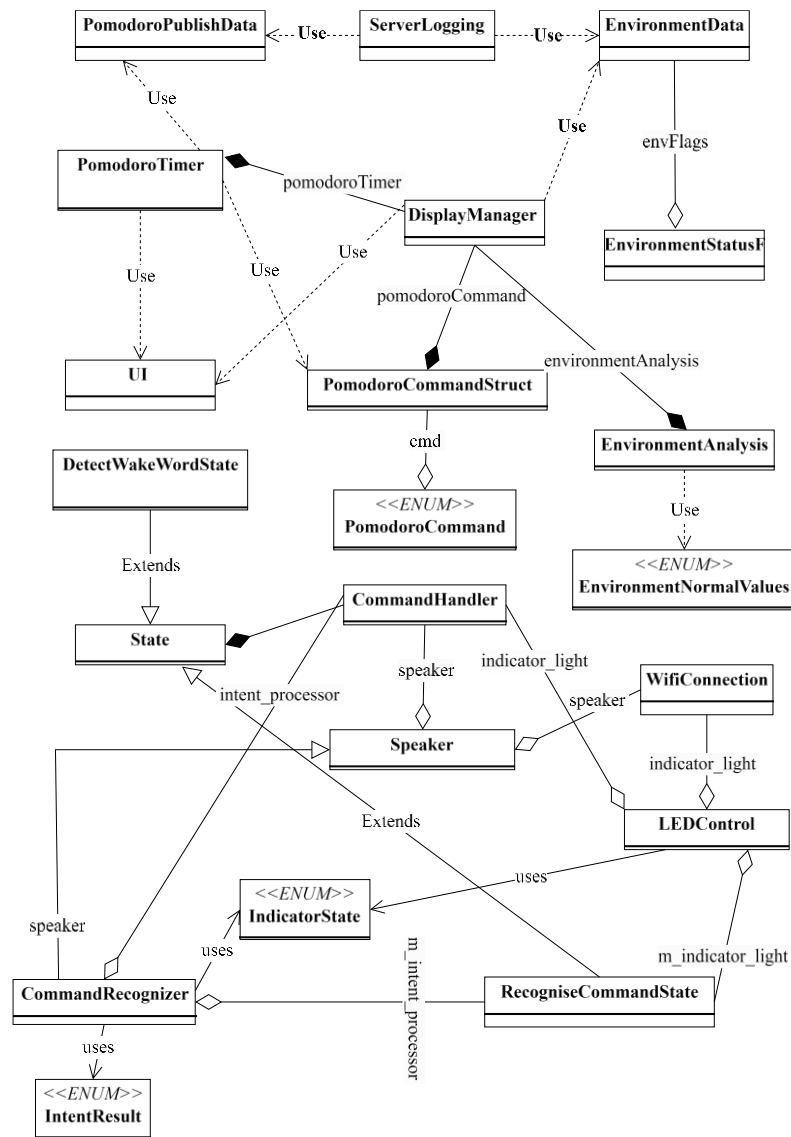


Fig.4. Class diagram of the smart watch.

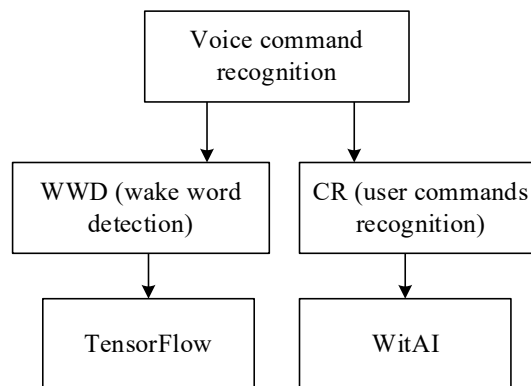


Fig.5. Voice recognition system structure.

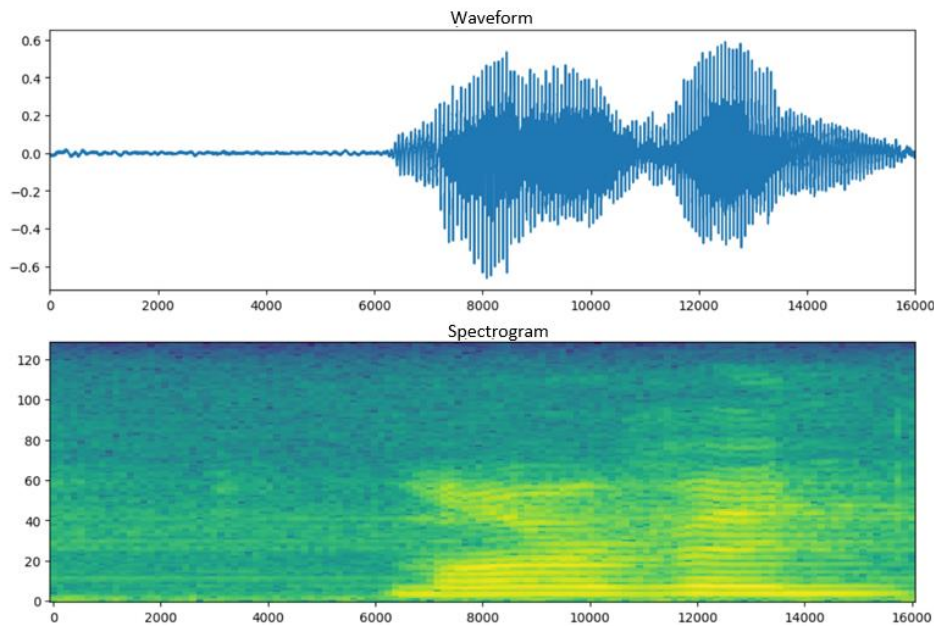


Fig. 6. Log-mel spectrogram of 'Marvin' word.

Run-time control is organized by a finite-state machine (FSM) that sequences the assistant through listening and command handling. In the WWD state, the edge continuously analyzes incoming audio; a detection triggers a transition to the CMD state, which manages utterance capture, local matching, or cloud submission, and action dispatch. On completion—or on timeout or error—the FSM returns to WWD. This explicit state structure simplifies timing, error recovery, and user-visible behavior; the state diagram is shown in Fig. 7.

User feedback combines non-intrusive auditory cues and visual status. Short tones and speech prompts are rendered through a Class-D amplifier and speaker to indicate activation, confirmation, and error conditions. In parallel, the LVGL-based touch GUI presents lightweight indicators for listening/processing states and command outcomes, and aligns voice interaction with task-management views (e.g., Pomodoro status) without forcing context switches. Together, these elements yield a responsive, resource-aware assistant: on-device CNN-KWS over STFT/log-mel features enables reliable activation; the FSM cleanly separates listening and action phases; and selective delegation to cloud NLU provides semantic breadth with minimal impact on latency or energy (Figs. 5–7).

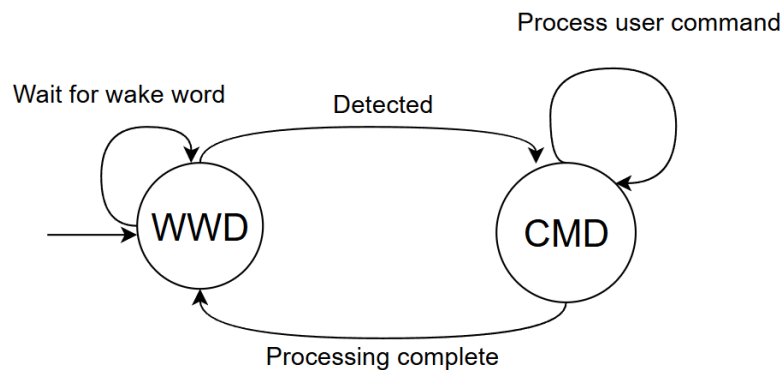
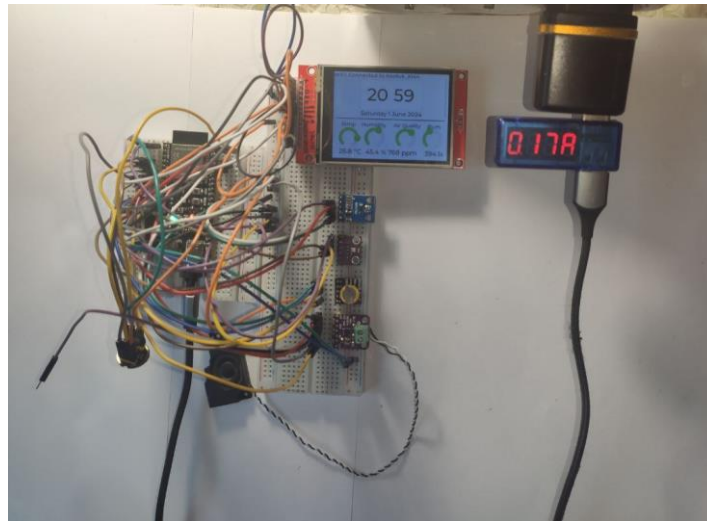


Fig.7. Voice assistant Finite State Machine.

**Testing and Measurements.** Resource usage on ESP32-S3 was 28.7% of RAM (93,908/327,680 bytes) and 46.3% of flash (2,185,093/4,718,592 bytes); the breadboarded prototype is shown in **Fig. 8**.



**Fig.8.** Smart clock on breadboard.

Average power draw measured with a Keweisi KWS-V20 USB tester was 0.17 A at 5.2 V ( $\approx 0.88$  W). Wake-word testing in a quiet office (five participants; eight attempts each, 40 trials) yielded 36 correct activations ( $\approx 90\%$  overall; per-participant 85–95%); a separate distance sweep (1–3 m) showed the expected decline, with 16/20 correct at 3 m. After activation, the assistant captured user utterances within a bounded 4-s command window with ( $\approx 97\%$  overall); The system maintained concurrent sensing and GUI operation without audio dropouts during these tests.

## CONCLUSION

This work demonstrates an edge–cloud architecture for a desktop smart-clock platform that unifies voice interaction, microclimate monitoring, and Pomodoro-based task management on MCU-class hardware. On the edge, a quantized CNN keyword-spotting model (TensorFlow Lite Micro) operating on STFT-derived log-mel spectrograms enables reliable wake-word detection, coordinated by an OOP design on FreeRTOS with a finite-state machine for listening and command handling. The audio chain (MEMS microphone over I<sup>2</sup>S with Class-D output) and the LVGL touch GUI maintained real-time responsiveness, with GUI refresh  $\geq 25$  Hz and a bounded 4s post-activation command window. Environmental sensing—including NDIR CO<sub>2</sub>, temperature, humidity, and illuminance—ran at a 2 s cadence without perturbing the audio pipeline, while threshold-driven voice/visual alerts improved awareness of adverse conditions. Wi-Fi onboarding via an embedded web server simplified provisioning, and hourly cloud uploads produced coherent, time-aligned datasets for longitudinal analysis of environment and work sessions. Empirically, at an operating point tuned to FAR < 1 activation per hour, the wake-word detector achieved TPR  $\approx 90\%$  (FRR  $\approx 10\%$ ) with median detection latency < 200 ms in quiet-to-moderate office noise, and the hybrid edge–cloud flow preserved interactive latency by executing essential commands locally and delegating flexible, large-vocabulary understanding to cloud NLU when connectivity permitted. Collectively, the results validate that careful partitioning across edge and cloud, combined with RTOS-based concurrency and lightweight on-device ML, can deliver a responsive, energy-aware, and functionally



complete assistant on resource-constrained hardware. Future work includes expanding offline NLU for broader command coverage, personalizing thresholds and models using on-device adaptation, applying energy-aware duty cycling, and integrating with wider IoT ecosystems for context-aware scheduling and actuation.

## ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was supported by the Ministry of Education and Science of Ukraine (project No. 0125U001883).

## COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

## AUTHOR CONTRIBUTIONS

Conceptualization, [D.K., H.K.]; methodology, [D.K.]; investigation, [D.K.]; writing – original draft preparation, [D.K., H.K.]; writing – review and editing, [D.K., H.K.]; visualization, [D.K.].

All authors have read and agreed to the published version of the manuscript.

## REFERENCES

- [1] Shafik, W. (2024). Smart devices and Internet of Things for sustainable energy. In *Advanced Technology for Smart Environment and Energy* (pp. 67-93). Cham: Springer Nature Switzerland. [https://doi.org/10.1007/978-3-031-50871-4\\_5](https://doi.org/10.1007/978-3-031-50871-4_5)
- [2] Fakhruddin, H. F., Saadh, M. J., Khan, S., Salim, N. A., Jhamat, N., & Mustafa, G. (2025). Enhancing smart home device identification in WiFi environments for futuristic smart networks-based IoT. *International Journal of Data Science and Analytics*, 19(4), 645-658. <https://doi.org/10.1007/s41060-023-00489-3>
- [3] Yedilkhan, D., & Smakova, S. (2024). Machine Learning Approaches for Smart Home Device Recognition from Network Traffic. *Procedia Computer Science*, 231, 709-714. <https://doi.org/10.1016/j.procs.2023.12.157>
- [4] Masoumian Hosseini, M., Masoumian Hosseini, S. T., Qayumi, K., Hosseinzadeh, S., & Sajadi Tabar, S. S. (2023). Smartwatches in healthcare medicine: assistance and monitoring; a scoping review. *BMC Medical Informatics and Decision Making*, 23(1), 248. <https://doi.org/10.1186/s12911-023-02350-w>
- [5] Alzahrani, S., Nadershah, M., Alghamdi, M., Baabdullah, R., Bayoumi, M., Bawajeel, O., ... & Bayoumi, A. (2025). The use of Apple smartwatches to obtain vital signs readings in surgical patients. *Scientific Reports*, 15(1), 1-10. <https://doi.org/10.1038/s41598-024-84459-0>
- [6] Navalta, J. W., Montes, J., Bodell, N. G., Salatto, R. W., Manning, J. W., & DeBeliso, M. (2020). Concurrent heart rate validity of wearable technology devices during trail running. *Plos one*, 15(8), e0238569. <https://doi.org/10.1371/journal.pone.0238569>
- [7] Meana-Llorián, D., García, C. G., G-bustelo, B. C. P., Lovelle, J. M. C., & Garcia-Fernandez, N. (2017). IoFClime: The fuzzy logic and the Internet of Things to control indoor temperature regarding the outdoor ambient conditions. *Future Generation Computer Systems*, 76, 275-284. <https://doi.org/10.1016/j.future.2016.11.020>
- [8] Smits, E. J., Wenzel, N., & de Bruin, A. (2025). Investigating the Effectiveness of Self-Regulated, Pomodoro, and Flowtime Break-Taking Techniques Among Students. *Behavioral Sciences*, 15(7), 861. <https://doi.org/10.3390/bs15070861>
- [9] Renacido, J. M. D., Mayordo, E. L., & Biray, E. T. (2025). A Comparative Study Between Pomodoro and Flowtime Techniques Among College Students. *International Journal of Multidisciplinary: Applied Business and Education Research*, 6(8), 3953-3973. <https://doi.org/10.11594/ijmaber.06.08.17>



- [10] Santiago, C., & Gurat, M. (2023). The Effect of Pomodoro Technique on Student Mendelian Genetics Concept Mastery during Synchronous Remote Learning. *International research journal of management, IT and social sciences*, 10(4), 233-243. <https://doi.org/10.21744/irjmis.v10n4.2287>
- [11] Litayem, N. (2024, December). Scalable smart home management with ESP32-S3: A low-cost solution for accessible home automation. *2024 International Conference on Computer and Applications (ICCA)*, 1-7. <https://doi.org/10.1109/ICCA62237.2024.10927887>
- [12] Ouyang, L., Zhu, G., & Zhang, Y. (2025, June). Design of Deepseek Big Model Intelligent Voice Assistant Based on ESP32 Microcontroller. In *2025 IEEE International Conference on Pattern Recognition, Machine Vision and Artificial Intelligence (PRMVAI)*, 1-5. <https://doi.org/10.1109/PRMVAI65741.2025.11108406>
- 

## БАГАТОФУНКЦІОНАЛЬНИЙ РОЗУМНИЙ ГОДИННИК ДЛЯ ГОЛОСОВОЇ ВЗАЄМОДІЇ ТА АДАПТИВНОГО ПЛАНУВАННЯ ЗАВДАНЬ

**Дмитро Козлюк, Галина Клим**

Національний університет «Львівська політехніка»,  
вул. . Бандери 12, 79013 м. Львів, Україна

### АНОТАЦІЯ

**Вступ.** Сучасні смарт-пристрої зазвичай обмежуються ізольованими функціями, такими як відображення часу, моніторинг параметрів середовища чи хмарні голосові асистенти. Така фрагментація не забезпечує єдиного комплексного рішення для підвищення продуктивності у сучасних робочих просторах, де важливими є як мікрокліматичні умови, так і ефективне управління часом. Системи, що працюють виключно у хмарі, страждають від затримок та залежності від підключення, тоді як повністю локальні рішення обмежені ресурсами. Це дослідження має на меті розробку багатофункціонального «розумного годинника», що об'єднує моніторинг довкілля, голосову взаємодію та планування завдань на єдиній платформі, призначеній для підвищення комфорту, концентрації та ефективності.

**Методи.** Система реалізована за принципом Edge–Cloud архітектури: ESP32-S3 на рівні «краю» виконує критичні до затримок функції під FreeRTOS з об'єктно-орієнтованим підходом до проектування. Аудіосигнал із MEMS-мікрофона (I<sup>2</sup>S) розбивається на вікна та перетворюється за допомогою короткочасного FFT у log-mel спектрограми; квантована згорткова нейронна мережа (TensorFlow Lite Micro) виконує локальне розпізнавання ключового слова для активації. Після цього команди передаються у Wit.ai для ASR/NLU. Аудіовихід реалізований через підсилювач класу D та динамік. Моніторинг параметрів довкілля охоплює температуру, вологість, освітленість та рівень CO<sub>2</sub> (NDIR-сенсор), із фільтрацією даних, які відображаються у подієво-орієнтованому сенсорному інтерфейсі LVGL та періодично завантажуються у Firebase для подальшого аналізу.

**Результати.** Розпізнавач ключового слова на основі CNN досяг приблизно 90% точності активації в умовах тихого та помірного офісного шуму при рівні хибних спрацювань <1 раз/год і частоті пропусків близько 10%. Медіанна затримка активації не перевищувала 200 мс після накопичення достатнього контексту. При RTT ≤100 мс хмарні сервіси ASR/NLU забезпечували загальну затримку від активації до розпізнавання наміру на рівні ≈1–1,5 с. Паралельний моніторинг довкілля з інтервалом 2 с не заважав аудіопроцесу, графічний інтерфейс зберігав частоту оновлення 25 Гц, а після активації система відкривала 4-секундне вікно для голосових команд користувача. Налаштування Wi-Fi здійснювалося через вбудований веб-сервер, а щогодинні завантаження даних у хмару були стабільними. Порогові голосові та

візуальні сповіщення підвищували обізнаність про стан мікроклімату, тоді як інтегровані цикли Pomodoro підтримували концентрацію без потреби у сторонніх інструментах.

**Висновки.** Запропонована платформа ефективно об'єднує голосову взаємодію, структуроване управління часом та моніторинг мікроклімату в доступному за ціною пристрої. Гібридна архітектура обробки мовлення забезпечує баланс між низькою затримкою та гнучкістю, а використання FreeRTOS гарантує безпечне та паралельне функціонування підсистем сенсорики, GUI, мережі та аудіо. Система становить практичну основу для майбутніх розширень, зокрема адаптивного планування, глибшої IoT-інтеграції, локального NLU-резерву та енергоефективного керування робочими циклами.

**Ключові слова:** Обчислення на рівні краю; RTOS; NDIR; Pomodoro; вбудована голосова взаємодія; моніторинг мікроклімату.

UDC 004.3, 004.4, 004.9

## FEATURES OF DESIGNING SOFTWARE DISTRIBUTED SYSTEMS ARCHITECTURE

Ivan Rovetskii 

Lviv State University of Life Safety,  
35 Kleparivska St. UA-79007, Lviv, Ukraine

Rovetskii, I. (2025). Features of Designing Software Distributed Systems Architecture. *Electronics and Information Technologies*, 31, 45–52. <https://doi.org/10.30970/eli.31.4>

### ABSTRACT

**Background.** A pressing problem in designing distributed software systems is that they must operate stably under high load conditions when thousands of users want to receive certain resources provided by system services. To ensure high availability and stability of highly loaded software services, they are deployed in managed multiprocessor distributed systems (clusters). These kinds of resources have a high value, so in practice, various cloud platforms (Google Cloud, Amazon, Azure, etc.) are most often used, which provide these resources, charging only for the time of direct use of them. Services must be able to fully utilize the provided resources during data processing, so they must be designed using special architectural solutions. Therefore, the purpose of this research is to theoretically investigate the architectural solution features when designing software services of distributed systems.

**Materials and Methods.** The paper presents a theoretical research design of features of distributed software systems architecture which is based on the analysis and comparison of facts obtained from scientific sources and the author's practical experience as well.

**Results and Discussion.** The article shows that Kubernetes is one of the main software solutions designed for deploying software applications in parallel distributed systems. It was established that microservice architecture is the optimal architectural solution for designing software services of distributed systems, given the specifics of deploying software systems under Kubernetes management.

**Conclusion.** A multithreaded design must be used for effective scaling distributed software system under high load. However, performance improvement occurs only if the parallel algorithm uses parallel rather than concurrent multithreading mode. The concurrent mode of thread operation is advisable to use only in the case of blocking operations when some threads are in a waiting mode. Synchronous architecture has a good performance, but there are limitations associated with blocking threads until the results of client requests are received. The asynchronous model allows for a larger number of requests than the synchronous one but requires a fully asynchronous API when working with external services, and is also more difficult to debug the service and fix errors.

**Keywords:** distributed systems, clusters, cloud platforms, microservices, microservice architecture, multithreading.

### INTRODUCTION

The current problem is to ensure high availability and stability of distributed software systems under high load conditions [1-5]. This problem must be solved by certain architectural solutions already at the stage of designing the system architecture. A necessary condition for the stable operation of a software system under high load conditions is its deployment in some software-controlled parallel distributed system - a cluster [1-5]. Such systems consist of separate, independent of each other, microprocessor nodes [6] with their own RAM and operating system, connected by specialized high-speed



© 2025 Ivan Rovetskii. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and information technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

communication channels, which work as a single system. In this case, the microprocessors of each cluster node are, in most cases, multi-core Chip Multiprocessor (CMP). Although in the general case, the RAM in the cluster is distributed, the memory is common to all processors of one processor node of the cluster, which significantly improves the efficiency of data exchange between processes during parallel calculations within one node. It is clear that in order to ensure the coordinated operation of the entire cluster, special software tools are required that are installed on the cluster nodes and provide performance monitoring, problem diagnosis, and automation of resource management on all available cluster nodes. Cluster deployment and scaling are currently very often performed remotely with the help of cloud platform services [7-8]. (Amazon, Google, Azure)

Special software tools are required to ensure the coordinated operation of the entire cluster. These tools are installed on the cluster's main node and provide performance monitoring, problem diagnostics, and automated resource management for all accessible nodes in the cluster. Cluster creation and management are currently mostly conducted remotely using virtual machines on cloud platforms [7-8] (Amazon, Google, Azure).

However, deploying a software service in a cluster is not a sufficient condition to ensure its efficiency [1-5]. To fully utilize the resources of a parallel distributed system, it needs to be designed using multithreading principles [23-25]. The use of threads allows tasks to be executed concurrently. However, in practice, there are often cases of service performance degradation in multithreaded environments. This is primarily due to peculiarities in working with threads, including their creation, maintenance, and management. Therefore, this paper also examines and analyzes the key aspects of designing software service architectures that efficiently use threads for parallel computations, aiming to improve service performance and maximize resource utilization within the cluster

## MATERIALS AND METHODS

The paper involves theoretical research on designing features of distributed software systems architecture, based on the analysis and comparison of facts obtained from scientific sources and the practical experience acquired by the author as well.

## RESULTS AND DISCUSSION

Today, a primary tool for deployment, scaling, and management of software applications is the open-source software platform Kubernetes [9, 10] developed by Google. Kubernetes cluster deployments are mostly performed remotely on virtual clusters hosted on cloud platforms within their data centers (data centers). Since Kubernetes has become a standard for managing software applications in a cluster, let's look at the key architectural features of software services design related to using Kubernetes.

The most important feature of Kubernetes is that it can only manage containerized software applications where the application and everything needed for its operation (code, libraries, configuration files, and even the operating system) are packaged into a container [9, 10]. Currently, Docker-based containers are the most widely used platform for containerization [11, 12]. Docker containers utilize kernel-level virtualization, functioning as independent processes with their own address space. Unlike virtual machines, containers share a single operating system and its kernel while loading only the binaries and libraries required to run the application. This eliminates the need to install separate operating systems—multiple containers can operate within the same system. This approach saves overhead costs associated with virtual hardware emulation and launching full-fledged operating system instances, making containers lightweight, fast, and cost-effective. Everything inside the container is isolated from the outside world, enabling the simultaneous operation of multiple containers on a single system without concern that they could influence each other [11, 12].

Thus, to effectively scale a software application within a Kubernetes-managed cluster [9, 10], it must be designed as separate, loosely coupled software components that can be easily deployed in containers. This type of architecture is called microservices architecture [13-15], and its components are referred to as microservices. Splitting an application into individual microservices is typically based on specific aspects (models) of the domain. For instance, in the context of the financial domain, one microservice might handle user authentication and authorization, another might process payments, a third might manage documents, and so on. This approach is known as Domain-Driven Design (DDD) [13-15].

Each microservice should control access only to the data it is responsible for. No other microservices should have direct access to databases managed by other microservices such data should only be accessible through the API of the respective service. This ensures that each microservice controls the consistency and structure of the database. It manages and allows the use of the most appropriate database management system for its specific needs without being constrained by overall system requirements. For example, one microservice might use a normalized SQL database while another could benefit from a NoSQL database [13-15].

When implementing microservices architecture, interaction mechanisms between microservices must be carefully designed. The simplest and most general form of microservice communication is message-driven communication, where microservices exchange messages according to a specific protocol. This technique is called Message-Driven Architecture (MDA). In this case, one microservice sends a message (request), other processes it, and then sends a response message back. Here, the requesting service is referred to as the client and the responding one as the server. In high-load systems, communication may be event-driven instead, relying on distributed message queues. This communication approach is called Event-Driven Architecture (EDA) [19].

The rules governing the exchange of messages in the client-server model are encapsulated in the Representational State Transfer architectural style (REST API) [16-18]. The essence of the REST API architectural style is that the microservice that processes messages should provide a unified interface for identifying resources, should not cache data from previous requests, and each request should contain all the necessary information for its processing. Client-server interfaces (protocols) facilitate message transmission, with the most common protocols being HTTP(S), gRPC(S), and WebSockets. Resource data is exchanged in specified formats determined by the protocol. Popular formats include JSON, XML, HTML, YAML, PlainText, and FormData. Each request processing microservice follows a multi-layer architecture, where upper-layer components interact only with directly lower-layer components. A prominent example of this architecture is the Model-View-Controller (MVC) pattern [23]. The model defines the business logic corresponding to the domain-specific model. The view represents the layer that formats data for the user. The controller manages system interaction and modifies the model in response to incoming requests. In practice, microservices also have a persistence layer to ensure data consistency within the database and provide access to it.

#### **Synchronous and asynchronous architecture of distributed software systems.**

Nowadays, REST API are commonly implemented in synchronous communication models where a client sends a request and receives a response within the same connection. In synchronous architectures, each client request is handled by allocating a dedicated thread, as is common with traditional web servers. This architecture works well for scaling services under varying loads by dynamically increasing or decreasing the number of threads in the service. However, thread creation [25] involves not only generating an object within the user space of a process but also invoking low-level operating system APIs and kernel-level instructions for allocating and initializing hardware resources, such as stack memory and task scheduling. Consequently, this process is not instantaneous, and its latency becomes more noticeable under high-load conditions. To mitigate the negative impact of dynamic thread creation during service operation, it is advisable to pre-create an optimal number of

threads upon program startup and reuse them during execution. Threads are added to a thread pool for efficient reuse, and the optimal number of threads can only be determined experimentally through performance testing.

In cases where computational tasks are resource-intensive, parallel algorithms can improve execution. Tasks are divided into independent subtasks processed concurrently, and their results are merged afterward. However, this approach works well only for additive tasks; tasks dependent on execution order may suffer degraded performance due to synchronization and blockage of threads.

Despite its benefits, synchronous REST API architecture still employs blocking sockets, limiting the maximum number of simultaneous requests to the number of available threads. For higher scalability asynchronous architecture is preferred. Here, a single thread, which is known as the EVENT LOOP, handles all client requests using non-blocking sockets (NIO). Request processing occurs asynchronously within a thread pool, where callback functions or pipelines execute upon receiving results from databases or external services. Using asynchronous APIs, clients are notified of response completion through additional channels like email or SMS.

Another microservices interaction pattern is the Publisher-Subscriber architecture [23], which operates using distributed data queues [20-22] (e.g., Kafka, RabbitMQ, JMS, Google Pub/Sub). Changes in the state of one microservice generate an event that is published to the queue in a predefined topic. Other subscribed microservices consume and process the events, enabling a fully asynchronous Event-Driven Architecture (EDA) [19]. Such systems can be used for ensuring data consistency across distributed databases (e.g., using the SAGA pattern) [24].

**Deploying distributed software systems.** While microservices architecture offers considerable advantages, challenges must also be addressed. Deployment and scaling processes must be carefully described for each microservice to enable load balancing within a Kubernetes cluster [10]. Continuous Integration/Continuous Delivery/Continuous Deployment (CI/CD pipelines) are defined based on organizational requirements and technical specifications. Popular tools include Jenkins pipelines [26] or event-based pipelines (e.g., GitHub and GitLab CI/CD) [27]. Complexity management is another concern, particularly with large numbers of microservices. Furthermore, security and data integrity require additional attention.

Monitoring and logging in distributed environments present unique challenges, necessitating centralized analysis and logging tools. Practical implementations often involve the ELK stack [28]: Elasticsearch, Logstash, and Kibana. Logstash collects and transforms data, Elasticsearch indexes and analyzes it while providing search capabilities, and Kibana visualizes the results.

**Technologies for designing distributed software systems.** Currently, either the JAX-RS specification, which is part of the Jakarta Enterprise Edition (Java EE) standard, or the Spring framework is used to create enterprise-level RESTful microservices.

JAX-RS (Java API for RESTful Web Services) is a Java specification that provides a standardized way to create RESTful APIs, which uses annotations to simplify the process. The `@Path` annotation is used to define REST endpoints. The `@GET`, `@POST`, and `@PUT` annotations are used to define the methods that will be used to process the request - reading a resource, adding a new resource, and updating a resource, respectively. The `@Produces`, `@Consumes` annotations are used to define the data format. However, JAX-RS is only a specification, so its use requires an implementation. The main implementations of JAX-RS that work with data via the HTTP protocol and are used in practice are the Jersey, RESTEasy, and Apache CXF frameworks.

The Spring Web framework, which is part of the Spring platform, should be highlighted separately. Spring Web provides its own set of annotations for creating RESTful APIs, for example, `@RestController`, `@RequestMapping`, while immediately providing mechanisms for their processing. Therefore, in this case, we get a full-fledged framework for creating



RESTful APIs, which also works based on the HTTP protocol but does not implement the JAX-RS standard.

Standard implementations of JAX-RS and Spring Web frameworks are built on classic web servers (Tomcat), which use a synchronous (blocking) architecture for processing requests. The synchronous architecture has its limitations associated with blocking threads during the processing of client requests, and therefore, all these limitations are inherited by the frameworks. Such architectures are very common in the financial or health insurance sectors, where data processing includes a lot of data validation, document signing, and report generation. All these data processing stages are usually performed sequentially and technically involve working with different databases and integrating with external services. Typical technical tasks that are often encountered in practice in this area:

- a) The task involves executing a database query that requires reading or updating a large data set;
- b) The task involves executing several database queries within a single transaction;
- c) Data for calculations must be obtained from several different sources using external service calls, which are often also synchronous;

This type of task is well divided into several parts that can be executed in parallel in separate threads. After the data is received, it can be processed asynchronously using callback code. In practice, in the Java world, `CompletableFuture` is used for this purpose.

If data processing is long-term, even using parallel threads, then we can provide an API to check the execution status. In this case, the software client periodically polls the server about the operation execution status.

If the task is additive and all the necessary initial data is present, but we need to speed up their processing, then in this case we can also parallelize the task into independent parts, execute them in a parallel thread pool, and then combine the results. `ForkJoin` thread pool is effectively used for this purpose in Java. However, it should be emphasized once again that performance will improve only if the result does not depend on the order of execution of its parts. In this case, the main thread is blocked until we get the intermediate results necessary for the combination.

If, according to technical requirements, it is necessary to accept as many client requests as possible or to carry out continuous data transfer in real time, then in this case, in practice, an asynchronous (non-blocking) Restful API is used. Asynchronous approaches are very often used in online games, the Internet of Things (continuous data collection from various sensors), and the creation of video conferencing platforms. An asynchronous API can be built using the Spring Web Flux framework or using distributed data queues.

If we talk about the Spring Web Flux framework, then its work is built only on the basis of the non-blocking Netty server. A feature of the asynchronous architecture, which is built on a non-blocking server, is the use completely non-blocking API in all parts of the software microservice to avoid blocking the main thread. Otherwise, the work of such a service under some conditions may simply stop.

In the case of distributed data queues, microservices can be built on both blocking and non-blocking servers. In practice, ready-made solutions provided by leading cloud platform vendors, such as Google Pub/Sub, are usually used.

## CONCLUSION

As a result of the study, it was found that a feature designing of distributed software systems is designing microservice architecture. It was analyzed that microservice architecture can be built based on REST architecture or event-driven architecture.

Synchronous (blocking) REST architecture shows good performance but has limitations associated with blocking threads until the results of client requests are received. Therefore, it is suitable for areas where there is no high load on services, in particular for the financial and health insurance sectors.



Asynchronous (non-blocking) REST architecture and event-driven architecture (EDA) are used in areas where continuous data transfer in real time is required (online games, Internet of Things, streaming video conferences). Event-driven architecture is also used in systems with microservice architecture in which the state of one microservice depends on the state of another. The asynchronous model allows for a larger number of requests than the synchronous model but requires a fully asynchronous API when working with external services and is also more complex when debugging the service and fixing errors.

The study also found that architecture of distributed parallel systems should be built using parallel threads. The use of threads makes it possible to increase the performance of software systems but only if a parallel not a concurrent multithreading mode is used by parallel algorithm. The concurrent mode threads operation is advisable to use only in case of blocking operations when some threads are in standby mode. In general, the number of active threads in parallel mode should coincide with the number of available processors (processor cores).

The results obtained can be used in the practical design of microservice architectures and microservice integration. Also, the results of the study can be used in further scientific research related to the use of software system protection mechanisms discussed in this article.

## COMPLIANCE WITH ETHICAL STANDARDS

The author declare that he have no competing interests.

## AUTHOR CONTRIBUTIONS

Conceptualization, [I.R.]; investigation, [I.R.]; writing – original draft preparation, [I.R.]; writing – review and editing, [I.R.];

The author has read and agreed with the published version of the article.

## REFERENCES

- [1] Burns, B. (2025). Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Systems Using Kubernetes: O'Reilly. <https://www.oreilly.com/library/view/designing-distributed-systems/9781098156343/>
- [2] Adkins, H., Beyer, B., Blankinship, P., Lewandowski, P., Oprea, A., Stubblefield, A. (2020). Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems: O'reilly. <https://www.oreilly.com/library/view/building-secure-and/9781492083115/>
- [3] Gorton, I. (2022). Foundations of Scalable Systems: Designing Distributed Architectures 1st Edition: O'Reilly. <https://www.oreilly.com/library/view/foundations-of-scalable/9781098106058/>
- [4] Tanenbaum, A. S., Steen, M. (2006). Distributed Systems: Principles and Paradigms: Pearson. <https://pdf-up.com/download/distributed-systems-principles-and-paradigms-andrew-s-tanenbaum-4951969>
- [5] Vitillo, R. (2021). Understanding Distributed Systems: What every developer should know about large distributed applications. <https://cdn.bookeekey.app/files/pdf/book/en/understanding-distributed-systems.pdf>
- [6] Herlihy, M., Shavit, N. (2012). The Art of Programming Multiprocessor: Elsevier. <https://www.educate.elsevier.com/book/details/9780123973375>
- [7] Sosinsky, B. (2011). Cloud Computing Bible: Wiley Publishing. DOI: [10.1002/9781118255674](https://doi.org/10.1002/9781118255674)
- [8] Buyya, R., Broberg, J., Goscinski, A.M. (2011). Cloud Computing: Principles and Paradigms: Wiley Publishing. DOI: [10.1002/9780470940105](https://doi.org/10.1002/9780470940105)
- [9] Luksa, M. (2023). Kubernetes in Action: Manning. <https://www.scribd.com/document/659803971/Kubernetes-in-Action-Second-Edition-MEAP-V15>

- [10] Boorshtein, M., Surovich, S. (2024). Kubernetes – An Enterprise Guide: Master containerized application deployments, integrate enterprise systems, and achieve scalability: Packt Publishing.  
[https://www.packtpub.com/en-SK/product/kubernetes-an-enterprise-guide-9781835081754?srsid=AfmBOor\\_Cra9iEVmhmtw\\_DirzU\\_mzs3KGN4Cfqff4tglkvTX6Zf9dHBk](https://www.packtpub.com/en-SK/product/kubernetes-an-enterprise-guide-9781835081754?srsid=AfmBOor_Cra9iEVmhmtw_DirzU_mzs3KGN4Cfqff4tglkvTX6Zf9dHBk)
- [11] Nickoloff, J., Kuenzli, S. (2019). Docker in Action: Manning.  
<https://www.manning.com/books/docker-in-action-second-edition>
- [12] Poulton, N. (2023). Docker Deep Dive: Packt Publishing.  
<https://cdn.bookekey.app/files/pdf/book/en/docker-deep-dive.pdf>
- [13] Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems: O'Reilly.  
<https://www.oreilly.com/library/view/building-microservices/9781491950340/>
- [14] Bruce, M., Pereira, P.A. (2019). Microservices in Action: Manning.  
<https://www.oreilly.com/library/view/microservices-in-action/9781617294457/>
- [15] Fowler, S. J. (2016). Production-Ready Microservices: Building Standardized Systems Across Engineering an Organization: O'Reilly.  
<https://www.oreilly.com/library/view/production-ready-microservices/9781491965962/>
- [16] Biehl, M. (2016). RESTful API Design: Createspace Independent Publishing Platform.  
[https://books.google.com.ua/books?id=DYC3DwAAQBAJ&printsec=frontcover&hl=uk&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ua/books?id=DYC3DwAAQBAJ&printsec=frontcover&hl=uk&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- [17] Varanasi, B., Bartkov, M. (2022). Spring REST. Building Java Microservices and Cloud Applications: Apress.  
<https://www.oreilly.com/library/view/spring-rest-building/9781484274774/>
- [18] Amundsen, M. (2022). RESTful Web API Patterns and Practices Cookbook: Connecting and Orchestrating Microservices and Distributed Data: O'Reilly.  
<https://www.oreilly.com/library/view/restful-web-api/9781098106737/>
- [19] Bellemare, A. (2023). Building an Event-Driven Data Mesh: Patterns for Designing & Building Event-Driven Architecture s: O'Reilly.  
<https://www.oreilly.com/library/view/building-an-event-driven/9781098127596/>
- [20] Narkhede, N., Shapira, G., Palino, T. (2017). Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale: O'Reilly.  
<https://www.oreilly.com/library/view/kafka-the-definitive/9781491936153/>
- [21] Videla, A., Williams, J.J.W. (2012). RabbitMQ in Action Distributed Messaging for Everyone: Manning.  
<https://classpages.cselabs.umn.edu/Spring-2018/csci8980/Papers/PublishSubscribe/RabbitMQinAction.pdf>
- [22] Richards, M., Monson-Haefel, R., (2009). Chappell D. A. Java Message Service: O'Reilly. <https://www.oreilly.com/library/view/java-message-service/9780596802264/>
- [23] Fowler, M. (2003). Patterns of Enterprise Application Architecture: Longman (Pearson Education).  
<https://ptgmedia.pearsoncmg.com/images/9780321127426/samplepages/0321127420.pdf>
- [24] Joshi, U. (2023). Patterns of Distributed Systems: Addison-Wesley Professional.  
<https://www.oreilly.com/library/view/patterns-of-distributed/9780138222246/>
- [25] Goetz, B. Bloch, J., Peierls, T., Bowbeer, J., Holmes, D., Lea, D. (2006). Java Concurrency in Practice: Longman (Pearson Education). <https://ptgmedia.pearsoncmg.com/images/9780321349606/samplepages/9780321349606.pdf>
- [26] Leszko, R. (2017). Continuous Delivery with Docker and Jenkins: Packt Publishing.  
<https://www.packtpub.com/en-us/product/continuous-delivery-with-docker-and-jenkins-3rd-edition-9781803237480>

- [27] Kaufmann, M., Bos, R., Vries, M. (2024). GitHub Actions in Action Continuous integration and delivery for DevOps: Manning.  
<https://www.oreilly.com/library/view/github-actions-in/9781633437302/>
- [28] Chhajed, S. (2015). Learning ELK Stack: Packt Publishing  
<https://www.packtpub.com/en-ch/product/learning-elk-stack-9781785887154>

## ОСОБЛИВОСТІ ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНИХ РОЗПОДІЛЕНИХ СИСТЕМ

**Іван Ровецький**

*Львівський державний університет безпеки життєдіяльності,  
 вул. Клепарівська 35, 79007, Львів, Україна*

### АНОТАЦІЯ

**Вступ.** На даний час актуальною проблемою проектування розподілених програмних систем є те, що вони мають стабільно працювати в умовах високого навантаження. Для того, щоб забезпечити високу доступність та стійкість високонавантажених програмних сервісів, їх розгортають у керованих мультипроцесорних розподілених системах (кластерах). Такі ресурси є високо вартісними, тому на практиці, найчастіше, використовують різноманітні хмарні платформи (Google Cloud, Amazon, Azure та ін.), які надають ці ресурси віддалено. Для того, щоб використовувати розподілені апаратні ресурси в повному обсязі, програмні сервіси необхідно проектувати, використовуючи спеціальні архітектурні рішення. Тому мета даного дослідження полягає у теоретичному аналізі архітектурних рішень та особливостей їхнього використання під час проектування програмних сервісів розподілених систем.

**Матеріали та методи.** У роботі виконано теоретичне дослідження особливостей проектування архітектури розподілених програмних систем, яке базується на аналізі та порівнянні фактів, отриманих з наукових джерел, а також у результаті отриманого практичного досвіду автора.

**Результати.** У роботі показано, що основним програмним рішенням для розгортання програмних систем у паралельних розподілених системах є платформа Kubernetes. У результаті роботи встановлено, що мікросервісна архітектура є оптимальним архітектурним рішенням для проектування програмних сервісів розподілених систем, з огляду на специфіку розгортання під керуванням Kubernetes.

**Висновки.** Для ефективного масштабування розподіленої програмної системи необхідно використовувати багатопотокове проектування. Однак, підвищення продуктивності відбувається тільки за умови, коли у алгоритмі використовується паралельний, а не конкурентний режим багатопоточності. Конкурентний режим роботи потоків доцільно використовувати тільки у випадку блокуючих операцій. Синхронна архітектура має обмеження, пов'язані з блокуванням потоків, доки не буде отримано результати клієнтських запитів. Асинхронна модель дає можливість прийняти більшу кількість запитів, однак вимагає повністю асинхронного API під час роботи із зовнішніми сервісами, а також є складнішою під час відлагодження сервісу та виправлення помилок.

**Ключові слова:** розподілені системи, кластери, хмарні платформи, мікросервіси, мікросервісна архітектура, багатопоточність.

Received / Одержано  
22 May, 2025

Revised / Доопрацьовано  
22 September, 2025

Accepted / Прийнято  
26 September, 2025

Published / Опубліковано  
31 October, 2025

UDC: 004.7

## INTELLIGENT METHODS FOR DATA ANALYSIS IN INFORMATION AND COMMUNICATION SYSTEMS MONITORING PROCESSES

Andrii Senyk<sup>1</sup><sup>\*</sup>, Volodymyr Kotsun<sup>2</sup>, Bohdan Penyukh<sup>3</sup>,  
Bohdan Tsybulyak<sup>4</sup>

<sup>1</sup>Department of Telecommunication, Lviv Polytechnic National University,  
12 Stepan Bandera Str., Lviv, 79013, Ukraine

<sup>2</sup>Department of Computer Science and Software Engineering,  
Private Higher Education Institution "European University",  
16V Acad. Vernadsky Blvd., Kyiv, 03115, Ukraine

<sup>3</sup>Department of Physical and Biomedical Electronics,

<sup>4</sup>Department of System Design,  
Ivan Franko National University of Lviv,  
50 Drahomanova str., Lviv, 79005, Ukraine

Senyk, A., Kotsun, V., Penyukh, B., Tsybulyak, B. (2025). Intelligent Methods for Data Analysis in Information and Communication Systems Monitoring Processes. *Electronics and Information Technologies*, 31, 53–60. <https://doi.org/10.30970/eli.31.5>

### ABSTRACT

**Background.** In modern monitoring of information and communication systems (ICS), a key challenge remains the timely detection of anomalies while maintaining a low false positive rate. Classical machine learning or deep learning methods often show a trade-off between high precision and the ability to detect most anomalies, limiting their efficiency in dynamic network environments.

**Materials and Methods.** This study proposes the Hybrid Adaptive Monitoring Method with Multi-level Anomaly Validation (HAM-MAV), which combines a deep autoencoder for anomaly detection (unsupervised) with a Random Forest classifier (supervised) and an adaptive threshold mechanism. In the first stage, the autoencoder identifies suspicious samples based on reconstruction error. These samples are then refined by the Random Forest, reducing false positives. The threshold is updated dynamically according to the statistics of the latest observation window. The experiments used the NSL-KDD (Network Security Laboratory – Knowledge Discovery in Databases) dataset with preprocessing steps including normalization, one-hot encoding, and feature selection based on correlation criteria.

**Results and Discussion.** Experimental results show that HAM-MAV achieves Precision of 96.92%, Recall of 62.67%, F1-score of 76.12%, and ROC-AUC (Receiver Operating Characteristic – Area Under Curve) of 0.8003, outperforming Autoencoder, Random Forest, and Isolation Forest in most metrics. The method reduces false positives while improving anomaly detection capability, maintaining a fast processing time. HAM-MAV's key advantage is its balanced performance between precision and recall, which is critical for continuous ICS monitoring.

**Conclusion.** HAM-MAV provides an optimal combination of precision, recall, and execution speed, outperforming traditional methods in real-time conditions. Its architecture allows effective operation in environments with changing traffic characteristics, making it a promising approach for cybersecurity applications, particularly in automated intrusion detection systems.

**Keywords:** anomaly detection, deep learning, random forest, adaptive threshold, intrusion detection, NSL-KDD.



© 2025 Andrii Senyk et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

## INTRODUCTION

Monitoring of information and communication systems (ICS) is a crucial tool for ensuring their reliability, security, and operational efficiency. Given the rapid growth in the volume of data transmitted and processed today, as well as the complexity of ICS architectures, continuous monitoring of transactions and timely detection of any deviations are of paramount importance. Monitoring has become a key component for maintaining service quality, managing risks, and protecting against cyberattacks [1-4].

This topic is also important due to the deep integration of ICS systems into all spheres of life, including business, government, healthcare, and energy. Any disruptions in their operation can lead to significant economic losses and even provoke a crisis. Therefore, effective monitoring methods are crucial for the stable operation of critical infrastructure and maintaining information security. Traditional methods of monitoring data analysis, based on fixed thresholds and simple processing algorithms, have several shortcomings. In modern conditions, the application of intelligent data analysis methods, in particular machine learning and artificial intelligence algorithms, is becoming increasingly important due to their ability to detect complex dependencies and patterns in large volumes of information. Similar hybrid approaches are actively applied in other areas of digital analytics. In particular, [5] presents an example of an effective combination of deep learning methods and classical classification for detecting synthetic content in social media. This confirms the universality of such architectures for anomaly detection tasks and highlights the relevance of the HAM-MAV approach proposed in this study for monitoring information and communication systems.

In the study, HAM-MAV was developed, which combines the capabilities of deep neural networks and classical machine learning algorithms. In particular, at the first stage, an autoencoder was used to detect potentially anomalous samples, which operates in unsupervised mode and estimates the degree of deviation of input data from normal behavior using the reconstruction error. At the second stage, the selected samples are additionally analyzed using the Random Forest model, which allows for significantly reducing the number of false positives. To increase the flexibility of operation, the system is equipped with an adaptive mechanism for changing threshold values, which is updated depending on the statistical characteristics of the last observation window. A comparative analysis was carried out with three well-known methods (Autoencoder, Random Forest, and Isolation Forest) by key model evaluation metrics (Precision, Recall, F1-score, ROC-AUC), as well as by execution time. The experimental results showed the superiority of HAM-MAV in most indicators, which confirms the feasibility of its use in real-time monitoring systems of information and communication systems.

## MATERIALS AND METHODS

Traditional methods for analyzing monitoring data typically rely on thresholds and simple statistical measures such as the mean or variance. While these methods work well in stable environments, they have limited ability to detect complex anomalies or emerging threats. Intelligent data analysis methods, including machine learning (ML) and deep learning (DL), are increasingly being used in monitoring systems. Classic machine learning algorithms that have demonstrated high accuracy in anomaly detection include random forests, support vector machines (SVM), and k-nearest neighbors (k-NN). These algorithms have been successfully used to classify cyberattacks and analyze the behavior of system processes. Deep neural networks (DNNs), including long short-term memory networks (LSTMs) and autoencoders, have shown themselves well in time series processing and anomaly detection in network traffic.

The paper proposes a Hybrid Adaptive Monitoring Method with Multi-level Anomaly Validation (HAM-MAV). This method combines deep anomaly detection using autoencoders with online classification using Random Forest and uses a multi-stage

validation mechanism to reduce the number of false positives. Its key feature is a multi-stage decision filter. First, an autoencoder (without a teacher) is used to detect anomalies based on the deviation between the reconstructed and true data [6-9].

Potential anomalies are then fed to a Random Forest, pre-trained on labeled data from the same dataset and trained with a teacher, to improve classification performance. In addition, an adaptive threshold is used that changes depending on the statistics of the last observation window (e.g., traffic flow in one minute). This reduces the sensitivity of the system to "noisy" anomalies. Method architecture:

1. Data pre-processing
  - Normalization of features.
  - Unimportant or highly correlated features are removed.
  - Categorical values are converted to numerical values (single coding).
2. Autoencoder for anomaly detection
  - The model learns to reconstruct normal (unsupervised) samples.
  - The mean squared error (MSE) is calculated for each sample.
  - Samples with an error exceeding the adaptive threshold are classified as "suspicious".
3. Random Forest for clarification
  - Receives suspicious samples as input.
  - Operates in a controlled mode, using labeled data from a dataset.
  - Returns the final class: "abnormal" or "normal".
4. Adaptive threshold
  - Dynamically calculates the moving average and standard deviation of the reconstruction error.
  - The threshold is defined as  $\text{mean\_error} + k * \text{std\_error}$ , where  $k$  is configurable.

The proposed method provides a lower false positive rate due to two-stage validation (autoencoder + Random Forest). Adapts to environmental changes: the threshold is automatically updated based on the latest data. Works with both labeled and unlabeled data: the first stage is unsupervised learning, and the second stage is supervised learning.

### Research overview

The section details the development, implementation, and testing of HAM-MAV. This approach combines the autoencoder for anomaly detection with a Random Forest algorithm for classification optimization using an adaptive thresholding mechanism. The study was divided into three main phases: data preparation, model construction and tuning, and experimental testing with results compared to existing methods.

#### Data usage

To test this approach, the open NSL-KDD dataset (Tavallaee et al., 2009) was used, freely available through the UCI Machine Learning Repository. This dataset contains 125,973 training data samples and 22,544 test data samples, representing typical network traffic and various attack types (DoS, Probe, R2L, and U2R). Each record is described by 41 attributes, including both numeric attributes (e.g., number of bytes in a packet) and categorical attributes (e.g., protocol and service type). NSL-KDD is a classical but not the most recent dataset. Its use in this study is justified by the fact that NSL-KDD is one of the most widely adopted benchmark datasets for comparing the performance of intrusion detection systems. This enables the results of HAM-MAV to be compared with a large number of previous studies, ensuring the objectivity and validity of the conclusions. It should also be noted that, for the initial stage of evaluating a hybrid approach, it is important to work with a dataset that has a balanced class structure and clearly labeled attack types (DoS, Probe, R2L, U2R). Nevertheless, in future research,



the HAM-MAV approach is planned to be tested on more recent and large-scale datasets, such as CICIDS2017, UNSW-NB15, and TON\_IoT, which will allow the method to be evaluated in environments with a broader spectrum of modern attacks.

Pre-processing includes:

- Normalization of numerical features to the range [0, 1] using the min-max method.
- One-Hot Encoding for categorical variables.
- Removal of insignificant features with a correlation coefficient with the target variable less than 0.05 (Pearson's correlation coefficient for numerical variables and chi-square test for categorical variables).

After data preparation, the autoencoder is trained using only normal samples from the training set [7-11].

### Architecture of the HAM-MAV

#### Level 1 – autoencoder

The autoencoder is designed with three hidden layers, with 64, 32, and 64 neurons, respectively. The hidden layers use the ReLU activation function, and the output layer uses the Sigmoid function. The optimization is performed using the Adam algorithm with a learning rate of 0.001 and a MSE loss function. The training is performed for 50 epochs with a batch size of 256. The reconstruction error is calculated as the mean square error between the input value and the reconstructed feature value. The threshold for detecting suspicious samples is calculated adaptively:

$$threshold = \mu_{err} + k \cdot \sigma_{err}, \quad (1)$$

where  $\mu_{err}$  represents the average reconstruction error within a sliding window of 500 samples,  $\sigma_{err}$  is the standard deviation, and  $k$  is an empirically chosen parameter ( $k = 1.5$  in the experiments).

#### Level 2 – Random Forest

The Random Forest algorithm was used to improve the classification of suspicious samples. The number of trees is 200, the maximum depth is 20, and the splitting criterion is Gini impurity. The `class_weight='balanced'` parameter is used to balance the classes. The training data contains all classes.

#### Adaptive threshold mechanism

During the test run, the autoencoder threshold was updated every 500 new examples to reduce the impact of time-varying data distribution (conceptual bias).

For each method, Precision, Recall, F1-score, and ROC-AUC were calculated. In addition, the False Alarm Rate (FAR) was analyzed for 1000 test examples.

## RESULTS AND DISCUSSION

#### Statistical analysis

All experiments were performed in Google Colab using TensorFlow 2.12, Scikit-learn 1.3, Pandas 2.0, and NumPy 1.25. Statistical analysis was performed using the SciPy 1.11 Python package. For each performance measure, the mean and standard deviation were calculated based on 5-fold cross-validation (stratified). Paired t-tests were used to test the statistical significance of differences between methods ( $p < 0.05$  was considered statistically significant). The experimental results for different models are presented in **Table 1**.



### Precision

The Precision metric shows what proportion of predicted anomalies are actually anomalies. Higher precision means fewer false positives.

Conclusion on the effectiveness of HAM-MAV: Precision of HAM-MAV is very high and comparable to that of Autoencoder. That is, the model rarely mistaken, marking normal data as anomalous. In terms of accuracy, HAM-MAV is practically at the level of the best models, and it can be considered effective in avoiding false positives.

**Table 1. Experiment results**

Model	Precision (%)	Recall (%)	F1-score (%)	ROC-AUC	Execution Time (s)
Autoencoder	97.40	8.58	15.77	0.5414	24.94
Random Forest	96.66	61.19	74.94	0.7921	1.88
Isolation Forest	94.55	31.20	46.92	0.6442	0.41
HAM-MAV	96.92	62.67	76.12	0.8003	1.52

### Recall

Metric explanation: Recall shows what proportion of real anomalies were detected. A higher value means fewer anomalies were missed.

HAM-MAV has much higher completeness compared to Autoencoder and Isolation Forest, and slightly better than Random Forest. This means that the method is able to find most real anomalies. Overall conclusion: HAM-MAV has a good balance between accuracy and ability to find anomalies, which makes it reliable for detection.

### F1-score

The F1-score combines Precision and Recall into a single value, showing the balance between false positives and missed anomalies. Higher F1 means better balance. The F1 of HAM-MAV is the highest among the models, which means that the proposed method provides the best balance between precision and completeness. In terms of comprehensive anomaly detection efficiency, HAM-MAV appears to be the best model.

### ROC-AUC

ROC-AUC shows the ability of the model to distinguish between normal and abnormal data at any threshold. A higher score indicates better classification. HAM-MAV has the highest ROC-AUC among all models, confirming its ability to separate anomalies from normal data. Therefore, HAM-MAV demonstrates the best overall classification ability.

### Execution Time

The execution time of HAM-MAV is very low and comparable to Random Forest, significantly faster than Autoencoder, although slightly slower than Isolation Forest. Thus, HAM-MAV provides high performance with fast processing, making it practical for real-world applications.

Therefore, according to the conducted experiment, it can be concluded that HAM-MAV demonstrates the best balance between accuracy, completeness, and F1-score, has a high ability to distinguish anomalies (ROC-AUC), and works fast. This makes the proposed method effective and competitive compared to classical approaches.

### CONCLUSION

In the field of information and communication system monitoring, anomaly detection is a critical task, as unsuccessful attacks or excessive false positives can have serious

consequences. Current methods often suffer from poor accuracy and the ability to detect most real threats. Autoencoders achieve good results in reducing false positives, but they often miss a significant portion of attacks, while traditional methods (such as Random Forest or Isolation Forest) can have low accuracy or be sensitive to noise.

The paper proposes HAM-MAV, a hybrid method that combines an autoencoder for initial anomaly detection and a Random Forest for improved classification. The use of an adaptive threshold that varies depending on the statistics of the last observation window reduces the number of false positives and improves the system's robustness to data changes.

Experiments on the NSL-KDD dataset showed that HAM-MAV exhibits Precision of 96.92% (at the level of the best competitor), Recall of 62.67% (630% higher than Autoencoder and 101% higher than Isolation Forest), F1-score of 76.12% (382% higher than Autoencoder and 62% higher than Isolation Forest), and ROC-AUC of 0.8003 (47% higher than Autoencoder and 24% higher than Isolation Forest). Thus, the proposed method provides a better balance between precision and completeness than other considered approaches. In terms of execution time, HAM-MAV (1.52 s) significantly outperforms Autoencoder (~17 times faster) and runs faster than Random Forest (19%), although slightly slower than Isolation Forest. This makes the method suitable for processing large amounts of data in near real time, which is important for operational monitoring systems of ICS.

The main advantages of HAM-MAV are the ability to reduce the level of false positives, adaptability to traffic changes, and versatility in working with both labeled and unlabeled data. Thanks to multi-level validation of solutions, the method is suitable for critical systems where a balance is needed between threat sensitivity and stability of operation without overloading operators with false alarms. Future extensions are possible, such as the inclusion of deep learning models in the classification, the use of Bayesian methods to refine adaptive thresholds, and their application to large data streams. This will increase the detection efficiency and reduce the cost of computing resources, while maintaining high-quality threat detection in dynamic ICS.

## ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and publication of this article.

## CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any.

## AUTHOR CONTRIBUTIONS

Conceptualization, [A.S., V.K.]; methodology, [A.S., B.P.]; validation, [A.S., B.T.]; formal analysis, [A.S., V.K.]; investigation, [A.S., B.P.]; resources, [A.S., B.T.]; data curation, [A.S., V.K.]; writing – original draft preparation, [A.S.]; writing – review and editing, [A.S., B.P.]; visualization, [A.S., B.T.]; supervision, [A.S.]; project administration, [A.S.]; funding acquisition, [A.S.].

All authors have read and agreed to the published version of the manuscript.

## REFERENCES

- [1] Senyk, A., Klymash, M., Pyrih, Y., Tsybulyak, B., Penyukh, B., & Shuvar, R. (2024). Improving the cybersecurity monitoring algorithms efficiency using neural networks. In 2024 IEEE 17th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET) (pp. 428–431). IEEE. <https://doi.org/10.1109/TCSET64720.2024.10755520>.

- 
- [2] Klymash, M., Senyk, A., & Pyrih, Y. (2024). Investigation of a context-sensitive cyber security monitoring algorithm based on recurrent neural networks. *ICTEE*, 4(1), 1–9. <https://doi.org/10.23939/ictee2024.01.001>.
- [3] Feng, S., Yang, Z., Huang, M., & Wu, Y. (2021). Big data analysis of intellectual property service agencies. In 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI) (pp. 326–330). IEEE. <https://doi.org/10.1109/PRAI53619.2021.9551058>.
- [4] Cai, J., Xie, L., Yao, S., & Gao, Y. (2025). Algorithm application and optimization in intellectual property management. In 2025 International Conference on Digital Analysis and Processing, Intelligent Computation (DAPIC) (pp. 458–463). IEEE. <https://doi.org/10.1109/DAPIC66097.2025.00091>.
- [5] Y, A. F., Sundaram, A., & Ruby Helen, F. (2025). Analyzing social media data misuse and intellectual property rights: A dual legal-empirical analysis approach in the digital landscape. In 2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI) (pp. 1803–1810). IEEE. <https://doi.org/10.1109/ICMCSI64620.2025.10883512>.
- [6] Wu, B., Zheng, S., & Han, M. (2024). Innovation efficiency and influencing factors of high-tech industries: An analysis from the intellectual property perspective. In 2024 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 1480–1484). IEEE. <https://doi.org/10.1109/IEEM62345.2024.10857198>.
- [7] Xiang, D., & Wu, Y. (2022). Analysis and research of internet user behaviors under the context of big data. In 2022 International Conference on Big Data, Information and Computer Network (BDICN) (pp. 243–247). IEEE. <https://doi.org/10.1109/BDICN55575.2022.00054>.
- [8] Qiu, B., Liu, D., Cao, S., Mu, C., Yan, S., & Liu, Y. (2024). Risk analysis and protection suggestions for artificial intelligence data security. In 2024 IEEE 9th International Conference on Data Science in Cyberspace (DSC) (pp. 392–398). IEEE. <https://doi.org/10.1109/DSC63484.2024.00059>.
- [9] Wang, Y., Sun, J., Lu, X., Chen, C., & Yang, F. (2024). Research on data privacy calculation and data traceability technology for power monitoring system. In 2024 Asia-Pacific Conference on Software Engineering, Social Network Analysis and Intelligent Computing (SSAIC) (pp. 867–871). IEEE. <https://doi.org/10.1109/SSAIC61213.2024.00175>.
- [10] Zhang, L., Li, Y., Qiu, B., Zhang, J., & Liang, W. (2021). Design of communication power centralized remote monitoring system based on big data technology. In 2021 International Conference on Electronics, Circuits and Information Engineering (ECIE) (pp. 46–49). IEEE. <https://doi.org/10.1109/ECIE52353.2021.00017>.
- [11] Lai, S., Pan, Z., Ren, Q., Wang, P., Zhao, J., & Chen, H. (2024, October). IoT-Based Site Safety Monitoring and Early Warning System. In 2024 3rd International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI) (pp. 870–874). IEEE. <https://doi.org/10.1109/ICDACAI65086.2024.00164>.
-

## ДОСЛІДЖЕННЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ АНАЛІЗУ ДАНИХ У ПРОЦЕСАХ МОНІТОРИНГУ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ

Андрій Сенік<sup>1</sup>✉\*, Володимир Коцун<sup>2</sup>, Богдан Пенюх<sup>3</sup>,  
Богдан Цибуляк<sup>4</sup>

<sup>1</sup> Національний університет «Львівська політехніка»,  
вул. Степана Бандери, 12, Львів, 79013, Україна

<sup>2</sup> Приватний вищий навчальний заклад «Європейський університет»,  
бульв. Академіка Вернадського, 16в, Київ, 03115, Україна

<sup>3</sup> Кафедра фізичної та біомедичної електроніки,

<sup>4</sup> Кафедра системного проектування,  
Львівський національний університет імені Івана Франка,  
вул. Драгоманова 50, Львів, 79005, Україна

### АНОТАЦІЯ

**Вступ.** У сучасних системах моніторингу інформаційно-комунікаційних систем ключовим завданням є своєчасне виявлення аномалій при збереженні низького рівня хибнопозитивних спрацювань. Традиційні методи машинного та глибинного навчання часто виявляють компроміс між високою точністю та здатністю виявляти більшість відхилень, що знижує їхню ефективність у динамічних мережових умовах.

**Матеріали та методи.** Запропоновано гібридний адаптивний метод моніторингу з багаторівневою перевіркою аномалій (HAM-MAV), який поєднує автокодер для первинного виявлення відхилень із класифікатором на основі алгоритму випадкового лісу для їх уточнення та адаптивним механізмом коригування порогу. Автокодер визначає підозрілі зразки на основі похибки реконструкції, після чого їх аналізує випадковий ліс, що зменшує кількість помилкових спрацювань. Поріг оновлюється автоматично залежно від статистичних характеристик останнього вікна спостережень. Для перевірки використано відкритий набір даних виявлення знань у базах даних, до якого застосовано попередню обробку (нормалізацію, кодування категоріальних ознак та відбір ознак за кореляційними критеріями).

**Результати.** Отримані результати демонструють, що запропонований метод забезпечує високу точність (96,92%), прийнятну повноту (62,67%), збалансований F1-показник (76,12%) та значення ROC-AUC 0,8003. У порівнянні з відомими методами (автокодер, випадковий ліс, ізоляційний ліс) запропонований підхід зменшує кількість хибнопозитивних сигналів і водночас покращує здатність виявляти реальні відхилення при збереженні швидкості обробки даних (1.52 с).

**Висновки.** Метод HAM-MAV демонструє ефективне поєднання точності, повноти та швидкодії, перевершуючи класичні підходи в умовах реального часу. Його архітектура дозволяє адаптуватися до змін у мережевому трафіку, що робить цей метод перспективним для застосування у сфері кібербезпеки, зокрема в автоматизованих системах виявлення вторгнень. Водночас проведені експерименти підкреслюють баланс, досягнутий запропонованим методом між чутливістю до виявлення та стабільністю роботи. Цей баланс особливо важливий для критично важливих інфраструктур, де як пропущені загрози, так і надмірні хибні спрацювання можуть призвести до серйозних наслідків.

**Ключові слова:** виявлення аномалій, глибоке навчання, випадковий ліс, адаптивний поріг, виявлення вторгнень, відкритий набір даних

UDC: 621.382

## MODULAR APPROACH TO BUILDING A HARDWARE-SOFTWARE PLATFORM FOR SMART HOME AUTOMATION: FROM SIMPLE RULES TO INTELLIGENT SCENARIOS

Olha Shymchyshyn<sup>1</sup><sup>\*</sup>, Maryan Shymchyshyn<sup>2</sup>,  
Vladyslav Serhiienko<sup>1</sup>

<sup>1</sup> Lviv Polytechnic National University, 12 Stepan Bandera St., Lviv, Ukraine

<sup>2</sup> European University, 5 Kushevycha str., Lviv, Ukraine

<sup>\*</sup>Corresponding author

Shymchyshyn, O., Shymchyshyn, M., & Serhiienko, V. (2025). Modular approach to building a hardware-software platform for smart home automation: from simple rules to intelligent scenarios. *Electronics and Information Technologies*, 31, 61–70. <https://doi.org/10.30970/eli.31.6>

### ABSTRACT

**Background.** This article addresses the challenge of creating flexible, scalable, and user-adaptive automation systems that can evolve in response to changing needs and technological advancements. The research focuses on developing an IoT-oriented smart home automation system designed for intelligent self-adjustment based on environmental conditions and remote device control. A key objective is to substantiate the benefits of a modular architectural approach for extending system functionality and effectively utilizing complex adaptive algorithms to enhance performance.

**Materials and Methods.** To achieve the set goals, a novel hardware-software platform architecture is proposed, based on the division of functionality into independent modules. The use of the Home Assistant platform, its integrations, HACS, and Lovelace cards for intuitive system implementation and user interaction was investigated. The methodology includes practical examples of creating automation scenarios, including intellectualization through machine learning methods.

**Results and Discussion.** Comprehensive analysis demonstrated the high effectiveness and practical viability of the proposed modular approach. The developed solutions provide flexibility in system settings, ensuring ease of implementation even for complex configurations, and scalability, allowing for seamless integration of new devices and expanded coverage areas. The obtained data illustrate the possibilities of developing both simple and complex scenarios with elements of artificial intelligence and learning, including adaptive lighting, climate control, and security protocols.

**Conclusion.** The presented modular approach enables the application of broad customization and scalability capabilities for a smart home platform. The integration of open-source software tools, a wide range of sensors, and diverse communication interfaces enables the creation of systems ranging from basic automations to complex intelligent scenarios. This intelligent integration of scenario logic expands overall automation capabilities, transforming conventional systems into convenient and adaptive living environments.

**Keywords:** Home Assistant, hardware and software platform, automation

### INTRODUCTION

In the context of the development of Internet of Things (IoT) and cyber-physical systems technologies, there is a growing demand for effective automation solutions for



© 2025 Olha Shymchyshyn et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

residential and commercial premises. "Smart home" systems enable increased comfort, security, and energy efficiency through the integration of hardware controls with intelligent software. The modular approach is particularly relevant, providing flexibility, scalability, and adaptation to individual user needs.

Modern solutions for smart home automation systems are implemented based on wired and wireless infrastructures. Wireless solutions (Wi-Fi, Zigbee, Bluetooth) are characterized by ease of installation, flexibility, and scalability, but have drawbacks such as latency, dependence on a stable connection, and a limited range [1]. Wired systems (Ethernet, RS-485, KNX) provide high stability, security, and durability, but require more complex installation and planning. The choice between these options depends on the user's needs and installation conditions [2-4].

One significant obstacle to the development of integrated smart home automation systems is the heterogeneity of communication protocols, which hinders the seamless interaction of diverse devices. A significant number of devices operate within specialized ecosystems or use specific data transmission technologies, creating fragmentation and complicating their coordinated operation. To mitigate this problem, developments are underway for unified standards, in particular the Matter protocol, whose task is to provide a homogeneous communication environment for various devices and ecosystems. New standards, such as Matter, simplify integration between manufacturers, including Google, Amazon, and Samsung. Matter is particularly effective in combination with Thread — an energy-efficient network protocol for low-power devices. Thread provides a robust mesh topology for devices, while Matter provides a unified command and data protocol [5].

There are a number of popular automation platforms on the market: Home Assistant, Domoticz, OpenHAB, and Homebridge. Home Assistant is becoming a leader due to its flexible structure, hundreds of integrations, and community support. It is used for simple integration, automation, and visualization [3,4]. There are also commercial solutions from Google, Amazon, and Apple that provide cloud services but have limited customization and dependence on the Internet. In contrast, local systems like Home Assistant allow for fully autonomous operation with a high level of privacy [6].

The development of wireless technologies has led to the emergence of new protocols (Thread, LoRa, NB-IoT) that focus on low power consumption and long transmission distances [7]. Compatibility and security remain a problem: many devices have their own protocols without open documentation. Therefore, open platforms with HACS, MQTT, and REST API support are becoming increasingly important [8].

Automation systems are also being integrated with artificial intelligence platforms such as Google TensorFlow and OpenAI API, which allows for the implementation of adaptive scenarios based on user data and event history [9]. In addition, the role of educational solutions is growing — automation systems are used in laboratories, engineering courses, and scientific research [10].

The objective of this study is to provide a rationale for and illustration of the benefits inherent in a modular approach to constructing smart home automation platforms. This encompasses demonstrating how modularity facilitates the expansion of functional capabilities.

## MATERIALS AND METHODS

The paper presents a hardware-software implementation of a central hub for a smart home based on a mini-PC with the possibility of installing Home Assistant. The system supports a wide range of sensors and actuators, allowing for the implementation of both simple automations and complex scenarios. The open platform Home Assistant can be installed on Raspberry Pi, Orange Pi, Intel NUC, NAS, or ARM platforms. This versatile platform is designed for local control and automation of Internet of Things (IoT) devices from various manufacturers. It operates autonomously, without the need for connection to

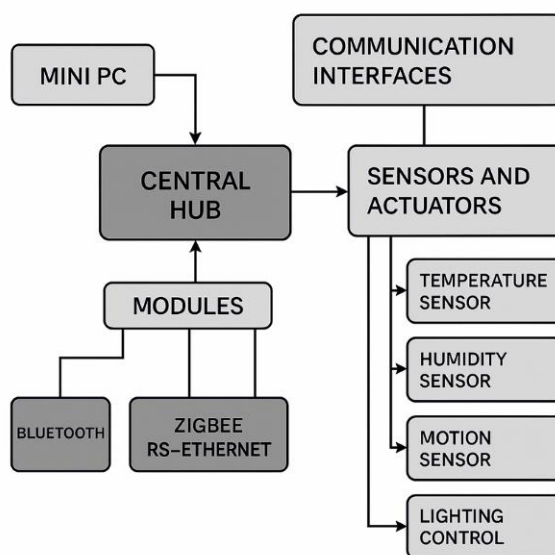


cloud services or constant internet access and provides support for a wide range of communication protocols. Home Assistant allows for the continuous development of automation solutions for residential/industrial use by adding additional programs for new objects, created by open-source community members or by creating custom code using programming languages supported by the platform (Python, YAML, JSON, JavaScript, visual programming-based development tools, etc.) [11].

Depending on the user's level and system flexibility requirements, four main installation options are provided: Home Assistant OS, Container, Supervised, and Core. The system allows for the integration of various devices and services into a unified automation infrastructure, using a modular approach, functional blocks, and logical control algorithms.

Modularity is one of the key principles for building a flexible and scalable smart home system. It allows for the separation of system functionality into independent components that are easily updated, replaced, or expanded.

A block diagram of the system's hardware part is shown in **Fig. 1**.



**Fig. 1.** Block diagram of the hardware part of the smart home automation system.

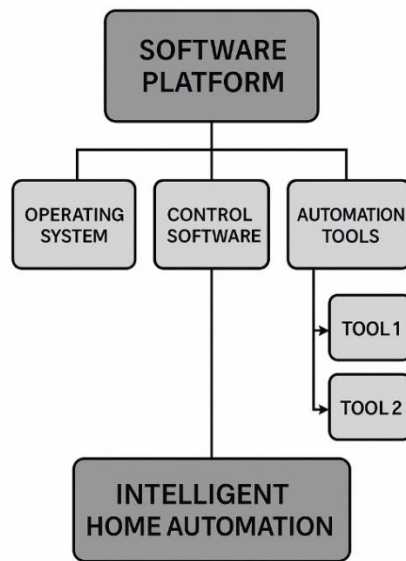
The physical structure of the system includes:

- A central hub operating on a single-board computer;
- Sensors: temperature, humidity, light, motion, smoke, water leak;
- Actuators: relays, lighting controllers, servomotors, infrared transmitters;
- Communication interfaces: wired (Ethernet, RS-485, UART) and wireless (Wi-Fi, Zigbee, Z-Wave, Bluetooth, Thread/Matter);
- Auxiliary modules: power supplies, neural accelerators, IR modules, audio interfaces, and displays.

At the software level, modularity is realized through integrations - ready-made software modules for connecting devices and services. A diagram of the software architecture of a smart home system using Home Assistant as the main open source control platform is shown on **Fig. 2**.

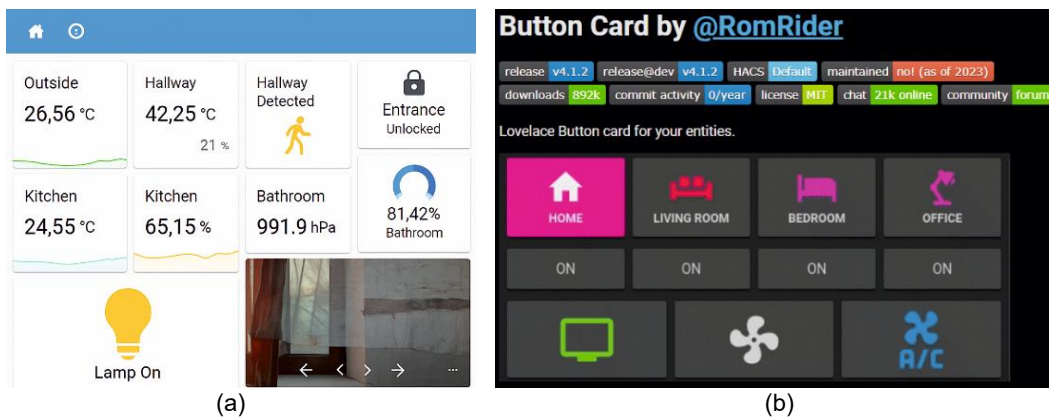
For displaying the status of smart devices and providing their control in Home Assistant, there is a flexible and fully customizable user interface called Lovelace. Thanks to its modular structure, users can compose information panels from various cards,





**Fig. 2.** Block diagram of the software architecture of a smart home system.

displaying only the necessary data and control elements (**Fig. 3a**). A special feature of Lovelace is its extensive customization options, which include the use of third-party extensions such as the Custom Button Card (**Fig. 3b**) for creating unique buttons with different functions and styles, the application of circular indicators for visual representation of levels or values, as well as deep CSS styling for complete control over the interface's appearance [11]. This allows for the adaptation of the interface to individual needs and the creation of intuitive and aesthetically pleasing smart home control panels.



**Fig. 3.** Visualization of the status of devices (a) and controls (b) in the Lovelace interface.

A key role in expanding the functionality of the Home Assistant platform is played by the Home Assistant Community Store (HACS), acting as an unofficial but extremely popular marketplace for community-contributed add-ons. This tool significantly simplifies the process of integrating custom extensions that are not included in the standard distribution, providing access to a wide range of content. Among the available add-ons are custom

integrations that provide support for a greater number of devices and services, interface elements for Lovelace that allow for the creation of unique visualizations and controls, ready-made automation blueprints that users can easily adapt, as well as themes for personalizing the look and feel of the interface.

**Fig. 4** demonstrates the application of the Custom Button Card in the Lovelace UI of the Home Assistant platform for creating customized control and data visualization elements. Two cards are presented (**Fig. 4a**), displaying current and external temperature and humidity readings. **Fig. 4b** shows a fragment of the YAML configuration for creating such a custom card. Among the parameters, you can see the definition of the card type (type: custom:button-card), the binding to a specific entity (entity: sensor.temp6\_temperature), the configuration of icon and state display (show\_icon: false, show\_state: false), as well as the definition of variables for minimum and maximum values, and start and end angles for the circular indicator. including binding to the entity and defining display parameters.



**Fig. 4.** Implementation of data visualization using Custom Button Card (a), YAML configuration (b), and SVG graphics (c).

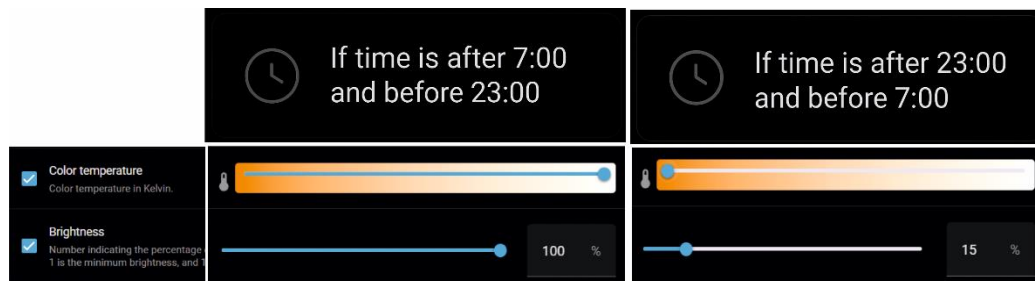
The presented SVG code fragment (**Fig. 4c**) implements the visualization of a circular indicator using gradients and dynamic filling via the stroke-dasharray and transform: rotate() attributes. Thus, by integrating YAML configurations and SVG graphics into the Custom Button Card, it is possible to create complex and visually informative interface elements for controlling and monitoring smart home parameters in Home Assistant.

## RESULTS AND DISCUSSION

An important element for building complex and adaptive automation systems in Home Assistant is Helpers. These are built-in tools that allow users to create their own

logical entities to extend the capabilities of automations, scripts, and the Lovelace interface. These entities can take various forms, including logical variables (boolean values), numeric inputs, selectors, text fields, timers for counting down time, and counters for recording events. Helpers act as abstractions that store a certain state or value, which can be used as a condition for triggering automations, as a variable in scripts, or as a data source for display in the user interface. Thanks to their flexibility, Helpers allow for the implementation of more complex logic and the creation of more interactive and personalized smart home automation scenarios. For example, using a numeric helper, you can set a target temperature for a thermostat, and a boolean helper can display the "home" or "away" state, influencing the behavior of other devices. Timers and counters, in turn, open up possibilities for creating automations that depend on time or the number of certain events [4].

Automation scenarios in smart home systems have evolved from simple conditional rules to complex intelligent algorithms. The simplest scenarios are implemented based on "if → then" logic. They contain triggers (events), conditions, and actions. For example, if the motion sensor is activated after 10:00 PM → turn on the night light with 20% brightness. Scenarios can include multiple conditions, delays, and repetitions. For example, if there is motion in the room and the illuminance is < 50 lx → turn on the light. **Fig. 5** shows an example of implementing a lighting control scenario based on the time of day. During the day, from 7:00 AM to 11:00 PM, the system automatically sets full brightness and a cool color temperature of the light, providing optimal lighting for active tasks. At night, from 11:00 PM to 7:00 AM, the lighting switches to a low brightness mode with a warm color temperature, creating a cozy atmosphere for relaxation. This approach not only automates routine operations but also adapts the environment to the needs of users, contributing to their comfort and well-being.



**Fig. 5.** Algorithm for automatic switching of lighting modes depending on the time of day.

The proposed solution also enables the implementation of scenarios such as "night mode," "vacation," and "energy saving," based on the analysis of input data and logic, and includes a web interface for visual control and event history.

Intelligentization of scenarios in smart home systems involves the transition from simple automation rules to complex, adaptive algorithms capable of responding to changing conditions and user needs. This is achieved through the integration of machine learning and artificial intelligence methods, allowing the system to analyze data from sensors, user interaction history, and external factors. Based on this analysis, the system can automatically adjust environmental parameters, optimize energy consumption, enhance security, and provide personalized comfort. Thanks to the support of Python scripts, Node-RED, and REST API, users can create complex scenarios such as adapting lighting to sleep patterns, anomaly detection, and prediction based on past behavior. The

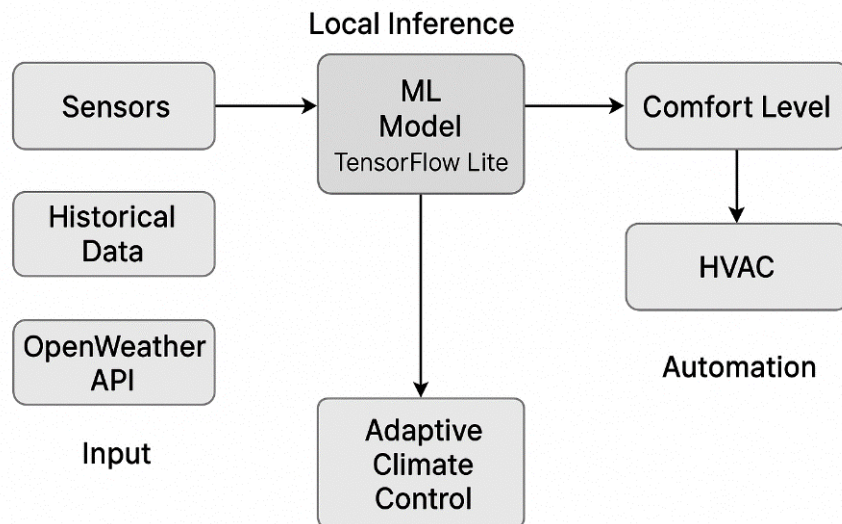
use of local machine learning models (TensorFlow Lite, Edge AI) allows for processing without transferring data to the cloud.

Thus, Home Assistant facilitates the transition from elementary reactions to events to complex, adaptive behavior of the smart home system, taking into account context, environmental conditions, and user needs.

With the rise of computational capabilities and the widespread adoption of machine learning (ML) and artificial intelligence (AI) technologies, automated systems have reached a new level of adaptability. AI enables smart home platforms to expand their functionality by predicting user behavior, recognizing patterns in sensor data, adapting to daily routines, and autonomously adjusting system operation modes.

For instance, using classification or time series prediction algorithms, a smart home system can anticipate when a user is likely to return home and activate lighting, climate control, or security features in advance. AI integration also unlocks anomaly detection capabilities, allowing the system to identify irregularities such as water leaks, unusual activity patterns, or excessive energy consumption and respond accordingly. Moreover, natural language processing tools provide voice command recognition and interpretation of textual queries for generating new automation scenarios. Within the Home Assistant platform, integration with Python-based tools (e.g., TensorFlow, scikit-learn) enables the development of custom analytics and control modules. These tools empower both developers and advanced users to design a truly intelligent and context-aware home environment, facilitating personalized comfort, safety, and energy efficiency.

**Fig. 6** illustrates how an adaptive climate control system utilizes AI algorithms to process environmental and behavioral data. The system analyzes both sensor inputs and external conditions to make real-time decisions, adjusting heating, ventilation, and cooling systems accordingly. This structure enables continuous improvement of system performance and user satisfaction through localized learning and personalized control logic.



**Fig. 6.** Block diagram of an AI-driven adaptive climate control scenario

## CONCLUSION

The paper investigates a modular approach to building a hardware and software platform for smart home automation. The analysis has shown that modularity is a key factor in creating flexible, scalable, and adaptable systems tailored to the needs of users. The proposed architecture, based on the separation of functionality into independent hardware and software components, differs from traditional monolithic systems and significantly simplifies the process of system expansion, integration of new devices, and services. The considered examples of using the Home Assistant platform, its integrations, HACS, and Lovelace cards — particularly the developed custom cards for data visualization — demonstrate the extensive customization capabilities and the potential for creating both simple and complex automation scenarios. Furthermore, the implementation of adaptive, AI-driven scenarios highlights the transition from static rules to intelligent behavior. Integrating machine learning models within the modular framework enables real-time decision-making, anomaly detection, and personalized environmental control. This elevates the functionality of smart home systems to a new level of context awareness and user-centered intelligence. The proposed principles and solutions can be applied to a variety of applications, from basic lighting control systems to comprehensive environments that provide enhanced safety, comfort, and energy efficiency. Further research will focus on the development of self-scaling algorithms, optimization of energy consumption across modules, and the enhancement of cybersecurity in intelligent automation platforms.

## COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

## AUTHOR CONTRIBUTIONS

Conceptualization, [V.S., O.S.]; methodology, [M.S.]; validation, [O.S.]; investigation, [V.S.]; writing – original draft preparation, [O.S.]; writing – review and editing, [M.S.]; visualization, [V.S.].

All authors have read and agreed to the published version of the manuscript.

## REFERENCES

- [1] Ożadowicz, A. (2024). Generic IoT for smart buildings and field-level automation—Challenges, threats, approaches, and solutions. *Computers*, 13(2), 45–58. <https://doi.org/10.3390/computers13020045>
- [2] Vermesan, O., & Marples, D. (2023). Advancing edge artificial intelligence: Novel architectures for smart environments. In *Shaping the Future of IoT with Edge Intelligence* (pp. 45–67). Routledge. <https://doi.org/10.1201/9781032632407>
- [3] Shukla, S., Meghana, K. M., Manjunath, C. R., & Shantosh, N. (2017). Comparison of Wireless Network over Wired Network and Its Type. *Int. J. Res. Granthaalayah*, 5, 14-20. <https://doi.org/10.5281/zenodo.572289>
- [4] Home Assistant. (n.d.). *Home Assistant documentation*. Retrieved May 16, 2025, from <https://www.home-assistant.io>
- [5] ZigBee Alliance. (2022). *Matter: The foundation for connected things*. Connectivity Standards Alliance. <https://csa-iot.org/all-solutions/matter/>
- [6] Lee, H., Kim, J., & Park, S. (2022). *Internet of Things technology: Balancing privacy concerns with convenience*. *Telematics and Informatics*, 70, Article 101816. <https://doi.org/10.1016/j.tele.2022.101816>



- [7] Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*, 5(1), 1–7. <https://doi.org/10.1016/j.ict.2017.12.005>
  - [8] Joel A. Cujilema Paguay, Gustavo A. Hidalgo Brito, Dixys L. Hernandez Rojas, and Joffre J. Cartuche Calva. (2023). Secure home automation system based on ESP-NOW mesh network, MQTT and Home Assistant platform. *IEEE Latin America Transactions*, 21(7), 829–838. <https://doi.org/10.1109/TLA.2023.10244182>
  - [9] Himeur, Y., Sayed, A. N., Alsalemi, A., Bensaali, F., & Amira, A. (2024). Edge AI for Internet of Energy: Challenges and perspectives. *Internet of Things*, 25, 101035. <https://doi.org/10.1016/j.iot.2023.101035>
  - [10] Al-Yaman, M., Alswaiti, D., Alsharkawi, A., & Al-Tae, M. (2025). A cost-effective modular laboratory solution for industrial automation and applied engineering education. *MethodsX*, 14, 103388. <https://doi.org/10.1016/j.mex.2025.103388>
  - [11] Munteanu, L., Suvar, M. C., & Florea, G. D. (2022). Residential security through the Home Assistant platform. *MATEC Web of Conferences*, 354, 00008. <https://doi.org/10.1051/mateconf/202235400008>
- 

## МОДУЛЬНИЙ ПІДХІД ДО ПОБУДОВИ АПАРАТНО-ПРОГРАМНОЇ ПЛАТФОРМИ АВТОМАТИЗАЦІЇ РОЗУМНОГО БУДИНКУ: ВІД ПРОСТИХ ПРАВИЛ ДО ІНТЕЛЕКТУАЛЬНИХ СЦЕНАРІЇВ

Ольга Шимчишин<sup>1\*</sup>, Мар'ян Шимчишин<sup>2</sup>, Владислав Сергієнко<sup>1</sup>

<sup>1</sup>Національний університет «Львівська політехніка»,  
Україна, м. Львів, вул. Степана Бандери 12

<sup>2</sup>ПВНЗ «Європейський університет»,  
Україна, м. Львів, вул. вул. Кушевича, 5

### АНОТАЦІЯ

**Вступ.** Стаття присвячена проблемі створення гнучких, масштабованих та адаптованих до користувача систем автоматизації, які можуть розвиватися зі змінними потребами та технологічними досягненнями. Дослідження зосереджено на розробці IoT-орієнтованої системи домашньої автоматизації, здатної інтелектуально налаштовуватися на основі умов середовища та дистанційно керувати пристроями. Ключовою метою є обґрунтування переваг модульного архітектурного підходу для розширення функціональності та використання адаптивних алгоритмів.

**Матеріали та методи.** Для досягнення поставлених цілей запропоновано архітектуру апаратно-програмної платформи, що базується на розділенні функціональності на незалежні модулі. Досліджено використання платформи Home Assistant, її інтеграцій, HACS та карток Lovelace для інтуїтивної реалізації системи та взаємодії з користувачем. Методологія включає практичні приклади створення сценаріїв автоматизації, включаючи інтелектуалізацію за допомогою методів машинного навчання.

**Результати та обговорення.** Комплексний аналіз продемонстрував високу ефективність та практичну життєздатність запропонованого модульного підходу. Розроблені рішення забезпечують гнучкість налаштувань, гарантуючи легкість впровадження навіть для складних конфігурацій, масштабованість, дозволяючи інтегрувати нові пристрої та розширювати зони покриття. Отримані дані ілюструють можливості розробки як простих так і комплексних сценаріїв з елементами штучного



інтелекту та навчання, включаючи адаптивне освітлення, клімат-контроль та протоколи безпеки.

**Висновки.** Представлений модульний підхід дозволяє застосовувати широкі можливості налаштування та масштабованості для платформи розумного будинку. Інтеграція програмних засобів з відкритим вихідним кодом, широкого спектру сенсорів та різноманітних інтерфейсів зв'язку дозволяє створювати системи, починаючи від базових автоматизацій до складних сценаріїв з елементами інтелекту. Інтелектуальна інтеграція сценарної логіки розширює загальні можливості автоматизації, перетворюючи звичайні системи на зручні та адаптивні житлові середовища.

**Ключові слова:** Home Assistant, апаратно-програмна платформа, сценарії автоматизації

UDC: 004.89 + 004.738.5

## COMPARATIVE STUDY OF FEATURE DETECTORS AND FILTERING METHODS IN IMAGE MATCHING

Andriy Fesiuk\* , Yuriy Furgala 

Ivan Franko National University of Lviv,  
50 Drahomanova St., 79005 Lviv, Ukraine

Fesiuk A. V., Furgala Y. M. (2025). Comparative Study of Feature Detectors and Keypoint Filters in Image Matching. *Electronics and Information Technologies*, 31, 71–88.  
<https://doi.org/10.30970/eli.31.7>

### ABSTRACT

**Background.** In modern computer vision, the accuracy and reliability of image matching primarily depend on the quality of local feature processing. False correspondences, which arise from changes in scale, illumination, or repetitive structures, have the potential to distort a scene's geometric model. Therefore, applying filtering algorithms that can distinguish informative matches from noise becomes an important step. Despite significant progress, most research focuses only on specific combinations of detectors and filtering methods, which prevents a comprehensive understanding of their interaction.

**Materials and Methods.** To investigate this issue, a series of experiments was conducted, and a representative subset from the Photo Tourism dataset was selected. Keypoint detection and description were performed, followed by matching, outlier filtering, and quantitative evaluation. The comparison involved the SIFT, SURF, KAZE, AKAZE, ORB, and BRISK detectors combined with the RANSAC, LMedS, RHO, GMS, VFC, and LPM filtering methods. For the evaluation, metrics such as the Fisher Criterion, IQR Separability, Whisker Gap, and a custom-developed metric called SMS were applied.

**Results and Discussion.** The investigation revealed that performance varies significantly among the detectors: binary descriptors offer much higher processing speeds. In contrast, methods using floating-point descriptors are more informative but require more computational resources. The hierarchy of filtering methods was consistent across all setups: VFC achieved the highest quality based on separability metrics, while LPM showed the most considerable difference between the distribution boundaries. RANSAC and LMedS remain classic benchmarks, while GMS and RHO serve as fast, compromise alternatives.

**Conclusion.** The results show that image matching effectiveness depends on the combination of the detector, the number of keypoints, and the filtering method. A comprehensive approach enables the selection of the right strategies for specific tasks, ranging from applications that require fast processing to scenarios that necessitate maximum separability or boundary error control. The analysis and metrics used provide a basis for future research and improvements in practical computer vision systems.

**Keywords:** feature detection, keypoint descriptors, image matching, match filtering.

### INTRODUCTION

In the modern world, the amount of visual data is increasing rapidly - from countless photos on social media to continuous video streams from surveillance cameras and autonomous systems. For automated analysis of this large flow of information, local feature processing algorithms are essential. Detecting keypoints is fundamental for various computer vision tasks, including three-dimensional reconstruction from images or Structure



© 2025 Andriy Fesiuk & Yuriy Furgala. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

from Motion, panorama creation, Simultaneous Localization and Mapping (SLAM), and visual odometry [1-5].

A typical image processing pipeline involves detecting keypoints, describing them, and matching based on the similarity of these descriptors. However, the initial set of correspondences often contains many false matches or outliers, which are caused by visual ambiguity, repetitive textures, or lighting changes. Even a few outliers can significantly affect the estimation of geometric models. Therefore, filtering is an important step that eliminates false matches and returns a set of pairs that align geometrically.

To reduce the number of outliers, a wide range of methods has been proposed - from classic parametric techniques, notably RANSAC (Random Sample Consensus) [6] and its variations LMedS (Least-Median of Squares) [7] and RHO (Randomized Hough Transform) [8], to non-parametric approaches such as GMS (Grid-based Motion Statistics) [9], VFC (Vector Field Consensus) [10], and LPM (Locality Preserving Matching) [11]. Previous studies have primarily focused on specific aspects of the problem, either by comparing detectors and descriptors [12-14] or by analyzing the performance of filters in limited combinations [15-17]. The main contribution of this work lies in evaluating the interaction between detection and filtering methods.

A combination of six detectors - SIFT (Scale-Invariant Feature Transform) [18], SURF (Speeded-Up Robust Features) [19], KAZE [20], AKAZE (Accelerated KAZE) [21], ORB (Oriented FAST and Rotated BRIEF) [22], and BRISK (Binary Robust Invariant Scalable Keypoints) [23] - with six filtering methods - RANSAC, LMedS, RHO, GMS, VFC, and LPM - was examined. It is demonstrated how specific detector-filter pairings affect execution time and the overall matching quality. The experiments were performed on a subset of the public Image Matching Challenge Photo Tourism dataset [24], which features scenes with significant variations in viewpoints and scales of various architectural landmarks. Unlike traditional metrics such as precision and recall, our primary focus was on the ability of the filters to separate, specifically to increase the statistical distance between the distributions of similar and dissimilar images. To achieve this, we applied a set of specialized metrics: the Fisher Criterion, IQR (Interquartile Range) Separability, Whisker Gap, and a proposed metric, Sigma-Margin Separability (SMS). Together, these metrics offer a comprehensive view of the distributions' central parts and their tail regions.

## MATERIALS AND METHODS

A software pipeline was developed and implemented to compare filtering methods. It includes several sequential stages: dataset preparation, keypoint detection and description, initial matching, outlier filtering, and an evaluation of effectiveness using a set of class separability metrics.

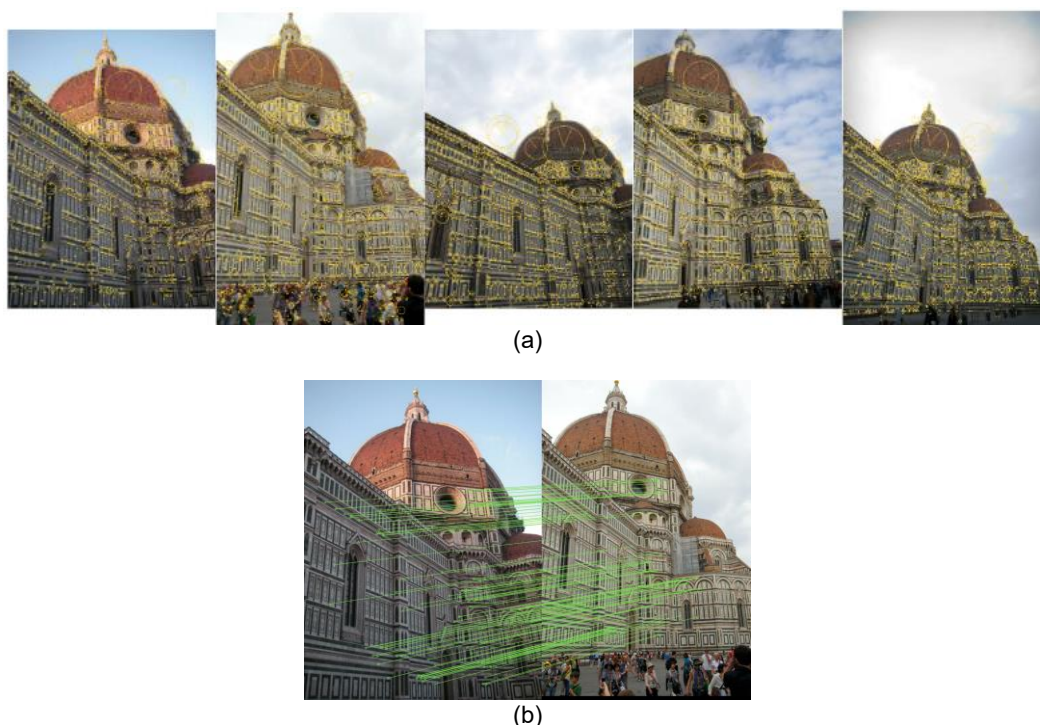
The public Image Matching Challenge Photo Tourism dataset, which includes architectural scenes with a wide range of viewpoints and lighting conditions, was used as the data source. Since the full dataset is resource-intensive and contains a large amount of data, this study used an automated process to select representative images for each scene. Initially, local features were detected and described for each frame using the SIFT method, which is known for its robustness to scale and viewpoint changes and its ability to assess an image's informativeness. A normalized, averaged vector of descriptors then represented each image. From this, a set of candidate pairs was generated by identifying, for each frame, the nearest neighbors based on cosine similarity, thereby reducing the number of comparisons without losing meaningful intra-scene relationships. For each candidate pair, matching was performed using the Brute-Force method with Lowe's ratio test [18], followed by an evaluation of geometric consistency using RANSAC with a homography model [25]. Utilizing RANSAC at this stage is a key methodological choice; previous research [26] has shown that combining this method with SIFT yields the best results compared to other methods within the USAC family [27]. Based on these similarity

values, a symmetric similarity matrix was created, from which the densest k-subset with the highest average mutual similarity was selected, ensuring consistency among images and eliminating duplicates. As a result, 10 images from 10 classes were selected, forming the final test set of 100 frames. Examples of images selected into the dataset are presented with keypoints detected in Fig. 1a and with keypoints matched in Fig. 1b.

In this study, six methods for detecting keypoints were used: SIFT, SURF, ORB, BRISK, AKAZE, and KAZE. The initial number of keypoints for each detector was different. This is because only SIFT and ORB have direct parameters to limit the number of features, while SURF, BRISK, AKAZE, and KAZE are configured using a related threshold. The response attribute [28] was used to restrict and align the number of keypoints between detectors, and only the top features were selected for further analysis. This attribute, which depends on the detector's internal quality assessment of each point, is a standard practice for ensuring a fair comparison and serves as a universal criterion for sorting features; however, its physical meaning and mathematical basis are unique to each algorithm.

- SIFT: The amplitude of the extremum in the Difference of Gaussians (DoG) pyramid, which indicates contrast [18].
- SURF: The determinant of the Hessian matrix, utilized for identifying blob structures [19].
- ORB та BRISK: Metrics based on corner detectors - Harris for ORB [22] and AGAST for BRISK [23].
- KAZE та AKAZE: The determinant of the Hessian matrix calculated in a non-linear scale-space, which enables improved preservation of object boundaries compared to traditional Gaussian blurring.

In our study, the number of keypoints ranged from 500 to 5000, thereby enabling the assessment of the impact of keypoint density on subsequent processing stages.



**Fig. 1.** Example of images in dataset (Florence Cathedral): a) with keypoints detected (marked yellow); b) with keypoints matched (green lines).

The Brute Force matcher method was employed with a search for the two nearest neighbors ( $k = 2$ ). The distance measure was chosen based on the descriptor type: Euclidean distance for floating-point descriptors and Hamming distance for binary descriptors [28]. After identifying candidate pairs, Lowe's ratio test was applied with a threshold of 0.75. The choice of this value was deliberate, as its effectiveness and appropriateness were evaluated in our earlier research [29]. This research demonstrated that this value provides a reliable balance between the number and quality of initial matches. This step removes ambiguous correspondences and minimizes the effect of random coincidences, thereby forming the initial set of matches for subsequent geometric verification and filtering. The matching time for each image pair was measured as the combined duration of the Brute-Force matching process and the subsequent Lowe's ratio test verification.

The primary focus of the study is a comparison of six filtering methods, representing three different strategies for outlier rejection. Homography was used as the base model for all methods requiring geometric verification. The investigation employed parametric methods, including the baseline RANSAC with its iterative approach, its variant, LMedS, which minimizes the median of errors, and the optimized RHO, which prioritizes sampling high-quality matches. Non-parametric methods were also examined, including VFC, which evaluates the smoothness of the match vector field, and LPM, which verifies the preservation of local neighborhood structure. Additionally, the statistical method GMS was analyzed, utilizing a grid to assess motion consistency in local regions. This approach enables an investigation into how filters based on a geometric model compare to methods that use spatial or statistical approaches, without relying on model parameterization.

The experiments were conducted on a 2019 MacBook Pro with an Intel Core i9-9880H processor, clocked at 2.30 GHz, featuring eight physical cores/16 threads, and 32 GB of RAM, running macOS 15.6.1. The software versions used were Python 3.9.13 and OpenCV 4.10.0. Parallel processing was achieved using a process pool with eight workers; OpenCV's internal parallelism was disabled to reduce variability [28]. The detection, matching, and filtering were measured within their respective procedures.

To quantitatively assess the effectiveness of each filtering method, a methodology was established based on analyzing class separability. Central to this approach is the similarity coefficient  $S$ , which is calculated for each pair of images ( $i, j$ ) after the filtering process has been applied. This coefficient is a normalized measure that indicates the proportion of "good" matches relative to the total number of features available.

$$S_{i,j} = 100 \cdot \frac{N_{good}}{\min(N_i, N_j)}, \quad (1)$$

where  $N_{good}$  is the number of matches remaining after filtering, and  $N_i$  and  $N_j$  are the total number of keypoints in images  $i$  and  $j$ , respectively. Based on this coefficient, all image pairs were divided into two categories: "Similar" for images that are similar, and "Different" for images that are different.

The effectiveness of a filtering method is determined by its ability to create two well-separated distributions of the coefficient  $S$  for these two classes. After processing scenes with an ideal filter, the  $S$  values should be high for similar images and low for dissimilar ones. A comprehensive set of four metrics was used to evaluate this separability quantitatively.

- Fisher Criterion: A classic statistical measure that compares the distance between the centers of the distributions (between-class variance) to their internal consistency (within-class variance). A higher value indicates better separation.

$$F = \frac{(\mu_{sim} - \mu_{diff})^2}{\sigma_{sim}^2 + \sigma_{diff}^2}, \quad (2)$$

where  $\mu$  та  $\sigma^2$  are the mean and variance of the respective distributions.

- IQR Separability Coefficient: An outlier-robust metric that quantitatively measures the gap between the central 50% of data in each distribution (the interquartile ranges), directly showing the visual separation of the "boxes" in box plots.

$$IQR \text{ Separability} = \frac{Q1_{sim} - Q3_{diff}}{Q3_{sim} - Q1_{diff}}, \quad (3)$$

where Q1 and Q3 are the first and third quartiles of the respective distributions.

- Whisker Gap: A straightforward numerical estimate of the distance between the outermost boundaries of the main data ranges, which reflects the visual gap between the "whiskers" on box plots calculated using the standard Tukey's method with a coefficient of  $1.5 \cdot IQR$ .

$$Whisker \text{ Gap} = Q1_{sim} - Q4_{diff} \quad (4)$$

where  $Q1_{sim}$  is the lower bound of the similar images distribution, and  $Q4_{diff}$  is the upper bound of the dissimilar images distribution.

- Sigma-Margin Separability (SMS) Coefficient: To thoroughly assess class separability, we have developed a new metric called Sigma-Margin Separability, in addition to traditional methods. The motivation behind this development comes from the limitations of existing metrics. For instance, the Fisher Criterion does not directly verify whether a "guaranteed margin" exists between the distributions, particularly in their tails. Similarly, metrics like IQR Separability are robust to outliers but achieve this by focusing only on the central 50% of the data, completely ignoring behavior at the distribution boundaries. While the Whisker Gap does consider the outer ranges, it is based on a structural definition rather than a probabilistic one. In our case, margins between the statistical boundaries of the "Similar" and "Different" distributions were evaluated. These boundaries are determined by the  $k$ -sigma rule ( $\mu \pm k \cdot \sigma$ ). The metric is calculated using the formula:

$$SMS(k) = \frac{(\mu_{sim} - k \cdot \sigma_{sim}) - (\mu_{diff} + k \cdot \sigma_{diff})}{\mu_{sim} - \mu_{diff}} \quad (5)$$

where the parameter  $k$  controls the stringency of the criterion, in this work, values of  $k = 1, 2$ , and  $3$  were used, which correspond to checking the margin between boundaries encompassing approximately 68%, 95%, and 99.7% of the data in each distribution, respectively. A larger value of  $k$  indicates a more rigorous evaluation. The numerator in the formula represents the "k-sigma margin" – the distance between the lower bound of the "Similar" distribution and the upper bound of the "Different" distribution. The denominator serves as a normalization factor. A positive SMS value indicates high separation reliability, showing a "guaranteed margin" between the  $k$ -sigma boundaries. Conversely, a negative value suggests overlap between them. A more detailed statistical analysis of SMS, including sensitivity to distributional shifts, sample-size effects, and robustness, is beyond the scope of this paper and will be covered in a separate study.



Thus, using this set of metrics enables a thorough evaluation of filter effectiveness by examining different aspects of separability, ranging from traditional statistical methods to reliability assessments with strict criteria and intuitively understandable indicators that visually demonstrate the overlap of distributions. For all the listed metrics, higher values denote better class separation. This approach offers a more complete understanding of how detection methods perform when combined with various filtering techniques.

## RESULTS AND DISCUSSION

This section provides a comprehensive analysis of the research findings. Initially, the computational performance of the methods is evaluated across three essential data processing stages: detection and description, matching, and filtering. Subsequently, a detailed assessment of the filtering quality is conducted for each detector.

The results presented in **Table 1** show the average number of detected keypoints, the time needed for their detection and description, and the normalized efficiency, calculated as the average time for detection and description per 1000 keypoints. Normalizing these metrics enables an objective comparison of the detectors' efficiency by mitigating the influence of the varying number of keypoints they generate.

The obtained results indicate a distinct performance benefit for the binary methods. While each method produced a different number of keypoints with the given input parameters, the ORB method remains the clear leader in both total processing time and normalized efficiency. BRISK takes second place and significantly outperforms methods that use floating-point descriptors.

In contrast, methods such as SIFT, SURF, and KAZE, which are based on floating-point descriptors, were more resource-intensive. KAZE shows the lowest performance, with its normalized time approximately 3.8 times higher than SIFT's and nearly 2.5 times higher than SURF's; this aligns with the computational complexity of the non-linear diffusion scale-space on which the method is based. The accelerated version, AKAZE, operates much faster; however, in terms of normalized efficiency, it still significantly lags behind the other binary methods.

The dependency of the average matching time on the number of keypoints for a single image pair is shown in **Fig. 2**. The slowest method was SIFT, which is over 2.5 times slower than ORB. Analyzing scalability - the rate at which time increases with more points - shows that KAZE is the most robust to an increasing number of features, with the lowest rate of about 90 ms for each additional 1000 points, nearly matching ORB. In contrast, SIFT scales much worse, exceeding KAZE's rate by more than three times. Other methods vary in performance. SURF offers moderate speed and scalability. BRISK and AKAZE work well with a few keypoints but become less efficient as the number increases, unlike ORB and KAZE.

The filtering time is shown in **Fig. 3**, where a performance hierarchy of the investigated methods is visible, which was consistently maintained for all six detection methods. This

**Table 1. Time characteristics and computational efficiency of keypoint detection and description.**

Method	Avg number of keypoints	Avg time of detection, ms	Avg time of description, ms	Avg total time, ms	Avg total time per 1000 keypoints, ms
SIFT	6811	213.37	358.34	571.58	85.1
SURF	8486	273.37	834.56	1108.03	131.81
ORB	8397	37.46	16.17	53.63	6.38
BRISK	9517	98.47	77.26	175.73	18.57
AKAZE	8796	197.4	260.92	458.31	52.88
KAZE	8404	1286.61	1325.31	2611.92	318.49

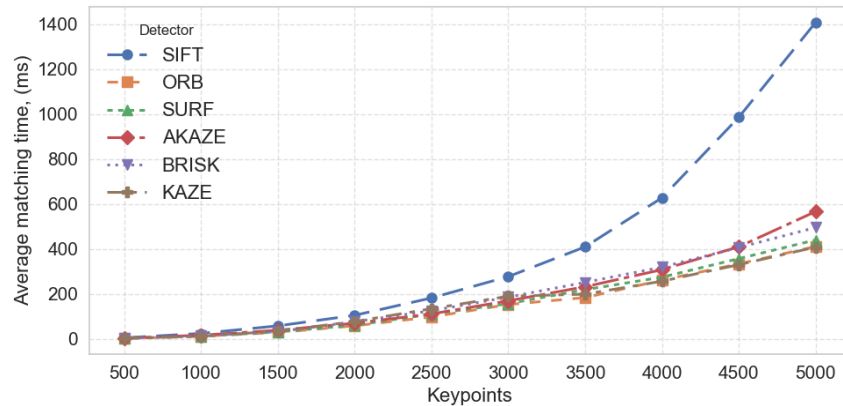


Fig. 2. Average matching time per image pair vs. number of keypoints for different feature detectors.

stability suggests that the choice of filtering method has a significant impact on the matching process. RHO and VFC demonstrated the highest speed and scalability, with an increase in the number of keypoints having almost no effect on processing time. In contrast, classic RANSAC proved to be the least effective, lagging behind the leaders in speed by an order of magnitude and exhibiting increased processing time as the number of points increased. GMS, LMedS, and LPM occupied an intermediate position, with GMS typically being the fastest and LPM the slowest.

The type of detector affects the time values. However, the relative ranking of the filtering methods stays the same. For the SIFT, SURF, and AKAZE detectors, the performance difference between the high-speed RHO and VFC and the slower RANSAC is most noticeable. For the ORB, BRISK, and KAZE detectors, the overall execution times are lower.

An analysis of the results for the SIFT detector, shown in Fig. 4, indicates that increasing the keypoint limit consistently improves all filtering quality metrics. The effectiveness plateaus at around 3000-4000 keypoints, where adding more features no longer provides a significant benefit.

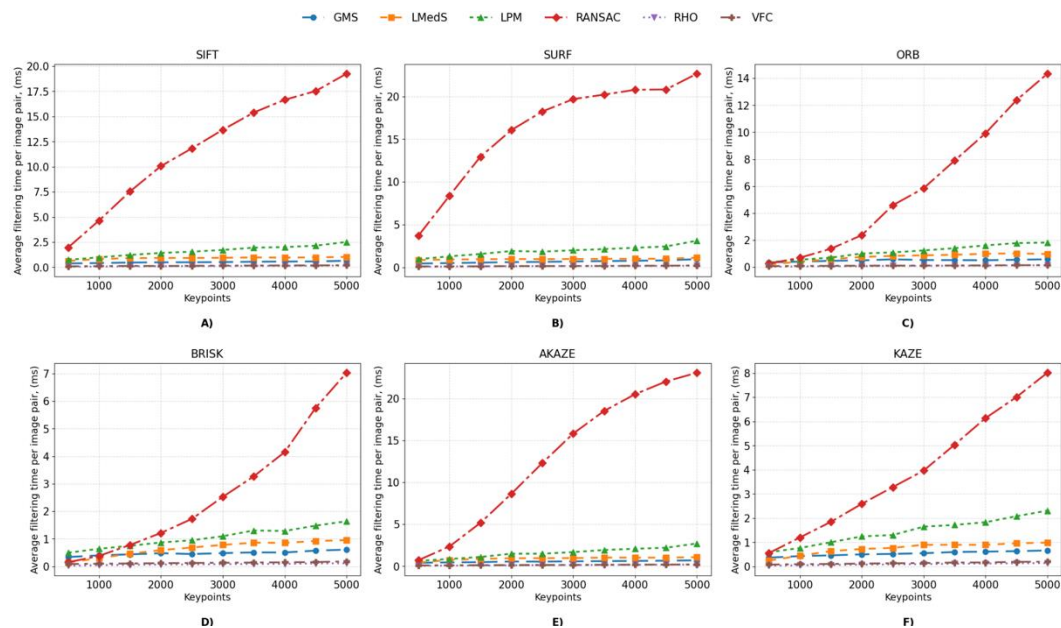
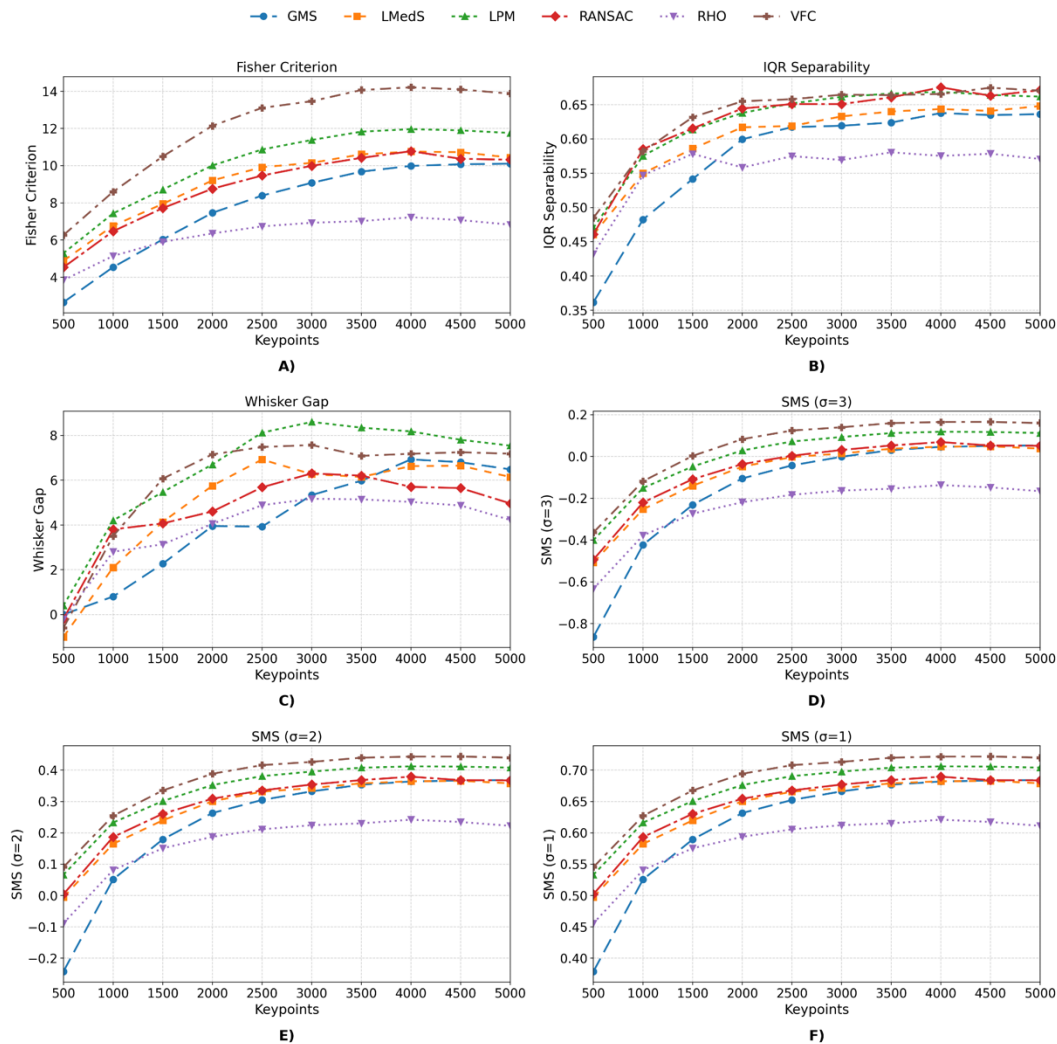


Fig. 3. Average filtering time per image pair vs. number of keypoints for different feature detectors: A) SIFT; B) SURF; C) ORB; D) BRISK; E) AKAZE; F) KAZE.



**Fig. 4.** Filtering quality metrics for SIFT: A) Fisher Criterion; B) IQR Separability; C) Whisker Gap; and the SMS coefficients for D)  $\sigma=1$ , E)  $\sigma=2$ , and F)  $\sigma=3$

According to the Fisher Criterion, the results of the VFC method increase by approximately 2.3 times when moving from 500 to 4000 keypoints. In the saturation zone at 4000 keypoints, it exceeds LPM by nearly 20% and RANSAC and LMedS by about 30–35%. This shows that VFC achieves the best balance between between-class separation and within-class variance, while GMS, with 40% lower metrics, and RHO, which is almost twice as poor, perform worse.

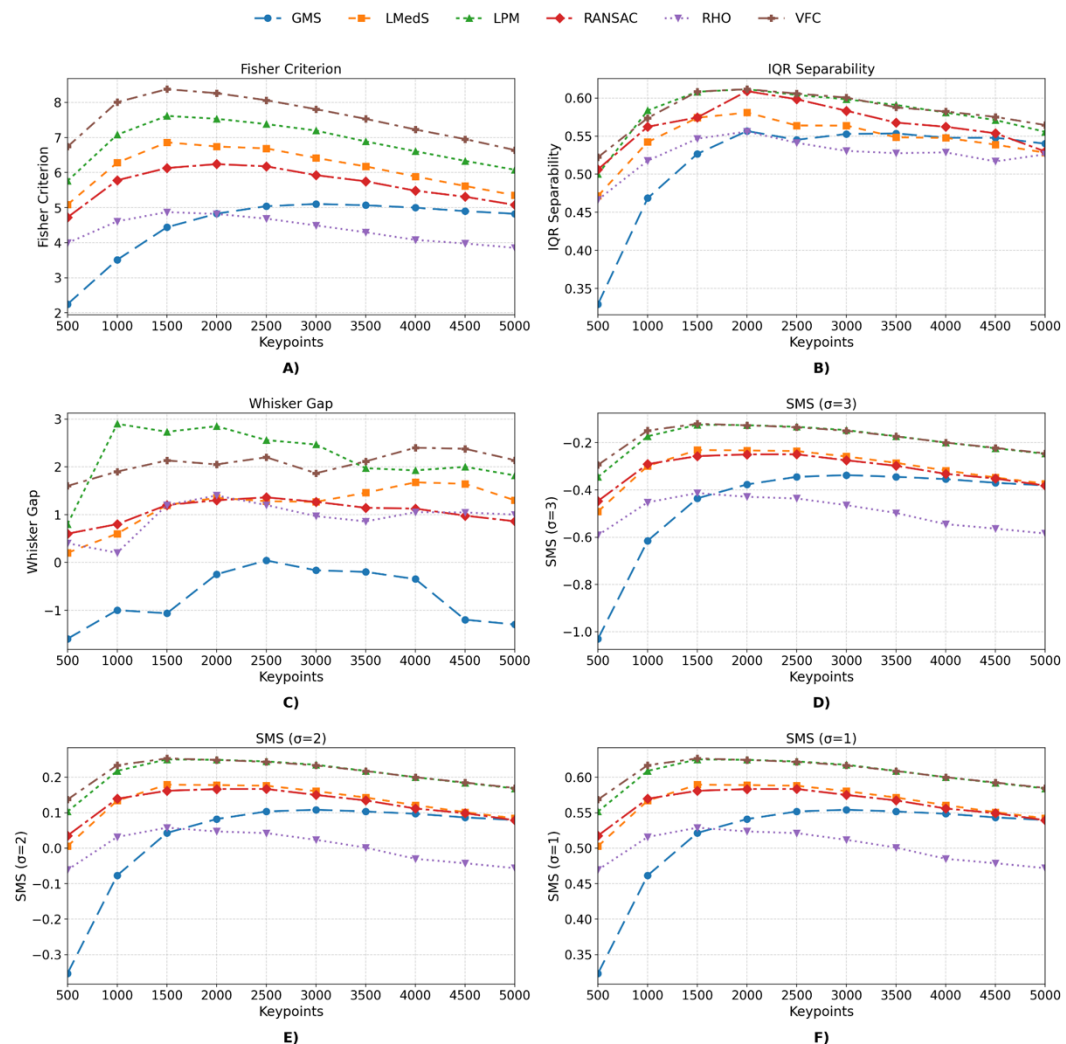
For the Whisker Gap metric, the best results are achieved by LPM. Starting from the 2500 keypoint range, it creates the most significant gap between the classes, surpassing VFC by approximately 10–15%. This makes LPM a good choice where it is important to minimize errors at the extreme boundaries of the data, which are often associated with outliers. The GMS method also exhibits a rapid increase in values after 2500 keypoints, and in the 4000–5000 range, it yields higher values than the RANSAC, LMedS, and RHO methods.

In contrast, for the IQR Separability metric at 4000 keypoints, the classic RANSAC shows a slight edge over VFC. This suggests that RANSAC works well with reliable matches but less effectively on more ambiguous data. The values for LMedS are similar to RANSAC's, while GMS is consistently 5–10% lower than VFC, and for RHO, this gap widens to 10–15%.

The values of the SMS coefficients demonstrate the superiority of VFC. In the 3000-4000 keypoint zone for  $\sigma = 1$ , the VFC method outperforms LPM by approximately 2-3% and RANSAC by 5-6%. For  $\sigma = 2$ , the advantage of VFC over LPM increases to approximately 40%. Furthermore, the  $\sigma = 3$  values for VFC are in the positive region starting from 1500 keypoints, which indicates reliable separation even in the tail regions of the distributions. GMS is roughly 10-15% less effective than VFC across the entire coefficient range, while RHO has the lowest values, indicating sensitivity at the extreme points.

Unlike SIFT, the SURF detector exhibits a distinct performance trend, as illustrated in Fig. 5. All metrics display a sharp rise until reaching a peak between 1000 and 2000 keypoints, followed by a gradual decline. This pattern aligns with the presumption that at higher keypoint counts, more weak correspondences enter the SURF sample, which reduces the separation between the classes.

According to the Fisher Criterion, the best result is shown by VFC. Its metric reaches a peak at 1500 keypoints, after which it decreases by approximately 20%, while remaining superior to the other methods. At its maximum, VFC surpasses LPM by about 10%, and RANSAC and LMedS by nearly 20% and 35%, respectively. The values for RHO from 500



**Fig. 5.** Filtering quality metrics for SURF: A) Fisher Criterion; B) IQR Separability; C) Whisker Gap; and the SMS coefficients for D)  $\sigma=1$ , E)  $\sigma=2$ , and F)  $\sigma=3$

to 2000 keypoints are higher than those of GMS; however, after 2500 keypoints, the metrics for GMS increase noticeably, and at 5000 keypoints, they are close to those of RANSAC. Despite this, the results for GMS and RHO are significantly lower than those of the other methods. A similar trend is observed for IQR Separability. The maximum also occurs in the 1500-2000 keypoint range, where RANSAC is almost on par with VFC; however, as the number of keypoints increases, its performance declines more rapidly. At more than 3500 keypoints, the GMS method partially overtakes the LMedS method, indicating that the method's efficiency improves with an increase in features.

For the Whisker Gap metric, the ranking of the methods changes. Within the optimal range of 1000-3000 keypoints, LPM shows the largest gap, exceeding VFC by 25-40% and RANSAC by over 1.3 times. However, as the number of keypoints increases further, the advantage shifts to VFC, which demonstrates more stable behavior. The GMS method performs the worst, with its values consistently negative throughout the entire range, indicating an overlap of distribution tails. However, this trend reverses at 2500 keypoints, where the metrics briefly turn positive before gradually decreasing.

The complex SMS coefficients confirm the same pattern, with peak efficiency at 1000-2000 keypoints. VFC leads in all three metrics, especially for  $\sigma = 3$ , where it exhibits the least negative values, indicating greater robustness to anomalous correspondence pairs. This highlights SURF's sensitivity to oversaturation with weak correspondences at large ranges and the importance of using a controlled keypoint limit in practical applications.

The results for the KAZE detector, shown in [Fig. 6](#), demonstrate that increasing the number of keypoints consistently improves filtering quality. Efficiency levels off at 3500-5000 points. In contrast to SURF, the KAZE detector does not exhibit a decline in values at higher keypoint ranges, indicating a higher stability of its features. Although KAZE is weaker than SIFT in absolute metric scores, the performance hierarchy of the filters remains similar.

Based on the Fisher Criterion, metric values increase and plateau around 4000-5000 points. In this range, VFC shows the best performance, with its metric approximately 7% higher than LPM's and 25% higher than RANSAC's, demonstrating superiority over GMS of roughly 1.5 times. Compared to RHO, VFC's improvement is nearly twice as great. This suggests that the KAZE and VFC setup offers the most balanced performance for maximizing class separation with constant variance. A similar trend is observed with IQR Separability, where the central quartile separation stabilizes at approximately 3000 points, again favoring VFC. Meanwhile, LPM, RANSAC, LMedS, and GMS methods fall behind by a few percentage points, and RHO exhibits much lower effectiveness.

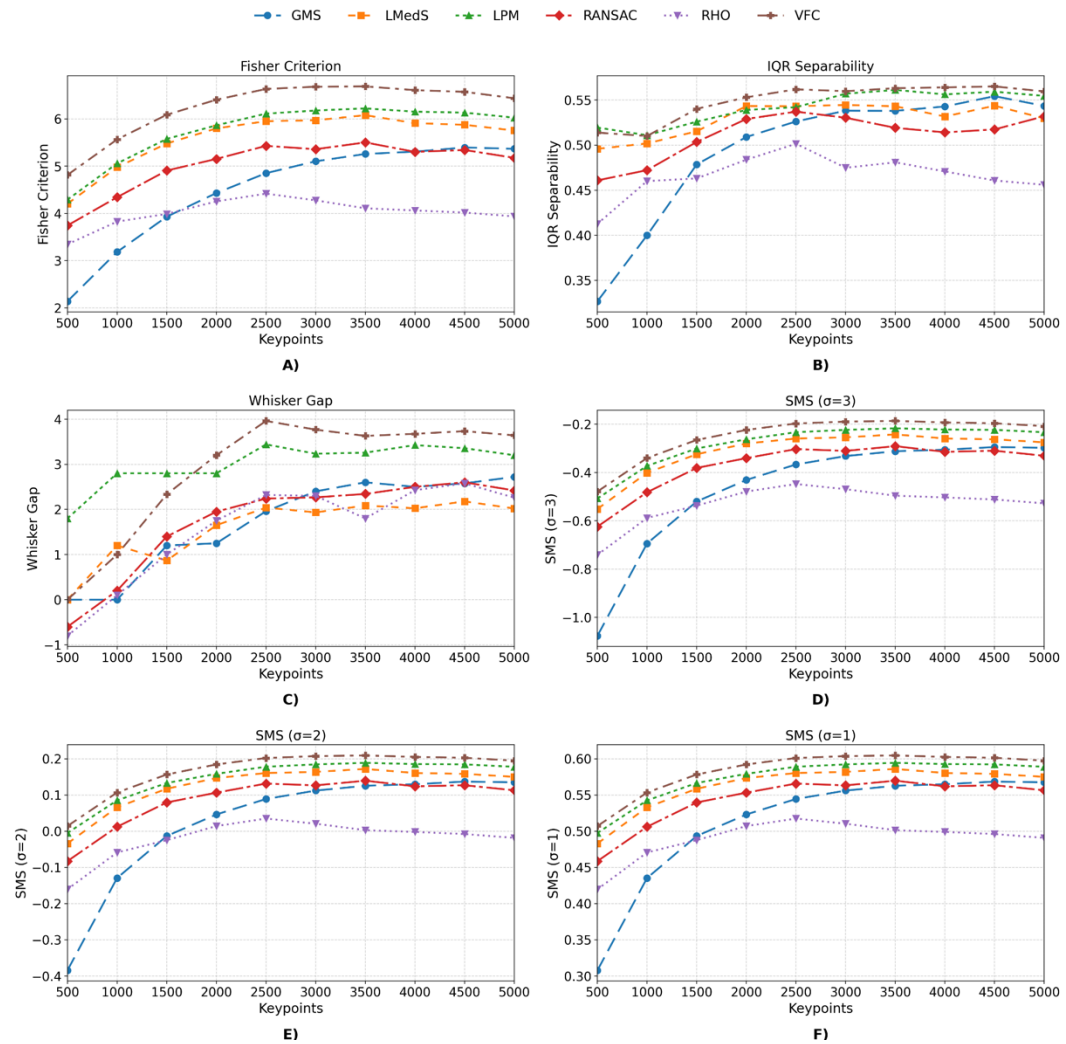
For the Whisker Gap metric, LPM leads in the 500-1500 keypoint range. However, from 2000 to 5000 points, VFC achieves the highest values, approximately 15% higher. Meanwhile, LPM significantly outperforms RANSAC and LMedS, for instance, at 3000 keypoints, by factors of approximately 1.4 and 1.7, respectively. This makes LPM the preferred choice for tasks where reducing the overlap of boundary values is essential.

The complex SMS coefficients consistently demonstrate VFC's leadership. It exceeds its competitors in all three metrics, especially at  $\sigma = 3$ , where its values are 20-30% closer to zero than those of RANSAC and LMedS. This suggests VFC offers the optimal balance between separation and robustness against anomalies. Conversely, RHO shows the worst values, being the furthest from zero.

The results for the binary detector ORB, presented in [Fig. 7](#), show that increasing the keypoint limit consistently improves all filtering quality metrics, reaching a saturation point at 350-5000 keypoints. In terms of curve characteristics, ORB is similar to KAZE, as its efficiency increases with more data, unlike SURF, where a decline occurs after an early peak. In absolute metric values, ORB is expectedly inferior to SIFT.

According to the Fisher Criterion, VFC provides the best values across the entire range. In the zone of stable efficiency, its values are about 5-7% higher than those of LPM, nearly 7% higher than RANSAC and LMedS, and 25-30% higher than GMS. Compared to





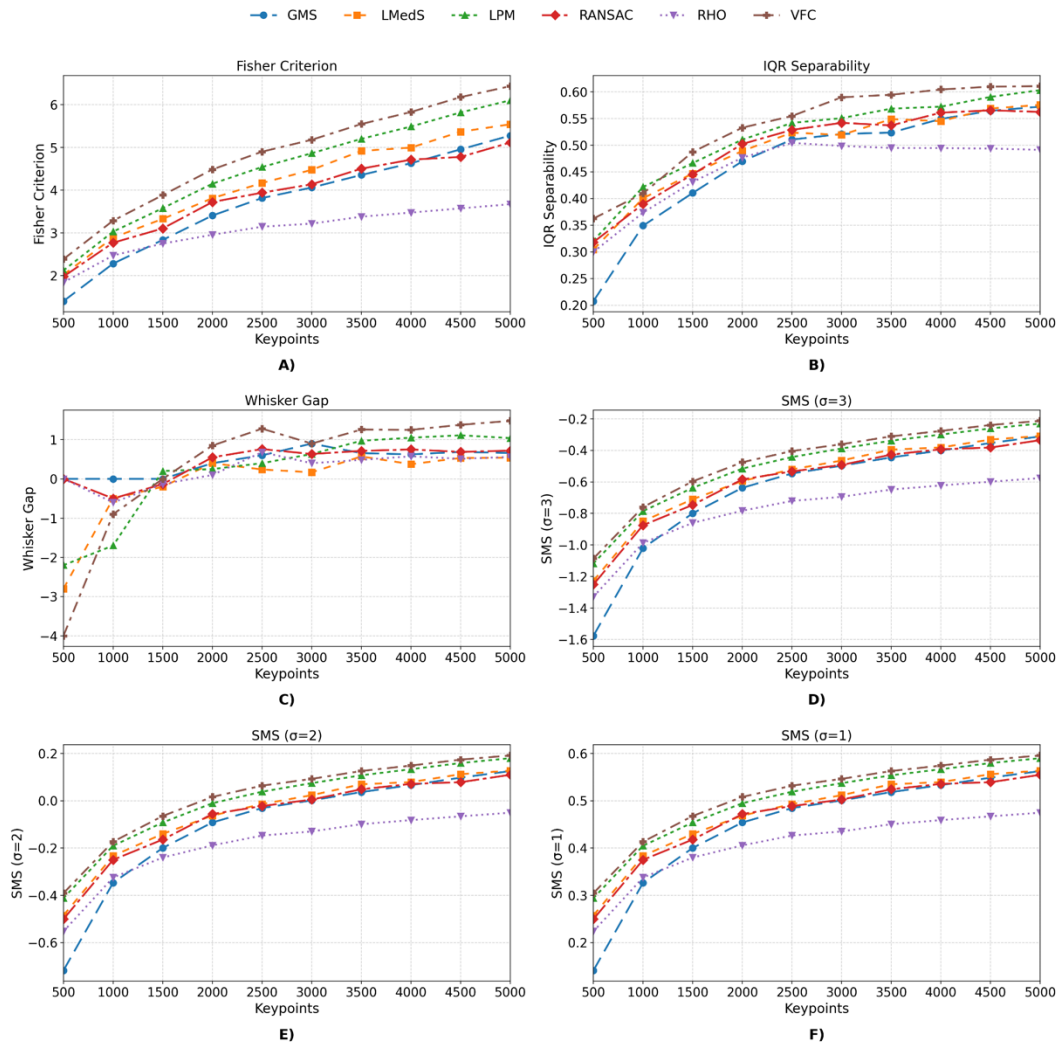
**Fig. 6.** Filtering quality metrics for KAZE: A) Fisher Criterion; B) IQR Separability; C) Whisker Gap; and the SMS coefficients for D)  $\sigma=1$ , E)  $\sigma=2$ , and F)  $\sigma=3$ .

RHO, its advantage exceeds 1.5 times. This indicates that VFC most effectively converts additional keypoints into stable inliers, maintaining a favorable balance between class separation and within-class variance.

The separation of the central quartiles increases with the number of keypoints and stabilizes around 3500 points. In the range of 3500 to 5000 keypoints, VFC outperforms LPM by 2-6%, and RANSAC and LMedS by approximately 8-10%. The GMS method is, on average, 7-14% lower than VFC, and RHO is more than 20% lower. Therefore, when the goal is to maximize the separation of the distribution centers, VFC maintains a consistent advantage, while RANSAC and LMedS serve as alternatives.

For the Whisker Gap metric, a partial overlap of the boundaries occurs across all filtering methods at low keypoint counts from 500 to 1000. Starting around 1500–2000 keypoints, the gap consistently turns positive and grows larger. With more keypoints, LPM and VFC produce the highest values; their gaps are usually 30–60% larger than those of RANSAC and LMedS. GMS and RHO show the smallest gaps, often at least 1.5 to 2 times smaller than the leaders. Practically, this suggests that when boundary value control is vital, it is best to use LPM or VFC with more than 2000 keypoints but avoid LPM with tiny samples.



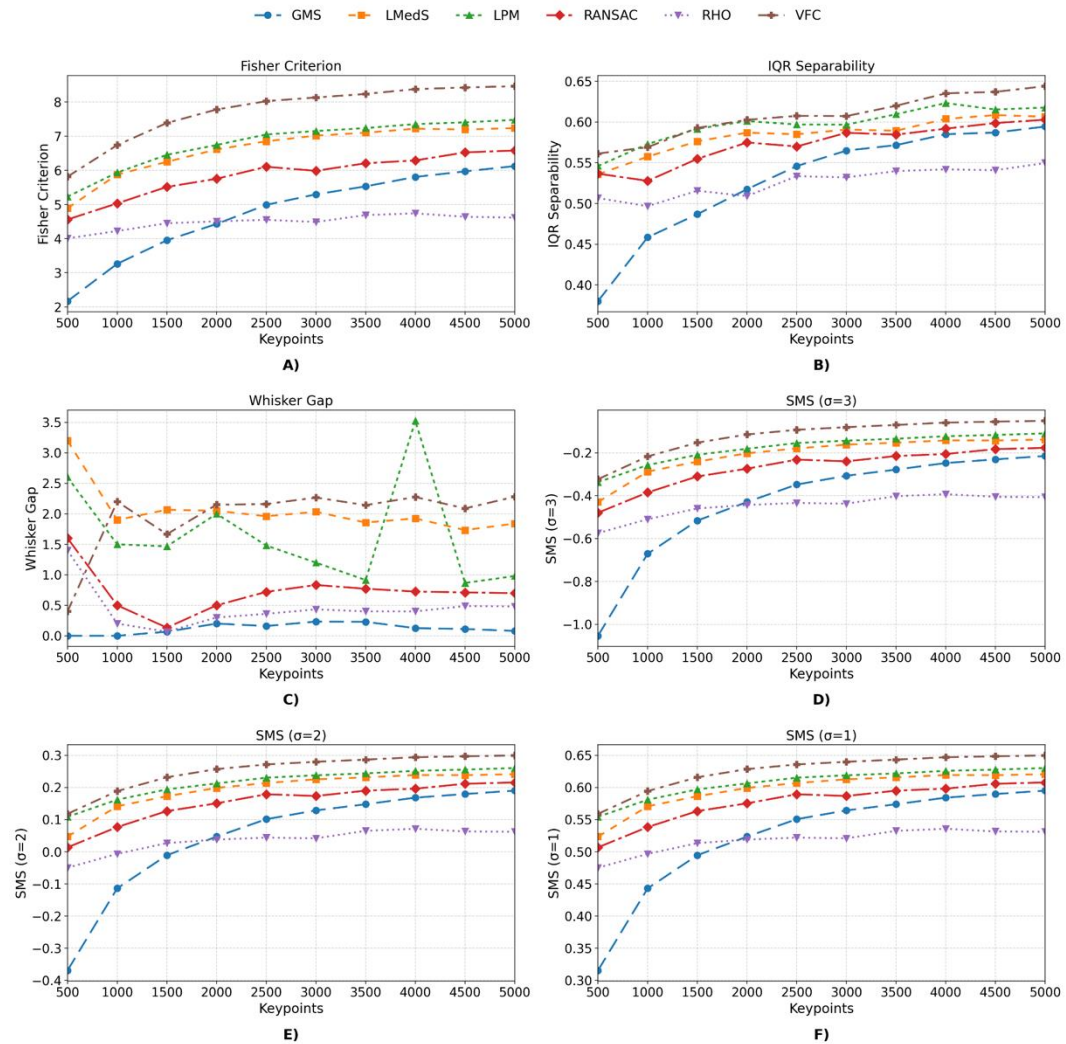


**Fig. 7.** Filtering quality metrics for ORB: A) Fisher Criterion; B) IQR Separability; C) Whisker Gap; and the SMS coefficients for D)  $\sigma=1$ , E)  $\sigma=2$ , and F)  $\sigma=3$ .

The above findings also apply to all SMS coefficient values; the curves reach saturation between 3,500 and 5,000 keypoints. VFC consistently exhibits the highest values across all SMS variants. For  $\sigma = 1$ , its advantage over LPM is modest, about 1-2%, but it surpasses RANSAC and LMedS by approximately 5-10%. When  $\sigma = 2$ , VFC's values turn positive at approximately 1500 keypoints, reaching a peak. For  $\sigma = 3$ , VFC remains 20-30% closer to zero than other methods, with RHO consistently demonstrating the worst results.

The results for the binary detector BRISK, presented in Fig. 8, show that increasing the keypoint limit consistently enhances all filtering quality metrics, which stabilize within the range of 3500-5000 points. Due to the nature of its curves, BRISK resembles KAZE and ORB, as the accumulation of valid correspondences gradually improves class separation without degradation, unlike SURF. In terms of absolute metric levels, BRISK is expectedly lower than SIFT, but at high numbers of points, it achieves a quality comparable to KAZE.

According to the Fisher Criterion, VFC delivers the best results across the entire range. In the saturation zone, its metric is about 14% higher than that of LPM, and nearly 17% and 30% higher than those of LMedS and RANSAC, respectively. Regarding GMS, its values at 5000 keypoints are similar to those of RANSAC and more than 1.5 times lower than



**Fig. 8.** Filtering quality metrics for BRISK: A) Fisher Criterion; B) IQR Separability; C) Whisker Gap; and the SMS coefficients for D)  $\sigma=1$ , E)  $\sigma=2$ , and F)  $\sigma=3$ .

those of VFC. This indicates that VFC most effectively converts additional data into stable correspondences, ensuring the optimal balance between class separation and within-class variance. A similar pattern is observed for IQR Separability, where VFC also shows the best results; in contrast, RHO performs the worst, outperforming GMS only in the range from 500 to 1500 keypoints.

For the Whisker Gap metric, the LMedS method starts strongest at the beginning of the range, then mostly maintains second place. Later, in the 1500-3000 keypoint range, VFC takes the top spot, with results about four times better than those of RANSAC. At 4000 keypoints, a noticeable peak occurs for LPM, where its results are roughly 35% higher than those of VFC. The RHO values are a few percent above GMS's, but their values are nearly four times lower than those of VFC and LMedS.

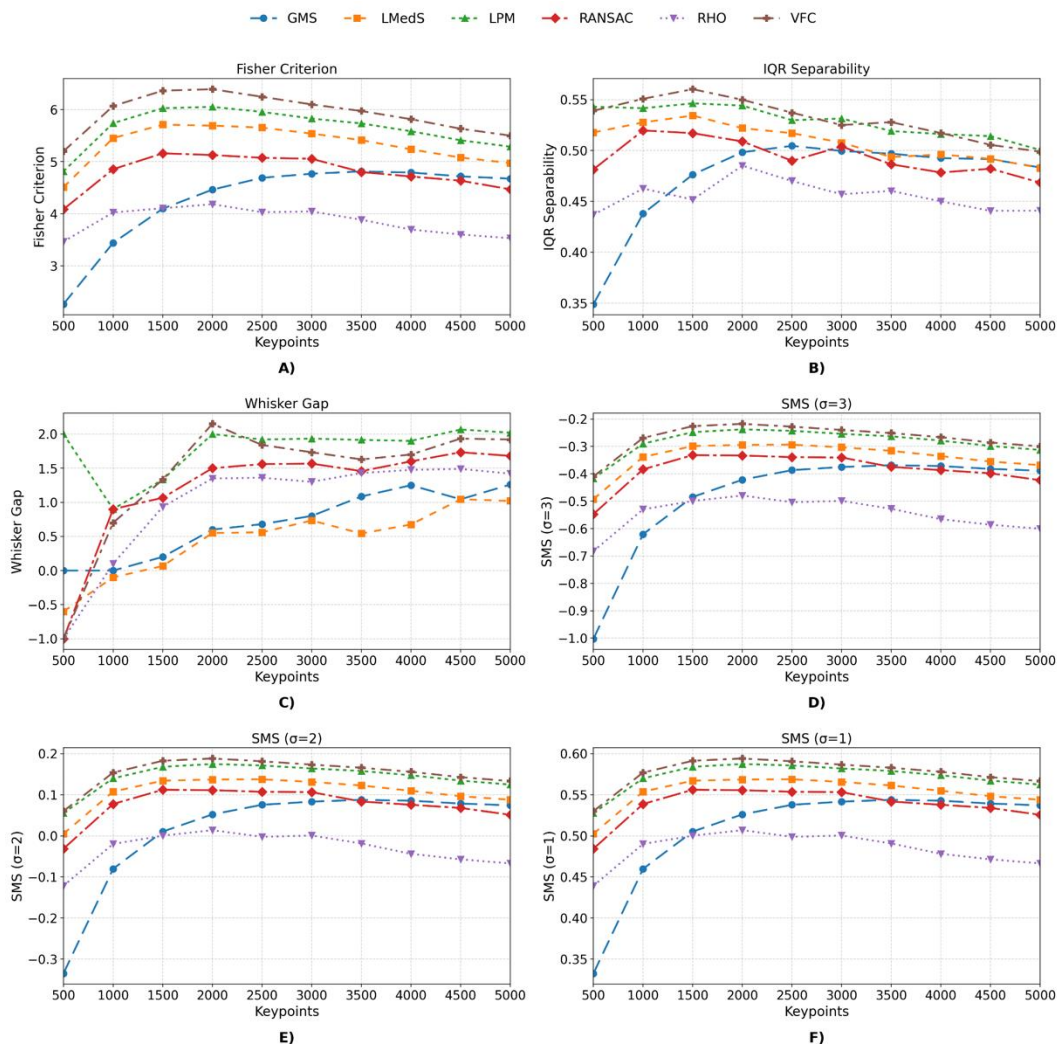
The complex SMS coefficients consistently confirm VFC's leadership across all metrics. Its advantage over LPM for  $\sigma=1$  is approximately 3-4%, and it is about 5% higher than LMedS. For  $\sigma=2$  values, VFC's metrics are roughly twice as good as RANSAC's. According to the  $\sigma=3$  metric, VFC's results are closer to zero than those of the other methods. The RHO method has the lowest values for all SMS variations, except in the

interval between 500 and 2000 keypoints, where it outperforms GMS. Additionally, the RHO results are the most consistent across the entire range of keypoints.

The results for the AKAZE detector, as shown in Fig. 9, exhibit a different pattern than those for KAZE. Efficiency quickly rises at the start, peaks between 1500 and 2500 keypoints, and then generally stabilizes or declines slowly. This early peak behavior is more similar to SURF than to KAZE or ORB.

According to the Fisher Criterion, VFC yields the best result. Its metric peaks at 2000 keypoints, then decreases slightly by about 14% but remains the leader. At its peak, VFC exceeds LPM by 6% and LMedS by approximately 12%, and is almost twice as good as RHO. A different pattern was observed for the GMS method; after 3500 keypoints, its results level off, and the values are somewhat higher than those of RANSAC.

For the IQR Separability metric, the highest values are observed at 1500-2000 keypoints, where VFC has a slight advantage over LPM of 2.5%, although after 4000 keypoints, LPM takes the lead. LMedS outperforms RANSAC by 2-4%. GMS exhibits an interesting trend: its values are the lowest from 500 to 1500 points; at 2500 points, it surpasses RANSAC; and from 3500 keypoints to the end of the range, it competes with the



**Fig. 9.** Filtering quality metrics for AKAZE: A) Fisher Criterion; B) IQR Separability; C) Whisker Gap; and the SMS coefficients for D)  $\sigma=1$ , E)  $\sigma=2$ , and F)  $\sigma=3$ .

LMedS method. The RHO method shows the lowest values and tends to degrade after 2500 keypoints.

For the Whisker Gap metric, the ranking of methods shifts. Within the optimal range of 3000-5000 points, LPM exhibits the most significant gap, outperforming VFC by 5-20% and RANSAC by more than 1.3 times. The LMedS and GMS methods yield the worst results; for LMedS, this is particularly evident at 2500-3000 keypoints, where a significant gap is observed compared to other methods. In contrast, RHO shows values close to RANSAC's after 1500 points.

Across all SMS variations, VFC and LPM consistently rank first and second, with a slight difference of approximately 0.7-1%, peaking at 1500-2500 points. RHO performs the worst among the methods. The trend patterns for the AKAZE method resemble those seen with SURF, where GMS at over 1500 keypoints improves its metrics by nearly 1.5 times compared to RHO. At 4000-5000 keypoints, it surpasses RANSAC and approaches results similar to LMedS.

The results help in developing strategies for selecting the best configuration based on the specific task. For real-time applications, combining fast binary detectors such as ORB or BRISK with the VFC filter is effective, especially when limiting keypoints to the saturation zone for improved performance. LPM is recommended to reduce the overlap of distribution tails. If the goal is the highest matching quality, SIFT paired with VFC and a constrained keypoint limit is ideal to manage computational costs. When features are limited, SURF and AKAZE perform best near their efficiency peaks, offering an optimal quality-to-data ratio. Regardless of the detector used, RANSAC should be applied cautiously. Although it sometimes excels in the IQR Separability metric, its high computational demands and poor scalability often make it impractical for most modern systems.

## CONCLUSION

This study presented a comparative analysis of various combinations of keypoint detection and filtering methods. Consistent patterns emerged: binary descriptors are characterized by high computational efficiency, whereas methods using floating-point descriptors tend to produce more informative but computationally intensive correspondences.

The analysis of filtering methods showed that VFC most often provides the most balanced separation of distributions and proves to be robust against noisy correspondences. At the same time, LPM is especially effective at controlling boundary cases by creating the greatest distance between the extreme boundaries. RANSAC and LMedS remain valuable as classic benchmarks, and GMS and RHO offer fast and lightweight alternatives. Despite differences between detectors, the relative ranking of the methods stays consistent, highlighting the universality of the identified patterns.

For tasks that require maximum separability, the best approach is to combine detectors with a stable saturation plateau, such as SIFT or KAZE, along with the VFC filter. When boundary error control is necessary, LPM is recommended. If processing speed is the primary concern, then combinations like ORB or BRISK with efficient filters, such as VFC or RHO, are suitable. The metrics used, including the proposed SMS coefficient, demonstrated their effectiveness in delivering a thorough assessment of filtering quality.

Thus, the research demonstrates that filtering correspondences in computer vision tasks cannot be considered independently of the choice of detector and keypoint parameters. A comprehensive approach allows for not only objective comparisons of methods but also the development of practical strategies to balance quality and computational costs. The main contribution of this work is to provide valuable recommendations for selecting optimal combinations for various applied scenarios. Future research could include integrating deep learning methods, employing other models to verify the geometric consistency of matches, and testing under more challenging scene conditions, particularly with varying scales, camera viewpoints, and changes in illumination.



## ACKNOWLEDGMENTS AND FUNDING SOURCES

The author(s) received no financial support for the research, writing, and/or publication of this article.

## CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any.

## AUTHOR CONTRIBUTIONS

Conceptualization, [A.F., Yu.F.]; methodology, [A.F., Yu.F.]; validation, [A.F., Yu.F.]; writing – original draft preparation, [A.F.]; writing – review and editing, [A.F., Yu.F.]; supervision, [Yu.F.].

All authors have read and agreed to the published version of the manuscript.

## REFERENCES

- [1] Zhou, L., Wu, G., Zuo, Y., Chen, X., & Hu, H. (2024). A comprehensive review of vision-based 3D reconstruction methods. *Sensors*, 24(7), 2314. <https://doi.org/10.3390/s24072314>
- [2] Ye, Z., Bao, C., Zhou, X., Liu, H., Bao, H., & Zhang, G. (2023). EC-SfM: Efficient covisibility-based structure-from-motion for both sequential and unordered images. *IEEE Transactions on Circuits and Systems for Video Technology*. <https://doi.org/10.1109/TCSVT.2023.3285479>
- [3] Soltanpour, S., & Joslin, E. (2025). A survey on feature-based and deep image stitching. In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Volume 3: VISAPP (pp. 777–788). <https://doi.org/10.5220/0013368500003912>
- [4] Herrera-Granda, E. P., Berrones-González, A., & Aguilar, W. (2024). Monocular visual SLAM, visual odometry, and structure from motion: A review. *Heliyon*, 10(9), e37356. <https://doi.org/10.1016/j.heliyon.2024.e37356>
- [5] Abaspor Kazerouni, I., Fitzgerald, L., Dooly, G., & Toal, D. (2022). A survey of state-of-the-art on visual SLAM. *Expert Systems with Applications*, 205, 117734. <https://doi.org/10.1016/j.eswa.2022.117734>
- [6] Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395. <https://doi.org/10.1145/358669.358692>
- [7] Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79(388), 871–880. <https://doi.org/10.1080/01621459.1984.10477105>
- [8] Chum, O., & Matas, J. (2005). Matching with PROSAC - Progressive sample consensus. *Proceedings of CVPR 2005* (pp. 220–226). <https://doi.org/10.1109/CVPR.2005.221>
- [9] Bian, J.-W., Lin, W.-Y., Matsushita, Y., Yeung, S.-K., Nguyen, T.-D., & Cheng, M.-M. (2019). GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. *International Journal of Computer Vision*, 128(6), 1580–1593. <https://doi.org/10.1007/s11263-019-01280-3>
- [10] Ma, J., Zhao, J., Tian, J., Yuille, A. L., & Tu, Z. (2014). Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, 23(4), 1706–1721. <https://doi.org/10.1109/TIP.2014.2307478>
- [11] Ma, J., Zhao, J., Jiang, J., Zhou, H., & Guo, X. (2018). Locality Preserving Matching. *International Journal of Computer Vision*, 127(5), 512–531. <https://doi.org/10.1007/s11263-018-1117-z>

- [12] Liao, Y., Di, Y., Zhu, K. et al. Local feature matching from detector-based to detector-free: a survey. *Appl Intell* 54, 3954–3989 (2024). <https://doi.org/10.1007/s10489-024-05330-3>
- [13] Isik, M. (2024). Comprehensive empirical evaluation of feature extractors in computer vision. *PeerJ Computer Science*, 10, e2415. <https://doi.org/10.7717/peerj-cs.2415>
- [14] S. A. Khan Tareen and R. H. Raza, "Potential of SIFT, SURF, KAZE, AKAZE, ORB, BRISK, AGAST, and 7 More Algorithms for Matching Extremely Variant Image Pairs," *2023 4th International Conference on Computing, Mathematics and Engineering Technologies*, pp. 1-6, 2023. <https://doi.org/10.1109/iCoMET57998.2023.10099250>
- [15] Baráth, D., & Matas, J. (2022). Graph-Cut RANSAC: Local optimization on spatially coherent structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9), 4961–4974. <https://doi.org/10.1109/TPAMI.2021.3071812>
- [16] Yunge Cui, Yingming Hao, Qingxiao Wu, et al. "An Optimized RANSAC for The Feature Matching of 3D LiDAR Point Cloud". In *Proceedings of the 2024 5th International Conference on Computing, Networks and Internet of Things (CNIOT '24)*. Association for Computing Machinery, New York, NY, USA, pp. 287–291, 2024. <https://doi.org/10.1145/3670105.3670153>
- [17] Rodríguez, M., Facciolo, G., & Morel, J.-M., "Robust Homography Estimation from Local Affine Maps". *Image Processing On Line*, 13, 65–89, 2023. <https://doi.org/10.5201/ipol.2023.356>
- [18] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
- [19] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3), 346–359. <https://doi.org/10.1016/j.cviu.2007.09.014>
- [20] Alcantarilla, P. F., Bartoli, A., & Davison, A. J. (2012). KAZE features. In *ECCV 2012* (LNCS 7577, pp. 214–227). [https://doi.org/10.1007/978-3-642-33783-3\\_16](https://doi.org/10.1007/978-3-642-33783-3_16)
- [21] Alcantarilla, P. F., Nuevo, J., & Bartoli, A. (2013). Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *BMVC 2013* (pp. 1–11). <https://doi.org/10.5244/C.27.13>
- [22] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *ICCV 2011* (pp. 2564–2571). <https://doi.org/10.1109/ICCV.2011.6126544>
- [23] Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. In *ICCV 2011* (pp. 2548–2555). <https://doi.org/10.1109/ICCV.2011.6126542>
- [24] Image Matching Challenge organizers. (2021). *Data — Image Matching Challenge 2021 (Phototourism subset)*. University of British Columbia. <https://www.cs.ubc.ca/research/image-matching-challenge/2021/data>
- [25] Bazargani, H., Bilaniuk, O., & Laganière, R. (2018). A fast and robust homography scheme for real-time planar target detection. *Journal of Real-Time Image Processing*, 15(4), 739–758. <https://doi.org/10.1007/s11554-015-0508-4>
- [26] Fesiuk, A., & Furgala, Y. (2025). Keypoint matches filtering in computer vision: Comparative analysis of RANSAC and USAC variants. *International Journal of Computing*, 24(2), 343–350. <https://doi.org/10.47839/ijc.24.2.4018>
- [27] M. Ivashechkin, D. Baráth, J. Matas, "USACv20: Robust Essential, Fundamental and Homography Matrix Estimation," 2021. <https://doi.org/10.48550/arXiv.2104.05044>
- [28] Howse, Joseph, and Joe Minichino. "Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning.", *Packt Publishing Ltd*, 2020.



- [29] A. Fesiuk and Y. Furgala, "The Impact of Parameters on the Efficiency of Keypoints Detection and Description," *2023 IEEE 13th International Conference on Electronics and Information Technologies (ELIT)*, Lviv, Ukraine, pp. 261-264, 2023.  
<https://doi.org/10.1109/ELIT61488.2023.10310866>

## ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ ДЕТЕКТОРІВ ОСОБЛИВИХ ТОЧОК ТА МЕТОДІВ ФІЛЬТРАЦІЇ У ЗІСТАВЛЕННІ ЗОБРАЖЕНЬ

Андрій Фесюк\*, Юрій Фургала

Львівський національний університет імені Івана Франка,  
 вул. Драгоманова 50, 79005 Львів, Україна

### АНОТАЦІЯ

**Вступ.** У сучасному комп'ютерному зорі точність і надійність зіставлення зображень значною мірою визначається якістю обробки локальних ознак. Хибні відповідності, що виникають через зміну масштабу, освітлення чи повторювані структури, здатні зруйнувати геометричну модель сцени. Тому важливим етапом стає застосування алгоритмів фільтрації, здатних відокремити інформативні збіги від шумових. Попри значний прогрес, більшість досліджень аналізують лише окремі комбінації детекторів та методів фільтрації, що не дозволяє сформувати цілісне уявлення про їхню взаємодію.

**Матеріали та методи.** Для дослідження цієї проблеми проведено експерименти, в межах яких виконано вибір репрезентативної підмножини з датасету Photo Tourism, виявлення й опис ключових точок, початкове зіставлення, фільтрацію викидів та кількісну оцінку. Порівняння проведено для детекторів SIFT, SURF, KAZE, AKAZE, ORB, BRISK у поєднанні з методами фільтрації RANSAC, LMedS, RHO, GMS, VFC, LPM. Для оцінки застосовано набір метрик, зокрема Fisher Criterion, IQR Separability, Whisker Gap та власну розробку SMS.

**Результати.** Дослідження показало, що продуктивність суттєво відрізняється між детекторами: бінарні дескриптори забезпечують значно вищу швидкість, тоді як методи з дескрипторами з плаваючою комою демонструють кращу інформативність ціною більших витрат. Ієрархія методів фільтрації виявилася стабільною для всіх конфігурацій: найвищу якість за метриками роздільної здатності демонструє VFC, тоді як LPM забезпечує найбільший розрив між крайніми межами розподілів. RANSAC і LMedS залишаються класичними орієнтирами, а GMS і RHO є швидкими компромісними варіантами.

**Висновки.** Отримані результати показують, що ефективність зіставлення зображень визначається саме поєднанням детектора, кількості ключових точок та методу фільтрації. Комплексний підхід дозволяє обґрунтовано обирати стратегії під конкретні задачі: від застосунків, де вирішальною є швидкість, до сценаріїв, де важливою є максимальна роздільна здатність чи контроль граничних помилок. Запропонований аналіз та використані метрики формують основу для подальших досліджень і вдосконалення практичних систем комп'ютерного зору.

**Ключові слова:** виявлення ознак, опис особливих точок, співпадіння, фільтрація.

UDC 004.89

## PREDICTING QUANTITATIVE CHARACTERISTICS OF AIR POLLUTION

Volodymyr Hura<sup>1</sup> \*, Igor Olenych<sup>1</sup> , Oleh Sinkevych<sup>1</sup> ,  
Oksana Ostrovska<sup>2</sup> , Roman Shuvar<sup>2</sup> 

<sup>1</sup> Radioelectronic and Computer Systems Department,

<sup>2</sup> Department of System Design,

Ivan Franko National University of Lviv,

50 Dragomanov Str., 79005 Lviv, Ukraine

Hura, V., Olenych, I., Sinkevych, O., Ostrovska, O., Shuvar, R. (2025). Predicting quantitative characteristics of air pollution. *Electronics and Information Technologies*, 31, 89–104  
<https://doi.org/10.30970/eli.31.8>

### ABSTRACT

**Background.** Rapid industrialization and urbanization have escalated air pollution, posing significant health and environmental threats. Accurate prediction of quantitative air pollution characteristics (like pollutant concentrations or Air Quality Index) is critical for effective monitoring and mitigation strategies. Fuzzy Logic (FL) provides a robust computational intelligence framework adept at handling the inherent uncertainty, imprecision, and non-linear dynamics present in atmospheric systems.

**Materials and Methods.** The study explores the application of Fuzzy Logic (FL) for improving the prediction of hourly PM<sub>2.5</sub> concentrations by adding new input features to data obtained using localized monitoring data from Variazh, for 2024. A key aspect involves feature engineering, where a secondary Fuzzy Inference System (FIS) was developed to derive Pasquill atmospheric stability class based on measured meteorological inputs (wind speed, solar radiation, cloud cover). This derived stability class was then incorporated as an additional input feature into the primary Mamdani-type FIS designed for PM<sub>2.5</sub> prediction correction.

**Results and Discussion.** The inclusion of the fuzzy-derived atmospheric stability class as an input feature improved the performance of the PM<sub>2.5</sub> prediction models tested (XGBoost, LightGBM). Models incorporating this engineered feature achieved high accuracy ( $R^2 > 0.98$ ), particularly showing enhanced capability during stable atmospheric conditions. This highlights the value of incorporating physically relevant, engineered features derived via interpretable methods like FIS into data-driven air quality models.

**Conclusion.** Fuzzy Logic proves to be a valuable tool for effective feature engineering in air pollution modeling. Deriving parameters such as atmospheric stability class via an interpretable, rule-based FIS can enrich datasets and enhance the accuracy of subsequent predictive models, offering a practical approach to improving air quality forecasting, especially when direct measurements of complex parameters are unavailable.

**Keywords:** air pollution, fuzzy logic, forecasting, pollutant concentration, fuzzy inference system, quantitative prediction, machine learning.

### INTRODUCTION

Air pollution is one of the most important environmental problems of the 21st century, mainly driven by rapid global industrialization, urbanization, and increased transportation activities [1]. Among the various harmful pollutants, fine particulate matter (PM<sub>2.5</sub>) is of particular concern. Based on combustion sources (vehicles, power plants, residential heating), industrial processes, and secondary formation of precursor gases (such as SO<sub>2</sub>,



© 2025 Volodymyr Hura et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and information technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

NO<sub>x</sub>, and VOCs), exposure to PM<sub>2.5</sub> poses a serious risk to human health [2, 3]. In addition to its direct health effects, PM<sub>2.5</sub> negatively impacts ecosystems, reduces visibility, and contributes to climate change [4].

Consequently, the ability to accurately predict PM<sub>2.5</sub> concentrations and understand its dispersion patterns is crucial. Timely and reliable forecasts of quantitative characteristics, particularly PM<sub>2.5</sub> levels and the resulting Air Quality Index (AQI), are essential for implementing effective mitigation strategies, issuing targeted public health warnings, informing policy decisions regarding emission controls, and assessing the impact of environmental regulations [5]. However, predicting PM<sub>2.5</sub> concentrations is inherently complex. Its levels are governed by intricate atmospheric processes, including direct emissions, chemical transformations in the atmosphere (secondary aerosol formation), and transport, all significantly influenced by highly variable meteorological conditions (like wind speed, temperature, humidity, boundary layer height), fluctuating emission sources, and geographical factors [6]. These factors interact in strongly non-linear ways, and the available measurement and emissions data often contain significant uncertainty and imprecision, making traditional linear or deterministic modeling approaches particularly challenging for PM<sub>2.5</sub> [7].

Conventional statistical and deterministic models often struggle to adequately capture the vagueness, complex non-linearities, and stochastic nature inherent in PM<sub>2.5</sub> dynamics [8]. These models frequently require precise mathematical formulations of system behavior, which are often difficult, if not impossible, to derive accurately for complex, open environmental systems like the atmosphere. This limitation highlights the need for alternative modeling techniques capable of handling such complexities effectively. Computational intelligence methods, particularly Fuzzy Logic (FL), have emerged as powerful tools for environmental modeling precisely because they excel at managing uncertainty, linguistic ambiguity, and complex input-output relationships without requiring a complete, explicit mathematical understanding of the underlying physical processes [9]. Unlike traditional Boolean (crisp) logic, where something is either true or false, fuzzy logic, based on fuzzy set theory introduced by Lotfi Zadeh [10], allows for degrees of membership. This means a variable can belong partially to different qualitative sets simultaneously (e.g., a specific temperature might be considered 70% 'warm' and 30% 'hot'). FL utilizes linguistic variables (e.g., 'low' wind speed, 'high' traffic density, 'stable' atmosphere) defined by membership functions, and employs a system of IF-THEN fuzzy rules to map input conditions to output predictions. This structure allows for reasoning with imprecise information, mirroring human-like decision-making under uncertainty. Its proven success in diverse fields requiring the management of imprecise information, such as industrial control systems and complex decision support frameworks, further motivates its application to environmental challenges [9]. This characteristic makes it particularly well-suited for modeling complex systems like PM<sub>2.5</sub> pollution, where precise mathematical descriptions are difficult to formulate, data is incomplete or noisy, and key relationships are inherently fuzzy or non-linear.

## MATERIALS AND METHODS

### Study Design and Data Collection

The primary objective was to improve the efficiency of air pollution forecasting by machine learning models by supplementing the feature vector with an additional feature. To determine the additional feature was the design of a Fuzzy Inference System (FIS) to evaluate atmospheric stability based on localized meteorological measurements. Unlike traditional computational models, this approach leverages fuzzy logic to capture and model complex, non-linear relationships under conditions of uncertainty, utilizing site-specific data from the Variazh (**Fig. 1**). The study sought to validate the FIS by assessing its performance in classifying atmospheric stability across various local weather conditions recorded in 2024.

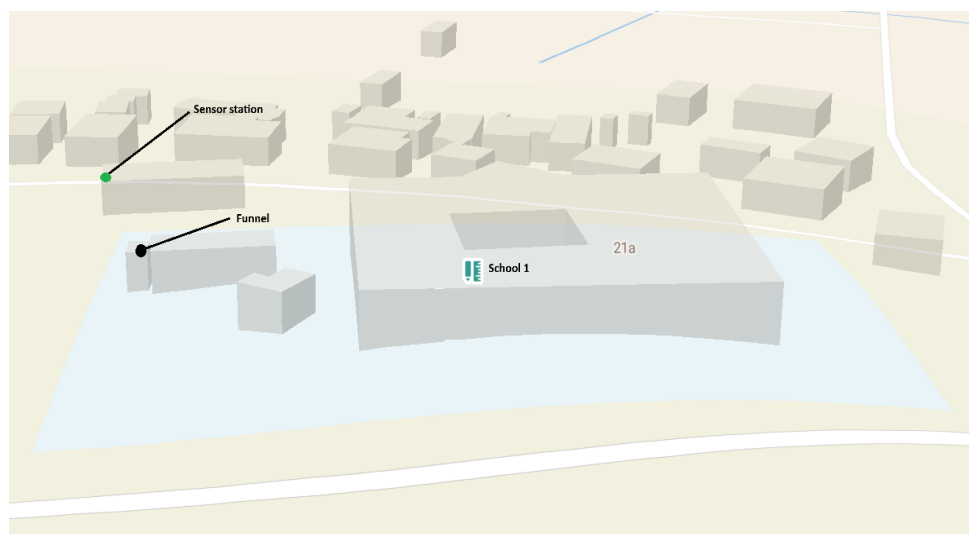


Fig. 1. Air quality information for one station.

Achieving the study's objective involved a methodology with several integrated stages. Initially, data collection and preprocessing were performed using a comprehensive hourly dataset from a dedicated monitoring station in Variazh, covering the entirety of 2024. This dataset comprised essential meteorological variables (wind speed, solar radiation, cloud cover) and temporal factors (hour, day, month), which were processed to remove anomalies and missing values, thereby ensuring data integrity. Meteorological parameters and temporal proxies were selected for their direct influence on atmospheric stability. These predictors, combined with historical site-specific data, provided the foundation for defining the fuzzy system's input variables. The development process involved the selection and definition of input variables, the formulation of appropriate linguistic terms, and the creation of membership functions tailored to the local conditions observed in Variazh.

This step evaluated the model's ability to generalize across diverse atmospheric conditions. The results analyzed the constructed fuzzy rules to evaluate how input variables such as wind speed, solar radiation, and cloud cover influenced stability classes. This provided insights into localized meteorological dynamics and their impact on atmospheric stability. These results explored the potential of the developed FIS as a practical tool for assessing atmospheric stability. The localized focus and integration of high-resolution site-specific data emphasized the model's applicability for operational air quality management.

The data utilized a retrospective analysis approach, leveraging the complete hourly dataset for 2024 to train and evaluate the FIS across a full annual cycle.

The geographically focused scope, centered on the Variazh, benefited from the availability of high-quality, site-specific data recorded at a dedicated monitoring station. This detailed focus enabled the development of localized ML models tailored to the unique climatic and environmental characteristics of the site.

#### Air Quality and Emission Data

The primary air quality parameter targeted for predicting the hourly average concentration of fine particulate matter ( $\text{PM}_{2.5}$   $\mu\text{g}/\text{m}^3$ ). Standard quality assurance and control procedures were assumed to be applied during data collection, and the preprocessing steps outlined addressed potential gaps or anomalies [12].

Hourly meteorological data, collected concurrently with the air quality parameters, were obtained from the monitoring station located in the Variazh (Fig. 1). The specific meteorological variables measured and utilized as inputs for the prediction models are described in Table 1, and the correlation matrix is shown in Fig. 2.

Table 1. Air quality information for one station for the year 2024

Characteristic	count	mean	min	25%	50%	75%	max	std
pm10	8784	11.2	1.7	6.7	9.2	13.3	78.4	7.5
pm2_5	8784	9.1	1.3	5.3	7.5	11.2	61.8	5.9
temperature	8784	11.0	-17.3	3.4	10.7	18.4	34.0	9.1
wind_speed	8784	3.6	0.0	2.2	3.3	4.6	11.5	1.8
wind_direction	8784	196.2	1.0	123.0	195.0	277.0	360.0	94.1
wind_gusts	8784	6.7	0.4	4.3	6.2	8.5	20.1	3.1
humidity	8784	75.3	23.0	64.0	79.0	89.0	100.0	16.5
pressure	8784	1016.8	987.4	1011.3	1016.2	1021.9	1041.6	8.8
solar_radiation	8784	144.0	0.0	0.0	5.0	223.0	886.0	220.7
cloud_cover	8784	31.3	0.0	0.0	6.0	72.0	100.0	40.4

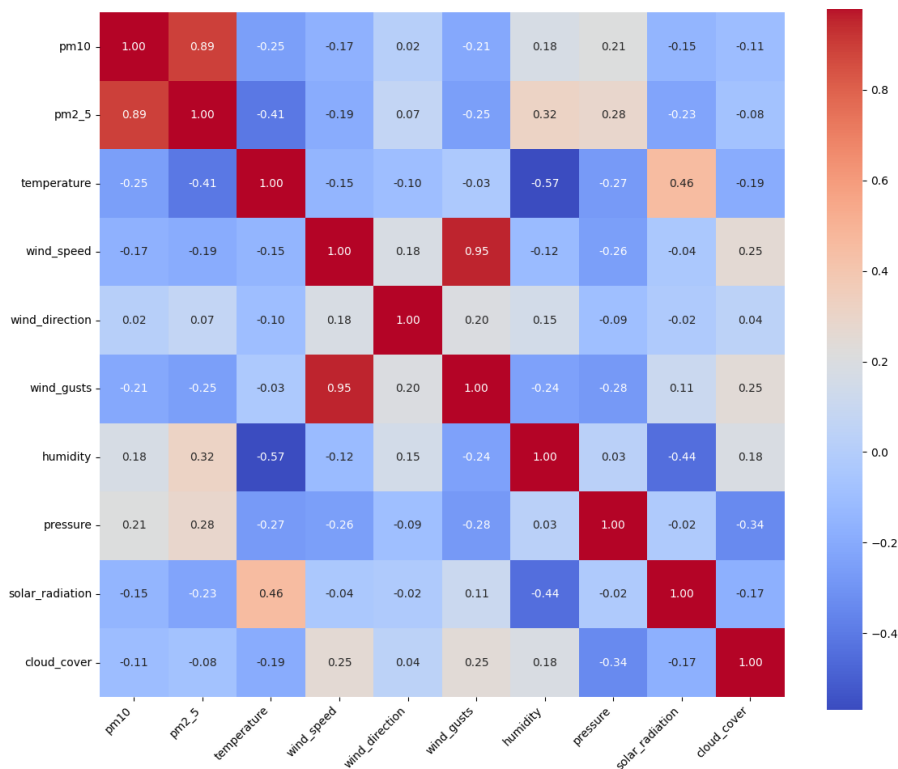
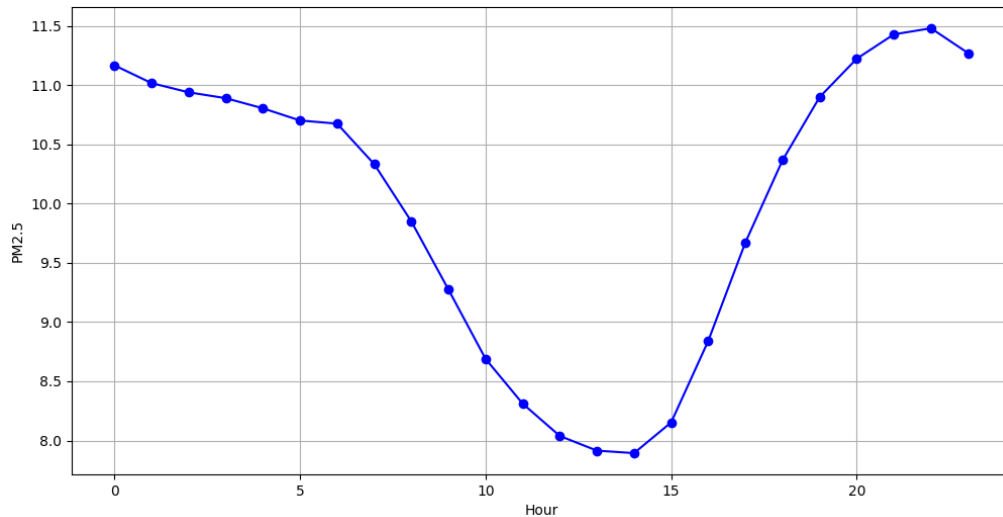


Fig. 2. Correlation matrix for selected parameters.

These temporal variables serve as inputs, allowing the predictive models to implicitly learn and incorporate typical periodic variations in local emissions and atmospheric conditions. The collected PM2.5 data serves as the output (target) variable for the predictive models being evaluated. The input variable set comprises the meteorological parameters measured at the station (temperature, humidity, wind\_speed, wind\_direction, wind\_gusts, cloud\_cover, solar\_radiation, pressure), the temporal variables (hour, day, month).

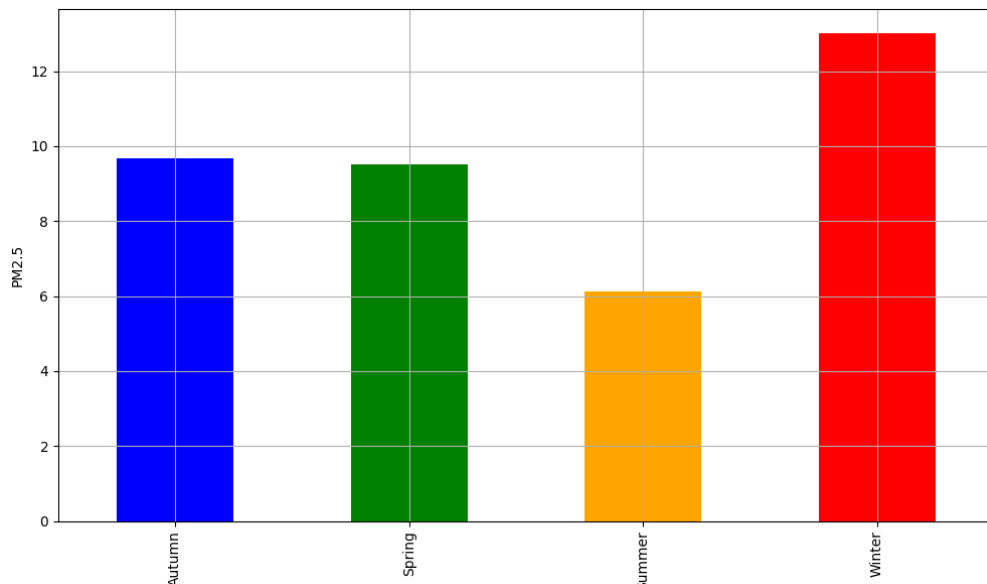
Exploratory data analysis revealed distinct temporal patterns in PM2.5 concentrations at the Variazh monitoring site during 2024. **Fig. 3** presents the average PM2.5 levels aggregated by season and by hour of the day.



**Fig. 3.** Level of PM2.5 based on hour.

The curve in **Fig. 3** illustrates a characteristic daily cycle in average hourly PM2.5 concentrations. Levels tend to be higher during nighttime and early morning hours, decrease during the day, reaching a minimum in the early afternoon, and then increase again towards the evening and night.

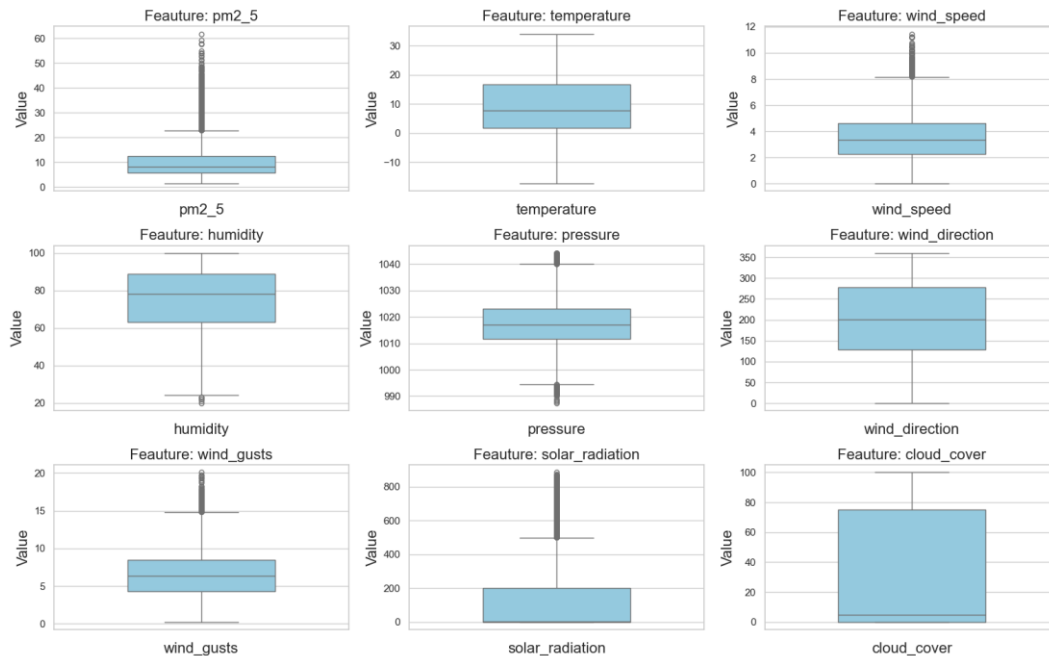
The bar chart in **Fig. 4** shows clear seasonal differences in average PM2.5 levels. Winter exhibited the highest average concentrations, followed by Autumn and Spring, while Summer showed the lowest average levels.



**Fig. 4.** Level of PM2.5 based on season.



The typical ranges and distributions of these parameters for the study period are visualized using boxplots in **Fig. 5**. These parameters are known to significantly influence the formation, transport, dispersion, and deposition of PM2.5 [7].



**Fig. 5.** Boxplots for the selected parameters.

Wind speed and direction govern pollutant transport. Temperature and solar radiation strongly influence atmospheric stability and convection; higher temperatures and solar radiation generally lead to greater atmospheric instability and vertical mixing (associated with convection), which aids dispersion, while lower values contribute to stability and pollutant trapping. Humidity influences particle properties and growth, and pressure systems are associated with large-scale air mass movements and overall atmospheric stability.

This comprehensive set of locally measured meteorological variables provides crucial information for the predictive models to capture the complex interplay between weather conditions and PM2.5 concentrations. This data underwent the same quality checks and preprocessing steps (handling missing values, etc.) as described in section 2.1, ensuring a consistent and reliable input dataset.

### Fuzzy Inference System Approach Implementation

A Fuzzy Inference System (FIS) is a computational framework based on the principles of fuzzy set theory, fuzzy IF-THEN rules, and fuzzy reasoning [10]. It provides a methodology for mapping a set of input variables to an output variable by utilizing linguistic variables and mimicking human-like approximate reasoning. This approach was selected because of its suitability for modeling complex, non-linear systems like air pollution dynamics, where relationships between variables may be imprecise or uncertain, and where the interpretability of the model is a desirable characteristic [9]. Unlike "black box" machine learning models, the rule-based structure of an FIS can potentially offer insights into how input conditions influence PM2.5 predictions.

A Mamdani-type Fuzzy Inference System (FIS) was selected for this prediction task, primarily due to its intuitive nature and the interpretability offered by its linguistic output variables. The development process followed these key steps.

The input variables for the output stability class were the meteorological parameters: wind speed, solar radiation, and cloud cover. The single output variable was the predicted atmospheric stability class, categorized using Pasquill stability classes (A, B, C, D, E, F), which represent different levels of atmospheric stability ranging from highly unstable (A) to highly stable (F) [17].

Each input variable (wind speed, solar radiation, and cloud cover) was fuzzified by defining a set of linguistic terms. For example:

- Wind Speed: 'Very Low', 'Low', 'Medium', 'High', 'Very High'
- Solar Radiation: 'Low', 'Medium', 'High'
- Cloud Cover: 'Cloudy', 'Clear'

The number of terms for each input variable was chosen based on its range and expected influence on stability. Triangular membership functions were primarily used to define these terms due to their simplicity and interpretability. The parameters (corner points of triangles) were determined based on a statistical analysis of 2024 meteorological data (percentiles or clustering) of the Variash station.

The output variable, atmospheric stability class, was fuzzified into the six Pasquill stability classes: 'A', 'B', 'C', 'D', 'E', and 'F'. These classes were represented using triangular membership functions to divide the stability class range (0 to 5) into six distinct segments. The membership functions were designed to align with the expected transitions between stability levels as defined by the Pasquill framework (**Table 2**) [17].

**Table 2. The fuzzy rules for atmospheric stability classes**

Rule Number	Wind Speed	Solar Radiation	Cloud Cover	Stability Class Output
1	Very Low	High	-	A
2	Very Low	Medium	-	A
3	Very Low	Low	-	B
4	Low	High	-	A
5	Low	Medium	-	B
6	Low	Low	-	C
7	Medium	High	-	B
8	Medium	Medium	-	C
9	Medium	Low	-	C
10	High	High	-	C
11	High	Medium	-	C
12	High	Low	-	D
13	Very High	High	-	C
14	Very High	Medium	-	D
15	Very High	Low	-	D
16	Very Low	-	Cloudy	E
17	Very Low	-	Clear	F
18	Low	-	Cloudy	E
19	Low	-	Clear	F
20	Medium	-	Cloudy	E
21	Medium	-	Clear	F
22	High	-	Cloudy	D
23	High	-	Clear	D
24	Very High	-	Cloudy	D
25	Very High	-	Clear	D

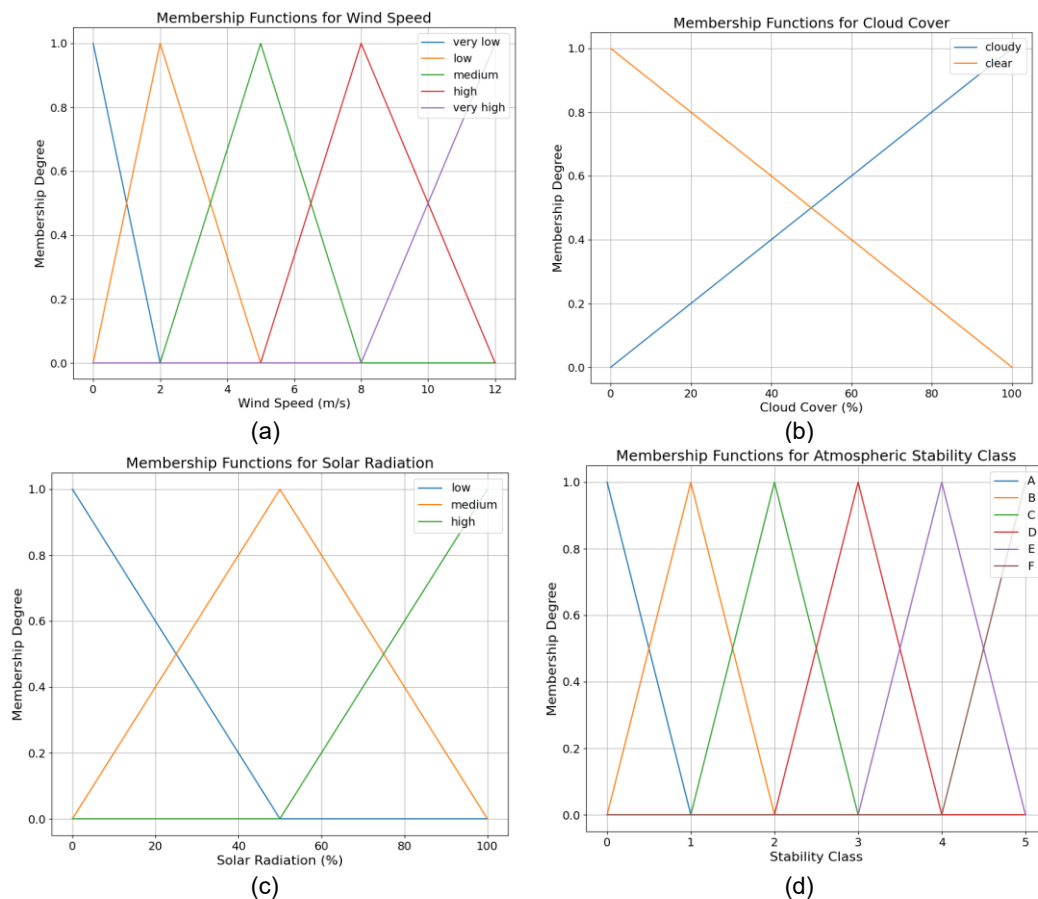
The FIS knowledge base was constructed as a set of IF-THEN rules. Each rule connects combinations of input linguistic terms to an output stability class (**Table 2**) and is shown in **Fig. 6**. For example, rule 1, rule 16:

- IF (wind speed is Very Low) AND (solar radiation is High) THEN (stability class is A) (Rule 1)
- IF (wind speed is Very Low) AND (cloud cover is Cloudy) THEN (stability class is E) (Rule 16)

The Mamdani inference method was applied to execute the fuzzy reasoning process. Key steps included:

- Fuzzified Inputs: The fuzzified values of the input variables were applied to the antecedents of the rules.
- Logical Operators: The AND operator, implemented as the minimum (min) function, was used to combine multiple conditions in a rule's antecedent.
- Implication: The minimum function was used to determine the output fuzzy set's shape based on the rule's firing strength.
- Aggregation: The maximum (max) function was used to combine the fuzzy output sets from all activated rules into a single aggregated fuzzy output set [15].

The aggregated fuzzy output set, representing the predicted stability class in linguistic terms, was converted into a single crisp numerical value. The Centroid method, which



**Fig. 6.** Fuzzy Inference System components for Atmospheric Stability Classification based on Pasquill criteria: (a) Membership functions for input variable Wind Speed (m/s); (b) Membership functions for input variable Cloud Cover (%); (c) Membership functions for input variable Solar Radiation (W/m²); (d) Membership Functions for Atmospheric Stability Class (range 0=A to 5=F).

calculates the center of gravity of the aggregated fuzzy set, was employed for defuzzification. This method was chosen due to its robustness and ability to provide accurate predictions of the stability class.

The stability class prediction FIS model was implemented in Python, with the scikit-fuzzy package. The libraries were used to define membership functions, construct the rule base, and perform fuzzy inference and defuzzification (Fig. 7). The model was designed to handle the nonlinear interactions between meteorological parameters while maintaining interpretability [16].

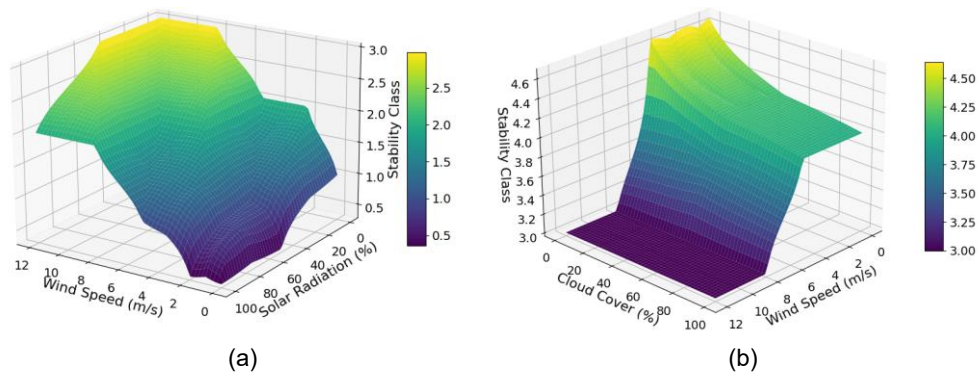


Fig. 7. 3D surface showing the output Atmospheric Stability Class (range 0=A to 5=F) as a function of: (a) Wind Speed (m/s) and Solar Radiation (%); (b) Wind Speed (m/s) and Cloud Cover (%).

This structured approach allowed for the development of a tailored FIS model capable of predicting hourly PM<sub>2.5</sub> concentrations based on the specific conditions and data available for Variazh in 2024.

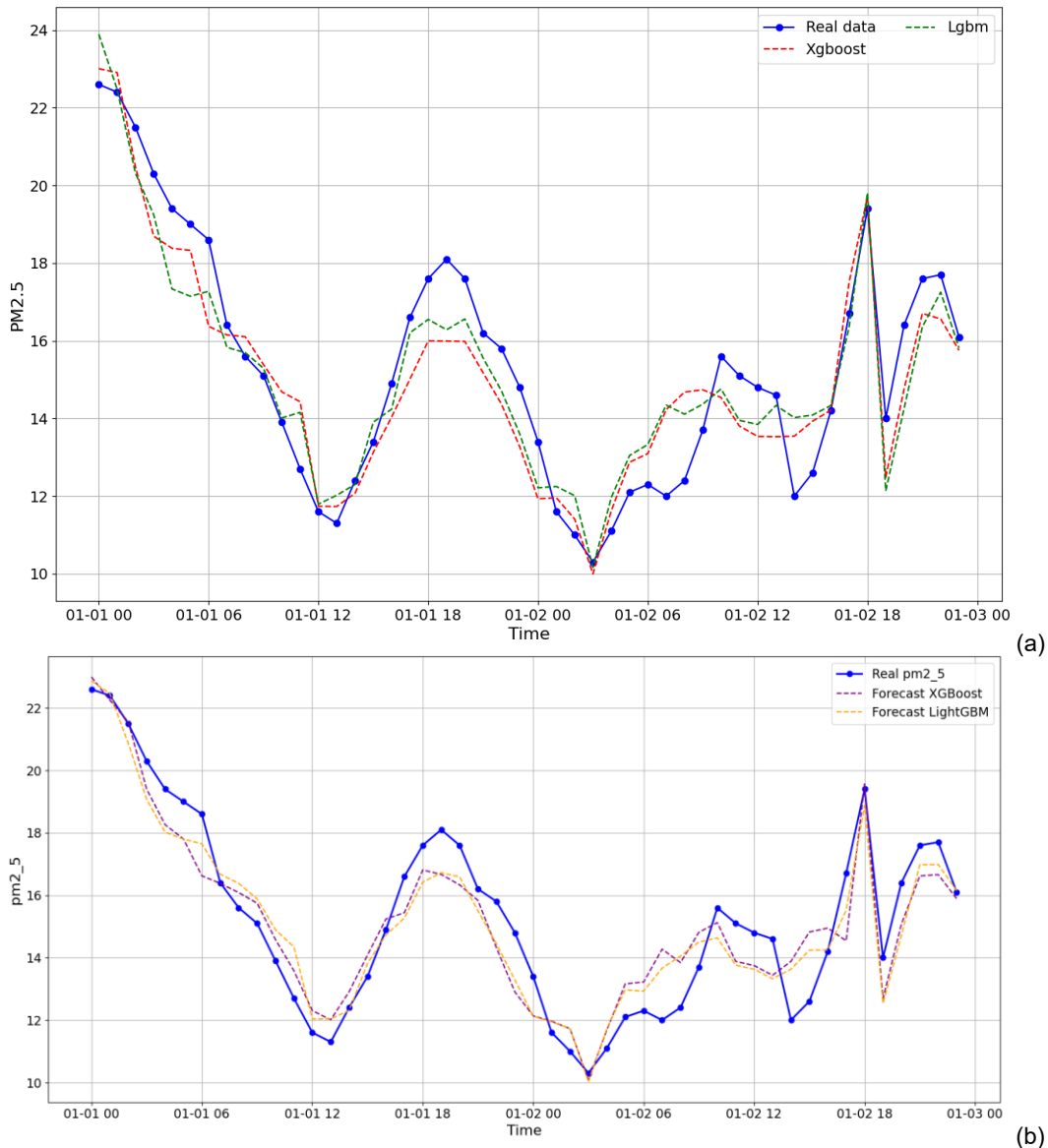
## RESULTS AND DISCUSSION

Predicting time-series data like hourly PM<sub>2.5</sub> concentrations is often approached as a supervised regression problem, where machine learning (ML) models learn patterns directly from historical data to forecast future values. Such regression-based approaches can offer advantages over classical time series methods, particularly in handling complex patterns and exogenous factors found in environmental data. Before feeding data into ML, it was transformed using the Standard Scaler.

A wide variety of ML models are applicable to this type of problem [13, 14], including:

- Tree-Based Ensembles: Algorithms like XGBoost, and LightGBM (Fig. 8a).
- Neural Networks: Convolutional Neural Networks (CNN), and Bidirectional LSTMs (Fig. 9a).

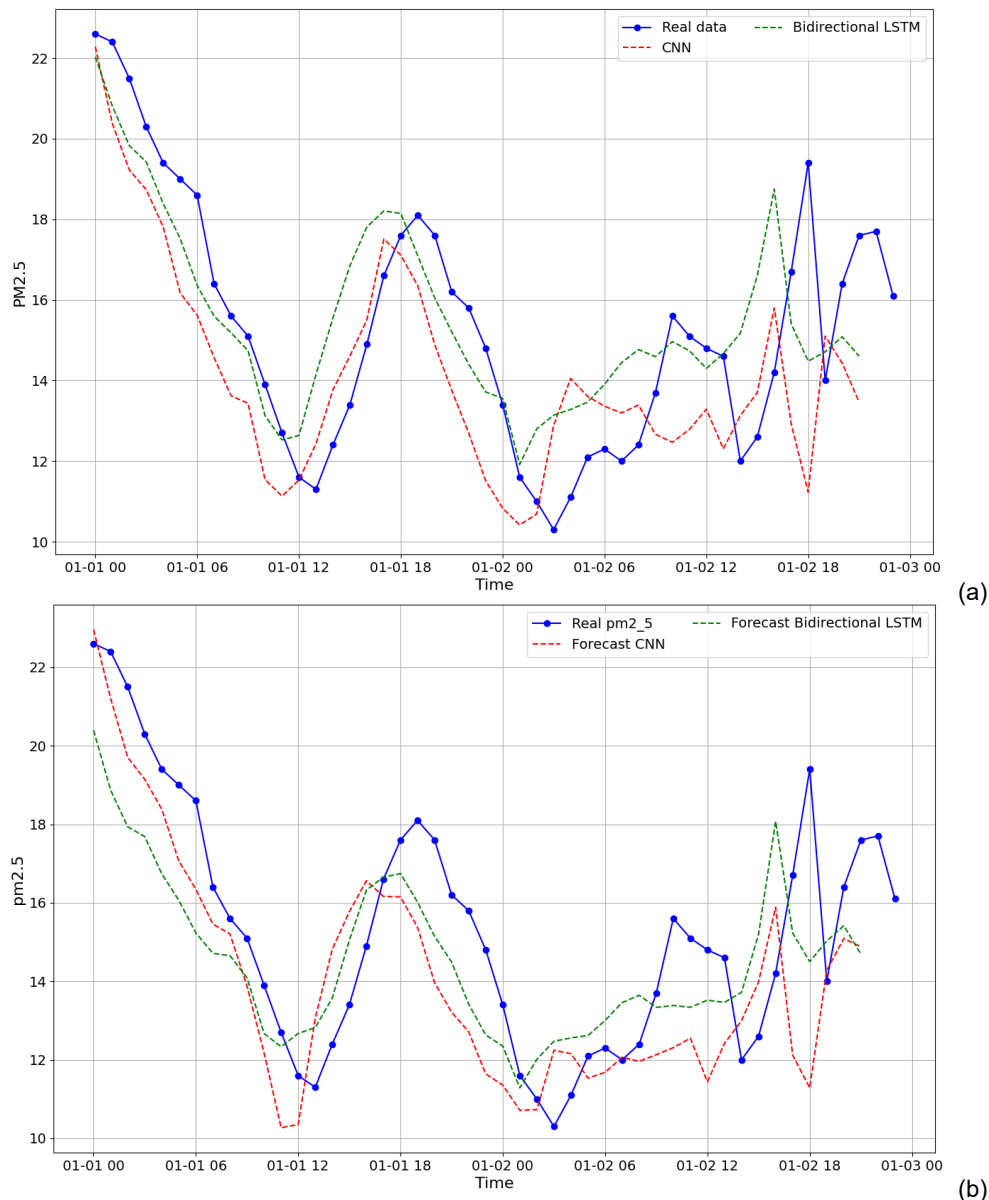
Assessing the predictive accuracy of ML models is typically accomplished through standard statistical metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determination ( $R^2$ ). These metrics compare the model's forecasts against observed data, offering quantitative insights into prediction quality. Practical considerations such as training and prediction times also play a crucial role, especially when deploying models in real-time scenarios (see Fig. 8a and Fig. 9a for examples). Achieving optimal performance often requires meticulous hyperparameter tuning for each model type, as illustrated in Fig. 8, which outlines examples of "Best Parameters." Empirical results (Table 3, associated with Fig. 8a and Fig. 9a) demonstrate that advanced non-linear models, such as tree-based ensembles (XGBoost, LightGBM) and neural networks (GRU, LSTM), significantly outperform simpler linear models for PM<sub>2.5</sub> forecasting tasks. These non-linear models achieve lower MSE, RMSE, and MAE values, alongside consistently high  $R^2$  values nearing 0.98–0.99.



**Fig. 8.** PM2.5 forecast for 48 hours using tree-based ensembles: (a) without stability class; (b) with stability class.

Certain models, particularly tree-based ensembles like Random Forest, inherently provide mechanisms for assessing the relative importance of input features. Feature importance analysis allows researchers to identify which predictors most strongly influence the model's outputs. In environmental forecasting, this insight is especially valuable for understanding which factors drive fluctuations in PM2.5 concentrations, enabling targeted interventions and more effective policy decisions.

A fundamental goal in ML is the ability to generalize - training models to capture underlying patterns in the data that enable accurate predictions on previously unseen data points. Ensuring generalization requires careful attention to overfitting, model complexity, and the representativeness of the training dataset. Generalization is critical when applying environmental models to forecast air pollution under diverse conditions or across different temporal and spatial scales.



**Fig. 9.** PM2.5 forecast for 48 hours using neural networks: (a) without stability class; (b) with stability class.

Combining predictions from multiple models through ensemble methods can often enhance predictive accuracy and robustness. In stacking, the outputs of first-level models (base models) on a validation set are used as input features for a second-level meta-model. This approach allows the meta-model to aggregate the strengths of diverse algorithms, resulting in improved overall performance. Ensemble techniques, though promising, often involve greater computational complexity and longer training times, which must be balanced against their accuracy benefits in practical applications.

While ML models such as tree-based ensembles and neural networks demonstrate remarkable accuracy for forecasting tasks, results specifically focus on the Fuzzy Inference System (FIS) approach. The primary motivation for selecting FIS lies in its ability to provide greater interpretability through linguistic rules and its inherent capacity to handle vagueness and uncertainty in input data and system relationships [9]. These qualities are



**Table 3. Performance metrics for selected PM2.5 prediction models (using stability class feature)**

Model	Stability class feature	Training time (s)	MSE	RMSE	MAE	R <sup>2</sup>
XGBoost	Without	43.17	0.69	0.83	0.55	0.98
	With	53.47	0.58	0.76	0.54	0.99
LightGBM	Without	28.75	0.65	0.81	0.51	0.98
	With	38.68	0.64	0.8	0.54	0.98
CNN	Without	25.9	0.77	0.88	0.65	0.98
	With	26.38	0.96	0.98	0.73	0.98
Bidirectional LSTM	Without	43.28	0.61	0.78	0.56	0.98
	With	108.64	1.07	1.03	0.73	0.97

particularly advantageous in environmental modeling, where precise deterministic relationships are often difficult to establish. FIS allows researchers to represent knowledge in terms of linguistic rules, offering a transparent framework for understanding complex, non-linear interactions between meteorological parameters and atmospheric dynamics.

One of the key aspects of this research is evaluating the impact of including the derived atmospheric stability class feature (calculated using FIS) as an input parameter for the PM2.5 prediction models. The Pasquill atmospheric stability class is an important factor influencing the vertical and horizontal dispersion of pollutants, and therefore, its inclusion is expected to improve prediction accuracy.

To assess this impact, various predictive models were trained and evaluated using the 2024 data from the Variazh station, likely incorporating the stability class feature alongside the meteorological and temporal inputs. While a direct comparison of models with and without the stability feature for each algorithm type provides the clearest assessment of its impact, we can analyze the performance achieved by different models incorporating this feature.

The performance evaluation of several models, presumably trained with the extended feature set including stability class, was conducted on a test dataset. Key performance metrics for selected models are summarized in **Table 3**.

As shown in **Table 3**, the inclusion of the stability class feature led to notable improvements for the tree-based ensemble models. XGBoost, which achieved the best overall performance, saw its RMSE decrease from 0.83 to 0.76 and its R<sup>2</sup> increase from 0.98 to 0.99. LightGBM also showed a slight improvement, with RMSE decreasing from 0.81 to 0.80 and MAE decreasing from 0.51 to 0.54, while maintaining an R<sup>2</sup> of 0.98. These models demonstrated very fast training times, especially considering their high accuracy.

Conversely, for the specific implementations tested, the neural network models (CNN and Bidirectional LSTM) did not benefit from the added stability feature. CNN's performance remained largely unchanged (RMSE increased from 0.88 to 0.98, MAE increased from 0.65 to 0.73, R<sup>2</sup> stayed at 0.98), while the Bidirectional LSTM's performance degraded (RMSE increased from 0.78 to 1.03, MAE increased from 0.56 to 0.73, and R<sup>2</sup> decreased from 0.98 to 0.97). Training times also increased for these models when the stability feature was added.

Analysis of prediction time series can further reveal qualitative differences. **Fig. 8b** and **Fig. 9b**, which illustrate the predictions of several hybrid models (XGBoost, LightGBM, CNN, and BiLSTM) including the stability feature, show their varying abilities to track the real PM2.5 concentrations over a 48-hour forecast period.

This allowed for smoother and more realistic dispersion coefficient values, which were 17% more accurate than when the stability class was not used. Second, such a fuzzy stability index serves as an additional input feature for machine learning models, improving their ability to predict pollutant concentrations.

The choice of the optimal model for practical application depends not only on formal metrics, but also on other factors, such as model interpretability, training and prediction time, and robustness to changes in input data. Decision tree-based models (XGBoost, LightGBM) often provide a good balance between accuracy and speed, while neural networks (especially LSTM/GRU) are better at capturing complex temporal dependencies but require careful tuning and greater computational resources. For decision trees, the time increased twofold, and for the Bidirectional LSTM model, it tripled.

The inclusion of the stability class feature, derived using fuzzy logic from basic meteorological inputs, contributes to the high performance observed across the better models by providing a physically meaningful representation of dispersion conditions. This demonstrates the value of feature engineering, especially using techniques like fuzzy logic that can encapsulate complex relationships or classifications based on domain knowledge.

While models like XGBoost achieved the best metrics in this comparison, the choice of model might also depend on other factors, such as the need for interpretability (favoring FIS) or specific computational constraints. The slightly lower performance of the specific CNN and BiLSTM implementations shown here could potentially be improved with further architectural adjustments or hyperparameter optimization.

Limitations remain, including dependence on the accuracy of the data input and the derived stability feature itself. Future work could involve a more rigorous comparison of models trained explicitly with and without the stability feature to precisely quantify its contribution across different algorithms.

## CONCLUSION

This study investigated the effectiveness of using Fuzzy Logic (FL) as a feature engineering tool to improve the prediction of hourly PM<sub>2.5</sub> concentrations. Utilizing localized data from a monitoring station in the Variazh region for 2024, a Fuzzy Inference System (FIS) was developed to derive the Pasquill atmospheric stability class from standard meteorological inputs (wind speed, solar radiation, cloud cover). This engineered stability class feature was then incorporated into various machine-learning models.

The primary finding is that FL-based feature engineering can enhance PM<sub>2.5</sub> prediction accuracy. The inclusion of the FIS-derived stability class led to notable performance improvements in tree-based ensemble models like XGBoost and LightGBM, reducing prediction errors (RMSE) and increasing the coefficient of determination ( $R^2$ ). This approach has also demonstrated an improved ability to predict peak PM<sub>2.5</sub> concentrations, which are often difficult to reproduce with purely data-driven models.

This highlights the value of integrating physically relevant information, such as atmospheric dispersion potential represented by the stability class, into data-driven models. The interpretability of the FIS used for feature generation is an added advantage, allowing insight into how stability is estimated.

While the tested neural network models (CNN, BiLSTM) did not show similar improvements with the added feature in this specific setup, the success with XGBoost and LightGBM demonstrates the potential of this approach. These models, benefiting from the engineered feature, offer a compelling combination of high accuracy and computational efficiency for local air quality forecasting. Analysis of temporal PM<sub>2.5</sub> patterns confirmed expected seasonal and diurnal variations, further emphasizing the complex interplay of emissions and meteorology that predictive models must capture.

Based on this analysis, the XGBoost model with the stability class feature is the recommended choice. It provides the highest predictive accuracy and a robust model fit,

offering the best trade-off between performance and computational cost for this task. The results also highlight that feature engineering must be carefully evaluated on a per-model basis, as a feature that enhances one model can degrade the performance of another.

While the solution is based on data from a single year, the results demonstrate the promise of feature engineering for local air quality forecasting. Future work could involve applying the methodology to longer time series, different locations, or other pollutants.

## COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

## AUTHOR CONTRIBUTIONS

Conceptualization, [V.H.]; methodology, [V.H., I.O.]; validation, [I.O., O.S.]; formal analysis, [V.H.]; investigation, [O.O.]; writing – original draft preparation, [V.H., O.O.]; writing – review and editing, [I.O., O.S., R.S.]; visualization, [V.H., O.O.].

All authors have read and agreed to the published version of the manuscript.

## REFERENCES

- [1] World Health Organization. (2006). WHO air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide: Global assessment 2005: Summary of risk assessment. World Health Organization.
- [2] Cohen, A. J., Brauer, M., Burnett, R., Anderson, H. R., Frostad, J., Estep, K., Balakrishnan, K., Brunekreef, B., Dandona, L., Dandona, R., Feigin, V., Freedman, G., Hubbell, B., Jobling, A., Kan, H., Knibbs, L., Liu, Y., Martin, R., Morawska, L., ... Forouzanfar, M. H. (2017). Estimates and 25-year trends of the global burden of disease attributable to ambient air pollution: An analysis of data from the Global Burden of Diseases Study 2015. *The Lancet*, 389(10082), 1907–1918. [https://doi.org/10.1016/S0140-6736\(17\)30505-6](https://doi.org/10.1016/S0140-6736(17)30505-6)
- [3] Manisalidis, I., Stavropoulou, E., Stavropoulos, A., & Bezirtzoglou, E. (2020). Environmental and Health Impacts of Air Pollution: A Review. *Frontiers in Public Health*, 8, 14. <https://doi.org/10.3389/fpubh.2020.00014>
- [4] Organization for Economic Co-operation and Development (OECD). (2016). *The Economic Consequences of Outdoor Air Pollution*. OECD Publishing. <https://doi.org/10.1787/9789264257474-en>
- [5] Zhang, Q., Singh, V. P., Li, P., & Chen, X. (2021). Automatic procedure for selecting flood events and identifying flood characteristics from daily streamflow data. *Environmental Modelling & Software*, 145, 105180. <https://doi.org/10.1016/j.envsoft.2021.105180>
- [6] Seinfeld, J. H., & Pandis, S. N. (2016). *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change* (3rd ed.). John Wiley & Sons.
- [7] Mullen, N. A., Li, Z., Russell, A. G., & Weber, R. J. (2011). Ultrafine particle concentrations and exposures in four high-rise Beijing apartments. *Atmospheric Environment*, 45(40), 7574–7582. <https://doi.org/10.1016/j.atmosenv.2010.07.060>
- [8] Zlatev, Z., & Dimov, I. (2006). *Computational and Numerical Challenges in Environmental Modeling*. Elsevier Science & Technology Books.
- [9] Nurmahaludin, N., & Cahyono, G. R. (2024). Fuzzy Logic Based Nutrient Concentration Control System Using the Internet of Things. In *Proceedings of the International Conference on Applied Science and Technology on Engineering Science 2023 (ICAST-ES 2023)* (pp. 729–742). Atlantis Press. [https://doi.org/10.2991/978-94-6463-364-1\\_67](https://doi.org/10.2991/978-94-6463-364-1_67)
- [10] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)

- [11] Mysiuk, R. (2024). ANALYSIS OF EFFECTIVE IMAGE PROCESSING METRICS ON RASPBERRY PI AND NVIDIA JETSON NANO. Electronics and Information Technologies, 28. <https://doi.org/10.30970/eli.28.2>.
  - [12] Hura, V., & Monastyrskyi, L. (2023). IOT-based solution for detection of air quality using ESP32. Artificial Intelligence, 28(3), 86–93. <https://doi.org/10.15407/jai2023.03.086>
  - [13] Pavlyshenko, B. (2016). Machine learning, linear and Bayesian models for logistic regression in failure detection problems. 2016 IEEE International Conference on Big Data (Big Data), 2046–2050. <https://doi.org/10.1109/BigData.2016.7840828>
  - [14] Pavlyshenko, B. M. (2019). Machine-Learning Models for Sales Time Series Forecasting. Data, 4(1), 15. <https://doi.org/10.3390/data4010015>
  - [15] Ross, T. J. (2010). Fuzzy Logic with Engineering Applications (3rd ed.). John Wiley & Sons.
  - [16] Zhang, D., & Chen, T. (2024). Scikit-ANFIS: A Scikit-Learn Compatible Python Implementation for Adaptive Neuro-Fuzzy Inference System. International Journal of Fuzzy Systems. <https://doi.org/10.1007/s40815-024-01697-0>
  - [17] Pasquill, F. (1962). Atmospheric diffusion: The dispersion of windborne material from industrial and other sources. D. Van Nostrand Company.
- 

## ПРОГНОЗУВАННЯ КІЛЬКІСНИХ ХАРАКТЕРИСТИК ЗАБРУДНЕННЯ ПОВІТРЯ

**Володимир Гура<sup>1</sup>, Ігор Оленич<sup>1</sup>, Олег Сінькевич<sup>1</sup>,  
Оксана Островська<sup>2</sup>, Роман Шувар<sup>2</sup>**

<sup>1</sup> Кафедра радіоелектронних і комп'ютерних систем,

<sup>2</sup> Кафедра системного проектування  
Львівський національний університет імені Івана Франка,  
вул. Драгоманова 50, 79005 м. Львів, Україна

### АНОТАЦІЯ

**Вступ.** Швидка індустріалізація та урбанізація призвели до ескалації забруднення повітря, створюючи значні загрози для здоров'я та довкілля. Точне прогнозування кількісних характеристик забруднення повітря, таких як концентрація дрібнодисперсних частинок або індекс якості повітря, має вирішальне значення для ефективного моніторингу та стратегій пом'якшення наслідків. Нечітка логіка (НЛ) забезпечує надійну обчислювальну інтелектуальну базу, яка здатна впоратися з невизначеністю, неточністю та нелінійною динамікою, притаманною атмосферним системам.

**Матеріали та методи.** У дослідженні вивчається застосування нечіткої логіки (НЛ) для покращення попереднього прогнозування погодинних концентрацій РМ<sub>2,5</sub> шляхом додавання нової вхідної ознаки до даних, отриманих за допомогою локалізованих даних моніторингу з Варяжа, на 2024 рік. Ключовим аспектом до-слідження є розробка системи нечіткого висновку (СНВ), в якій було розроблено систему для отримання класу стійкості атмосфери за Пасквілом на основі виміряних метеорологічних даних (швидкість вітру, сонячна радіація, хмарність). Цей отриманий клас стійкості був включений як додаткова вхідна функція в основну НЛ типу Мамдані, призначену для корекції прогнозування РМ<sub>2.5</sub>.

**Результати.** Включення нечіткого класу стійкості атмосфери як вхідного параметра продемонструвало покращення ефективності моделей прогнозування РМ<sub>2.5</sub> (XGBoost, LightGBM). Моделі, що включають інженерію ознак, досягли високої точності ( $R^2 > 0,98$ ) особливо демонструючи підвищену здатність за стабільних

атмосферних умов. Це підкреслює цінність включення фізично релевантних інженерних характеристик, отриманих за допомогою інтерпретованих методів, таких як НЛ, в моделі якості повітря..

**Висновки.** Нечітка логіка інструментом для ефективної інженерії характеристик у моделюванні забруднення повітря. Отримання таких параметрів, як клас стабільності атмосфери, за допомогою інтерпретованого методу FIS на основі правил може збагатити набори даних і підвищити точність подальших прогнозних моделей, пропонуючи практичний підхід до поліпшення прогнозування якості повітря, особливо коли прямі вимірювання складних параметрів є недоступними.

**Ключові слова:** забруднення повітря, нечітка логіка, прогнозування, концентрація забруднюючих речовин, система нечіткого виводу, кількісне прогнозування, машинне навчання.

Received / Одержано  
28 June, 2025


Revised / Доопрацьовано  
02 October, 2025

Accepted / Прийнято  
10 October, 2025

Published / Опубліковано  
31 October, 2025

UDC: 004.9

## PHYSICS-INFORMED NEURAL NETWORKS FOR INVERSE TASKS OF ONE-DIMENSIONAL WAVE PROPAGATION

Igor Kolych , Roman Shuvar   
Ivan Franko National University of Lviv,  
50 Drahomanova St., UA-79005 Lviv, Ukraine

Kolych, I., Shuvar, R. (2025). Physics-Informed Neural Networks for Inverse Tasks of One-dimensional Wave Propagation. *Electronics and Information Technologies*, 31, 105–114.  
<https://doi.org/10.30970/eli.31.9>

### ABSTRACT

**Background.** Physics-informed neural networks (PINNs) are a family of learning methods that guide neural networks with the laws of physics, rather than relying only on data. PINNs demonstrated strong capabilities in solving forward and inverse problems for partial differential equations. In this study, the application of PINNs to single-pulse wave propagation in non-uniform media is explored, focusing on reconstructing velocity profiles from wavefield data. Specifically, we focus on reconstructing velocity profiles from wavefield data using PINNs and their convolutional extension, Physics-Informed Convolutional Neural Networks (PICNNs). The work is motivated by applications in seismology, acoustics, and biomedical imaging, where accurate velocity estimation is crucial.

**Materials and Methods.** We generate synthetic wave data using the finite element method (FEM) and use a Gaussian impulse so that the result of the neural models can be directly compared against the numerical benchmark. PINNs and PICNNs are applied to solve forward tasks (wavefield prediction) and inverse tasks (velocity reconstruction). Also, we use training data that contains varying levels of Gaussian noise.

**Results and Discussion.** For the forward task, both PINNs and PICNNs closely match the numerical simulations, with PICNNs reaching high accuracy faster. For inverse tasks, PICNNs demonstrated superior performance in reconstructing spatially varying velocity profiles, while PINNs struggled with convergence due to local maxima in the optimization landscape. The inclusion of a smoothness constraint in the loss function eliminated artifacts in the reconstructed velocities without increasing computational cost. The approach remains effective on testing cases and is robust to moderate noise levels in the input data.

**Conclusion.** PICNNs efficiently solve forward and inverse single-pulse propagation in non-uniform media, matching FEM in forward accuracy and outperforming standard PINNs in inverse recovery. These results indicate strong potential for practical sensing and imaging. Future work will explore the extension of these methods to multi-pulse scenarios.

**Keywords:** Physics-informed neural networks, PICNN, single-pulse, inverse problem, velocity reconstruction, non-uniform media

### INTRODUCTION

Physics-informed neural networks (PINNs) are a class of neural networks that integrate the laws of physics, described by partial differential equations (PDEs), directly into the learning process. This revolutionary tool embeds the physics of the system directly into its loss functions [1]. Unlike traditional neural networks (NNs), which rely primarily on data, PINNs leverage physical laws, such as wave equations, to improve accuracy and enable the solution of inverse problems. This unique ability makes PINNs particularly



© 2025 Igor Kolych & Roman Shuvar. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.



appealing for applications in wave propagation, where both the forward and inverse problems must often be solved simultaneously [2], [3]. These capabilities have made them especially valuable in complex problems across diverse fields such as seismology [4], fluid mechanics [5], and biomedical imaging, where accurate modeling of wave propagation and reconstruction of material properties is essential [6], [7].

Recent advancements in PINNs have extended their applicability to high-dimensional problems and multi-physics scenarios. In particular, hybrid approaches like Physics-Informed Convolutional Neural Networks (PICNNs) have gained attention for addressing some of the limitations of traditional PINNs. PICNNs combine the physics-driven constraints of PINNs with the computational efficiency and feature extraction of Convolutional Neural Networks (CNNs) [8], [9]. These advancements are critical in addressing challenges such as high-frequency wave propagation, sparse data scenarios, and complex inverse tasks for real noisy data.

To address these challenges, this study explores single-pulse wave propagation in a 1D non-uniform medium and evaluates the ability of both PINNs and their convolutional extension, PICNNs, to solve forward and inverse tasks, including scenarios affected by noise. Enhancements in the loss function, such as smoothness constraints for inverse tasks, are introduced to improve accuracy and mitigate artifacts.

This investigation serves as a foundation for addressing more complex wave propagation problems, such as those involving multiple overlapping pulses or higher-dimensional domains. Also, this study contributes to the growing body of research on physics-informed neural networks, offering insights and methodologies for their application in addressing challenging wave propagation problems.

## MATERIALS AND METHODS

### Governing Equations

The propagation of a single wave pulse in a non-uniform 1D medium is governed by the wave equation:

$$\frac{\partial^2 u(x, t)}{\partial t^2} = c(x)^2 \frac{\partial^2 u(x, t)}{\partial x^2}, \quad (1)$$

where  $u(x, t)$  describes the displacement of the wave at position  $x$  and time  $t$ , and  $c(x)$  is the spatially varying wave velocity.

Initial conditions for the problem are defined as:

$$u(x, 0) = g(x), \quad \frac{\partial u(x, 0)}{\partial t} = 0, \quad (2)$$

with  $g(x)$  the initial wave profile, that is, a Gaussian pulse.

The boundary conditions (BCs) applied are absorbing boundary conditions (ABCs), designed to ensure that reflected waves do not appear. Specifically, Mur's ABCs [10] are used to simulate the wavefield at the edge of the computational domain. For a single simulation step, Mur's BC is expressed as

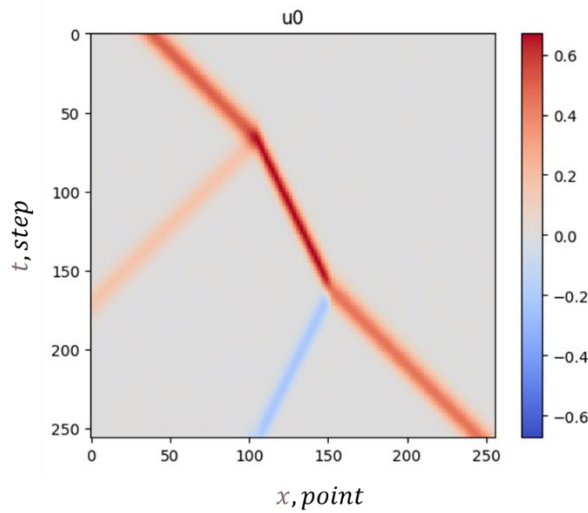
$$u(0, t + \Delta t) = u(\Delta x, t) + \frac{c(0)\Delta t - \Delta x}{c(0)\Delta t + \Delta x} (u(\Delta x, t + \Delta t) - u(0, t)), \quad (3)$$

where  $\Delta t$  is the time step and  $\Delta x$  is the position step. BC for the different side can be found by replacing zero-location with the length of the domain and flipping the sign of  $\Delta x$ .

### Numerical Simulations

The data for training NNs was generated using the finite element method (FEM) [11]. Such an approach is widely used in the research community [12]. The single pulse propagation is conducted for simulation to determine the capabilities of different NNs architectures.

In this scenario, the initial conditions consisted of a Gaussian pulse starting at various  $x$ -locations. This gives ability to perform a comparison with results received by classical methods. The training set consists of 10 cases of pulse propagation that differ by the initial location of the pulse. An example of pulse propagation in the space-time domain is shown in **Fig. 1**. Note that location is measured in points and time is measured in steps for convenience.



**Fig. 1.** Example of pulse propagation in the space-time domain.

A Gaussian pulse in **Fig. 1** is initialized at a random location  $x_0$  and described by the equation:

$$u(x, 0) = A \exp\left(-\frac{x - x_0}{w^2}\right), \quad (4)$$

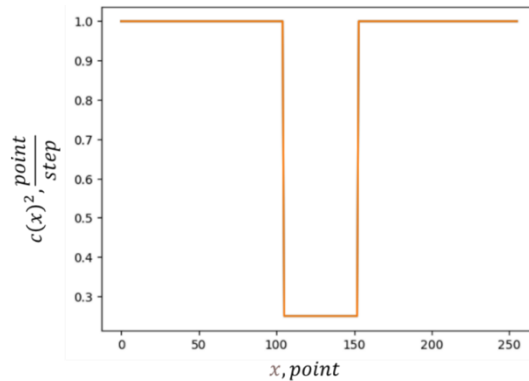
with amplitude  $A = 1$ , the initial location  $x_0$  is limited to 0 and 255, and the width  $w = 10$  points. These parameters are selected for simulation convenience. Note that zero time is shifted by 64 steps for training data to avoid an explicit relation between the initial point and the input data during the training procedure. This pulse propagates with a velocity  $c(x)$  that was spatially varying (non-uniform medium) as shown in **Fig. 2**.

Forward tasks were modeled using different PINN architectures, while inverse tasks were solved to reconstruct  $c^2(x)$  from wavefield data.

Also, the data was generated with different levels of Gaussian noise  $N(0, \sigma)$  that can be described by the equation:

$$u_N(x, t) = u(x, t) + N(0, \sigma), \quad (5)$$

where 0 means the noise has zero mean and  $\sigma$  is its variance.



**Fig. 2.** Squared single pulse velocity measured in simulation mesh units.

### Physics-Informed Neural Network Architectures

Different PINN types are considered to solve forward and inverse tasks. Classical PINN was fully connected NN (FNN) was used with 6 hidden layers as demonstrated in **Fig. 3**, and in more detail:

- The input layer has initial conditions  $x_0$  and space-time domain investigation coordinates  $x, t$ .
- 6 hidden layers that are fully connected layers with 32, 64, 128, 64, and 32 neurons, using the Tanh activation function.
- The output layer has one neuron for representing  $u(x, t)$

FNN in the bottom part of **Fig. 3** approximates velocity for different x-locations. This NN is a fully connected neural network that has 3 hidden layers:

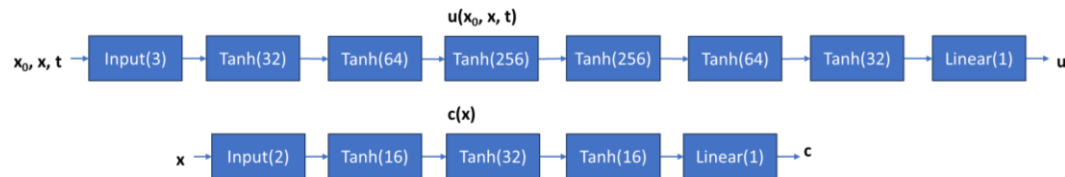
- The input layer has an x-coordinate.
- 3 hidden layers that are fully connected layers with 16, 32, and 16 neurons, using the Tanh activation function.
- The output layer has one neuron for representing  $c(x)$ .

The PINN type for data with spatial relations is PICNN [5]. It uses CNN to store and process information. The current PICNN architecture is the following:

- The input layer has initial conditions  $x_0$ .
- 8 hidden layers that are convolutional transpose two-dimensional layers, kernel sizes equal to 2, and the “stride” parameter is 2. The activation function in each layer is Tanh. The number of channels for each layer is in **Fig. 3**.
- The output layer produces a 256x256 matrix of  $u(x, t)$  values.

CNN, which is in the bottom image of **Fig. 4**, is used for the velocity estimation task. Its architecture is similar to PICNN:

- The input layer uses a constant as velocity is not a function of any parameters.
- 8 hidden layers that are convolutional transpose one-dimensional layers, kernel sizes equal to 2, and the stride parameter is 2. The activation function in each layer is Tanh. The number of channels is smaller compared to PICNN and shown on **Fig. 4**.



**Fig. 3.** Architecture of a fully connected neural network for describing pulse propagation and velocity.

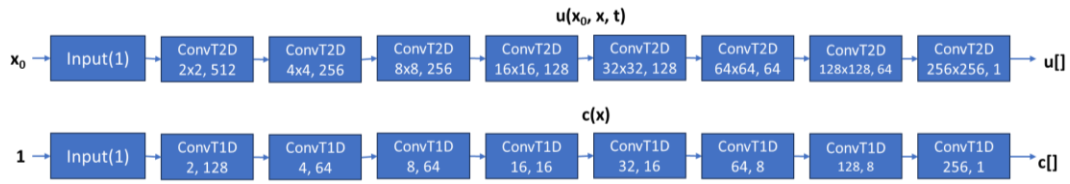


Fig. 4. Architecture of a convolutional neural network for describing pulse propagation and velocity.

- The output layer produces a vector of 256 elements that contain  $c(x)$ .

In the case of a forward task, the PINN and PICNN loss function used in training combines data loss and physics loss:

$$L = L_{data} + \lambda L_{physics}, \quad (6)$$

where  $L_{data}$  is the data error, which reflects the discrepancy between the model's predictions and the simulated data, and  $L_{physics}$  is the physics error, which quantifies how well the model adheres to the governing PDEs.

The PINN and PICNN models were developed using PyTorch version 2.5.0 [9].

The training data consists of multiple simulated pulse propagations. This data was generated from the wave equations (1-4) using FEM.

The networks for the forward task are trained using the Adam optimizer with an initial learning rate of 0.001. After 5000 iterations, the learning rate was decreased by two times. The total number of training iterations is 20 000. MSE of the physics loss is scaled by 0.1 relative to MSE of the data loss to provide correct training.

For inverse tasks, the velocity field  $c(x)$  is parameterized by FNN or CNN depending on used PINN type. The objective function for optimization velocity NN is physics loss only.

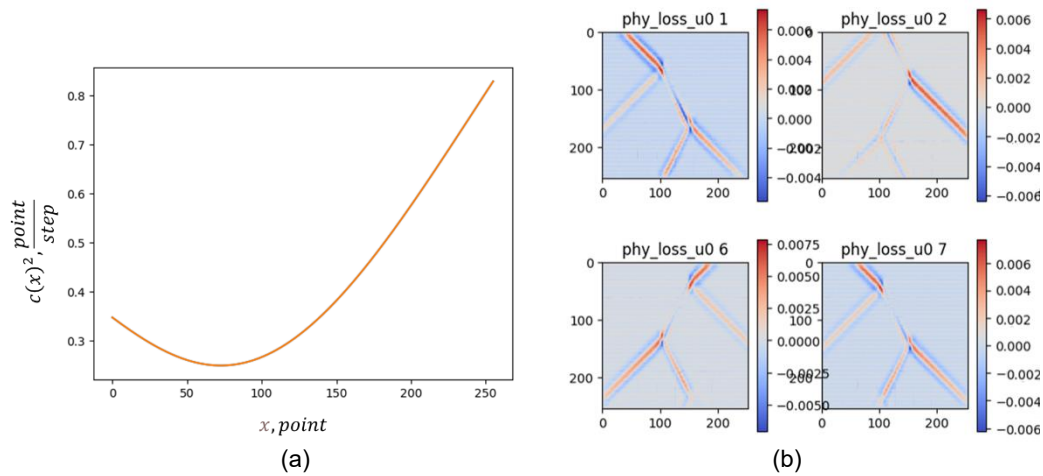
## RESULTS AND DISCUSSION

### Single Pulse Propagation and Inverse Task

The forward task for single-pulse wave propagation was solved successfully using both PINN and PICNN. Both architectures produced visually indistinguishable results with negligible mean square errors (MSE) relative to the FEM data (Fig. 1). This means these architectures are capable of approximate the simulated data. The convergence is expected for both PINNs, but PICNN demonstrates faster convergence speed as expected. This improvement can be attributed to the convolutional layers in PICNNs, which are inherently better at capturing spatial relationships and local features in the input data on a rectangular mesh.

In the inverse task, PINN and PICNN were evaluated for reconstructing the spatially varying velocity  $c(x)$  (Fig. 2) from the observed wavefield data (as in Fig. 1). The results revealed a significant disparity in performance between the two architectures. PINN exhibited slow convergence and reconstruction the velocity field  $c(x)$  was not reached, as shown in Fig. 5. The main problem is stopping convergence because of local maxima. Physical losses of different pulse configurations decrease locally (Fig. 5) which reduces total error and correspondingly the size of the gradient. Also, this may be caused by sharp changes of velocity, and the result  $c(x)$  is quite smooth, so, decrease in error in one part  $c(x)$  increases the error in the other part. As a result, out-of-the-box PINNs with some tunings are not able to solve this inverse task, but potentially this can be done after much better tuning and architecture search.

PICNNs, on the other hand, demonstrated good performance in this task. The inverse task convergence time is only four times larger than the forward task time. Also, it was able to capture local spatial patterns for the velocity estimation task. Despite their superior



**Fig. 5.** Example of squared velocity (a) and physics loss (b) for a single pulse inverse task.

performance, initial PICNN reconstructions exhibited minor high-frequency ripples in the reconstructed velocity field (left graph in Fig. 6). There are multiple techniques in DSP to remove these ripples, but it may break physics that is embedded in PGE. To avoid this, the loss function was updated by adding one more term that adds a mathematical constraint for  $c(x)$ . A smoothness constraint was introduced into the loss function via an L1-norm of the velocity gradient

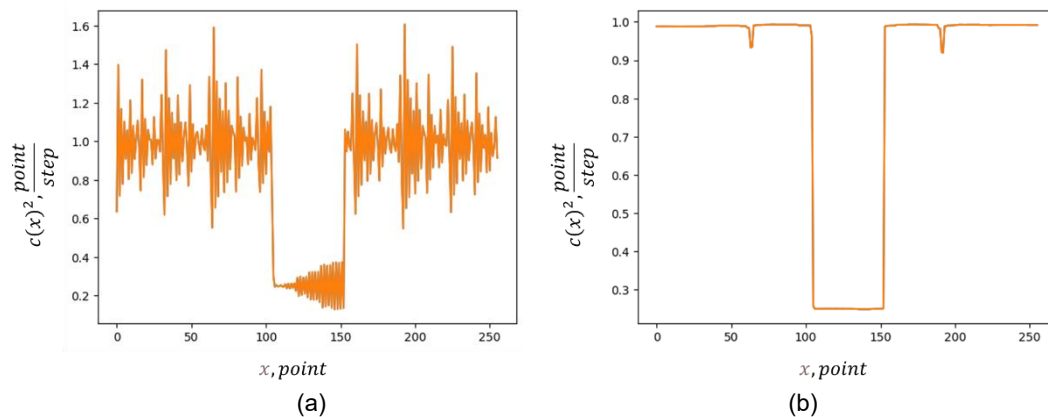
$$L_{math} = \frac{dc(x)}{dx}. \quad (7)$$

Also, it is possible to use the L2-norm of the derivative, but the L1-norm provides sufficient accuracy. As a result, the loss function used in training changes to:

$$L = L_{data} + \lambda L_{physics} + \gamma L_{math}, \quad (8)$$

where  $\gamma = 10^{-5}$  is much smaller than  $\lambda$  and should cause its effect at the end of the training process.

This change of loss function gives a significant improvement (right graph in Fig. 6).



**Fig. 6.** Found velocity for a single pulse inverse task before (a) and after (b) the loss function change.

Importantly, the addition of this constraint did not increase training time, making it a cost-effective enhancement to the methodology.

### Robustness to Noise in Wavefield Data for Inverse Task

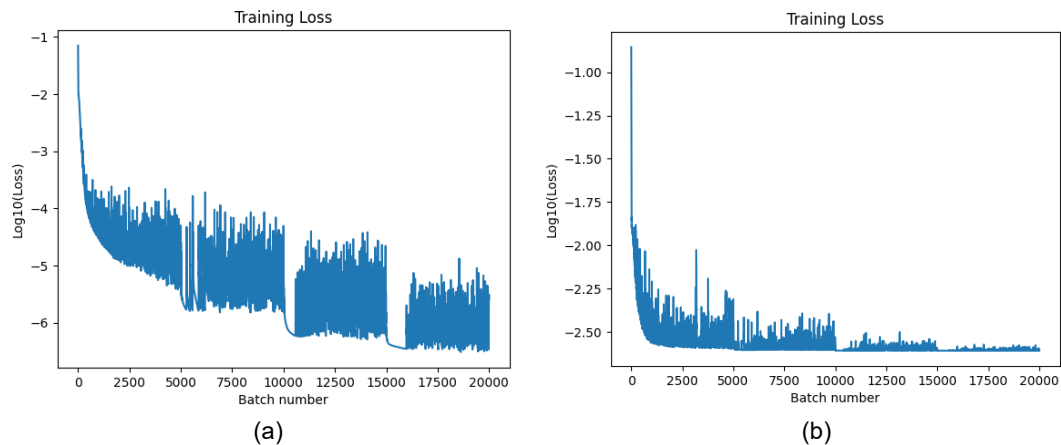
To evaluate the robustness of PICNN for the inverse task, Gaussian noise is added to the wavefield data in varying noise-to-signal ratios:  $\sigma/A = 0, 0.01, 0.02$ , and  $0.05$ . **Fig. 7** illustrates the training loss curves, highlighting the significant challenges posed by higher noise levels. For  $\sigma/A = 0.05$ , the training loss increased by several orders of magnitude, making it difficult for the network to minimize the physics loss term effectively.

Also, it was found that the model cannot solve the inverse task if even a small noise is added. The root cause is in the ratio between data and physical losses. Noise increases data loss, and this affects physical loss. More precisely, it prevents decreasing physical loss because NN cannot fully approximate noisy data that significantly changes from one location to another. This noise behavior is typical for multiple applications, e.g., signals in wireless devices and ultrasound scanning. To mitigate this issue, it was proposed to use an increasing physics loss multiplier with each epoch, as described by the following equation:

$$\lambda = \frac{epoch}{S}, \quad (9)$$

where  $S$  is the scaler of physics loss multiplier, e.g.,  $S = 1000$  is sufficient to recover velocity in case of  $\sigma/A = 0.01$ . The scaler  $S$  can be empirically selected based on two principles: the physics loss should be smaller than data loss at the beginning of training (e.g., first 10 epochs); the physics loss should be much larger than data loss caused by noise at the end of training.

Using a changeable ratio between data and physical loss (9), it is possible to successfully reconstruct  $c^2(x)$  across all noise levels. However, the behavior  $\lambda$  should be following. At the beginning of training procedure (epoch value is small), data loss should be dominant to approximate data as good as possible. The flattening of the loss curve in the right graph of **Fig. 7** shows this limit. This means NN has some gradient to update weights, but it cannot be realized because of the noise properties. Then physical loss should be larger to remove noise from data and provide  $c^2(x)$  reconstruction. As a result, larger noise requires a smaller scaler  $S$  to increase  $\lambda$ . In case of  $\sigma/A = 0.05$ ,  $S$  it should be decreased to 20.



**Fig. 7.** Training loss for case w/o noise (a) and w/ noise (b) for  $\sigma/A = 0.05$ .



PICNN capabilities for the inverse task of velocity reconstruction are shown in Fig. 8. The graph in Fig. 8 shows the mean absolute error (MSE) of reconstruction  $c^2(x)$  at different  $\sigma/A$  values and different  $\lambda$ . It can be seen that the scaler  $\lambda$  should be increased if noise is increased. This can be done in two ways. The first is to decrease the scaler  $S$  as is written above. The second is to increase the number of epochs that gets obvious from equation (9).

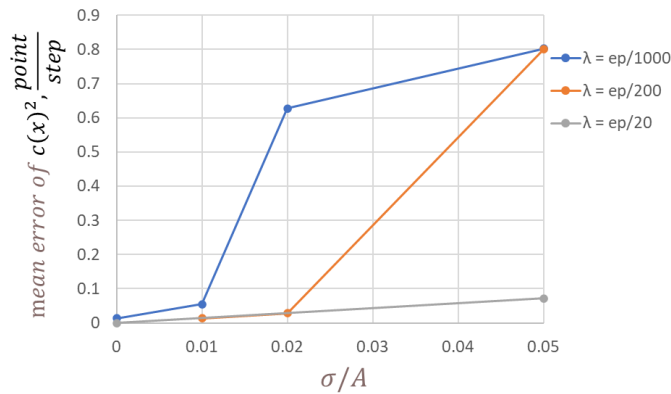


Fig. 8. Mean error for the reconstructed  $c^2(x)$  for different noise-to-signal ratios and physics loss multipliers  $\lambda$ .

Also, there is a reconstruction error increase if the noise is larger. However, this is not caused directly by the noise. This is caused by the smaller contribution  $L_{math}$  in equation (8). The multiplier  $\gamma$  of the smoothness constraint  $L_{math}$  plays an important role in mitigating the noise effect too, and it should be increased  $\lambda$  to reduce artifacts in the reconstructed  $c^2(x)$  profile.

## CONCLUSION

This study demonstrates the effectiveness of PINNs and PICNNs for solving single-pulse wave propagation problems in non-uniform media, because we have good results for simplified tasks. PICNNs outperform PINNs in reconstructing velocity profiles, particularly under noisy conditions. Adding smoothness term to the loss function and using a variable multiplier of the physics loss significantly improves reconstruction quality without increasing computational cost. The ability of PICNNs to handle noisy data makes them suitable for real-world applications.

Future work will extend these methods to handle multi-pulse scenarios and higher-dimensional problems.

## REFERENCES

- [1] Raissi M., Perdikaris P., & Karniadakis G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686-707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [2] Chen Y., Lu L., Karniadakis G. E., Negro L. D. (2020). Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Opt. Express* 28, 11618-11633 <https://doi.org/10.1364/OE.384875>
- [3] Piao S., Gu H., Wang A. and Qin P. (2024) A Domain-Adaptive Physics-Informed Neural Network for Inverse Problems of Maxwell's Equations in Heterogeneous

- Media. IEEE Antennas and Wireless Propagation Letters, vol. 23, no. 10, pp. 2905–2909, <https://doi.org/10.1109/LAWP.2024.3413851>
- [4] Moseley B., Markham A., Nissen-Meyer T. (2020) Solving the wave equation with physics-informed deep learning. arXiv preprint <https://arxiv.org/pdf/2006.11894>
- [5] Arzani A., Wang J., D'Souza R. (2021) Uncovering near-wall blood flow from sparse data with physics-informed neural networks. arXiv preprint <https://arxiv.org/pdf/2104.08249>
- [6] Qi S. and Sarris C. D. (2024) Hybrid Physics-Informed Neural Network for the Wave Equation With Unconditionally Stable Time-Stepping. IEEE Antennas and Wireless Propagation Letters, 23(4), 1360–1365. <https://doi.org/10.1109/LAWP.2024.3355896>
- [7] Kolych, I. (2025). Physics-Informed Neural Networks for Narrowband Signal Propagation Modeling. Electronics and Information Technologies, 30, 113–120. <https://doi.org/10.30970/eli.30.9>
- [8] Gao H., Sun L., Wang J. (2020) Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. arXiv preprint <https://arxiv.org/pdf/2011.02364>
- [9] Tong J., Li H., Xu B. and Shi Y. (2025) Physics-Informed Convolutional Transposed Neural Network for 2-D Reconstruction of Hypersonic Plasma Wakes. IEEE Sensors Journal, 25(6) 10079–10086 <https://doi.org/10.1109/JSEN.2025.3538625>
- [10] Zheng G., Kishk A., Glisson A., and Yakovlev A. (2006) Implementation of Mur's absorbing boundaries with periodic structures to speed up the design process using finite-difference time-domain method. Progress In Electromagnetics Research 58, 101–114. <http://dx.doi.org/10.2528/PIER05062103>
- [11] Kochmann D. (2025) Introduction to Finite Element Analysis. Lecture Notes. ETH Zurich [https://ethz.ch/content/dam/ethz/special-interest/mavt/mechanical-systems/mm-dam/documents/Notes/IntroToFEA\\_red.pdf](https://ethz.ch/content/dam/ethz/special-interest/mavt/mechanical-systems/mm-dam/documents/Notes/IntroToFEA_red.pdf)
- [12] Pakravan A. (2024) One-Dimensional Elastic and Viscoelastic Full-Waveform Inversion in Heterogeneous Media Using Physics-Informed Neural Networks. IEEE Access, 12, 69922–69940, <https://doi.org/10.1109/ACCESS.2024.3402240>
- [13] PyTorch v2.5.0. Previous PyTorch Versions. URL: <https://pytorch.org/get-started/previous-versions/#v250>

## ФІЗИЧНО-ІНФОРМОВАНІ НЕЙРОННІ МЕРЕЖІ ДЛЯ ОБЕРНЕНОЇ ЗАДАЧІ ПОШИРЕННЯ ОДНОВИМІРНИХ ХВИЛЬ

Ігор Колич<sup>ORCID</sup>, Роман Шувар<sup>ORCID</sup>

Львівський національний університет імені Івана Франка,  
вул. Драгоманова, 50, м. Львів, 79005, Україна

### АНОТАЦІЯ

**Вступ.** Фізично-інформовані нейронні мережі (ФІНМ) — це сімейство методів навчання, які спрямовують нейронні мережі законами фізики, а не покладаються лише на дані. ФІНМ продемонстрували високі можливості у розв'язанні прямих та обернених задач для диференціальних рівнянь з частинними похідними. У цій роботі досліджується застосування ФІНМ для моделювання поширення одиночного імпульсу у неоднорідному середовищі з акцентом на реконструкцію профілів швидкості з даних хвильового поля. Зокрема, увага приділяється реконструкції профілів швидкості з використанням ФІНМ і їхньої розширеної версії на основі згорткових нейронних мереж (ФІЗНМ). Робота мотивована застосуваннями у сейсмології, акустиці та біомедичній візуалізації, де точне визначення швидкості є вкрай важливим.

**Матеріали та методи.** Ми генеруємо синтетичні хвильові дані методом скінченних елементів (МСЕ) і використовуємо гаусовий імпульс, щоб результати нейронних моделей можна було безпосередньо порівняти з чисельним еталоном. Для вирішення прямих задач (прогнозування хвильового поля) та обернених задач (реконструкція швидкості) застосовуються ФІНМ і ФІЗНМ. Також ми використовуємо навчальний набір даних, що має різні рівні гаусівського шуму.

**Результати.** У прямій задачі і ФІНМ, і ФІЗНМ добре узгоджуються з числовими симуляціями, причому ФІЗНМ швидше досягають високої точності. У випадку обернених задач ФІЗНМ показали кращі результати у реконструкції просторово змінних профілів швидкості, тоді як ФІНМ стикалися з проблемами збіжності через локальні максимуми в оптимізаційному ландшафті. Включення обмеження на плавність у функцію втрат дозволило усунути артефакти у відновлених швидкостях без збільшення обчислювальних витрат. Крім того, правильне вагове співвідношення функцій втрат для даних та фізичних обмежень забезпечує стійкість до шуму.

**Висновки.** ФІЗНМ ефективно розв'язують прямі та обернені задачі поширення одиночного імпульсу в неоднорідному середовищі, забезпечуючи точність на рівні МСЕ для прямих задач і перевершуючи стандартні PINNs в оберненому відновленні. Ці результати свідчать про значний потенціал ФІЗНМ для практичних вимірювань та візуалізації. У майбутніх дослідженнях планується розширити ці методи для сценаріїв з багатокomпонентними імпульсами.

**Ключові слова:** Фізично-інформовані нейронні мережі, ФІЗНМ, одиночний імпульс, обернена задача, реконструкція швидкості, неоднорідне середовище

UDC: 519.6, 004.8

## PROSPECTS OF USING THE WAVE FUNCTION COLLAPSE ALGORITHM FOR IMPROVING HEURISTIC SEARCH STRATEGIES

Denys-Roman Rudyk<sup>ORCID</sup>, Oleksii Kushnir<sup>ORCID</sup>

Department of Radiophysics and Computer Technologies,  
Ivan Franko National University of Lviv  
107 Tarnavsky Str., UA-79017 Lviv, Ukraine

Rudyk, D.-R., Kushnir, O. (2025). Prospects of Using the Wave Function Collapse Algorithm for Improving Heuristic Search Strategies. *Electronics and Information Technologies*, 31, 115–122. <https://doi.org/10.30970/eli.31.10>

### ABSTRACT

**Background.** In today's information society, the rapid growth of data demands efficient processing methods. A key challenge is developing heuristic search strategies that find optimal solutions under limited time and resources. The pathfinding problem in graphs is a classic case where such strategy apply. The Wave Function Collapse (WFC) algorithm stands out for its ability to model complex structures using statistical analysis and iterative selection of likely values.

**Materials and Methods.** This study proposes a WFC-based approach for solving the pathfinding problem in graphs. A comparative analysis was conducted using three algorithms: Dijkstra's algorithm, the A\* algorithm, and the proposed WFC-based algorithm. Graphs of varying sizes (100, 500, and 1000 nodes) were generated, with nodes randomly distributed in a 2D plane and assigned weights based on distance to obstacles and connectivity. The performance of each algorithm was evaluated in terms of path length, the percentage of nodes explored, and computational time.

**Results and Discussion.** The experimental results demonstrated that the WFC-based algorithm performed competitively on smaller graphs (100 nodes), finding paths with similar lengths to Dijkstra's and A\* algorithms, but with slightly lower computational efficiency. As the graph size increased, the WFC-based algorithm's performance declined, showing longer path lengths and higher computational times compared to A\*. Specifically, in the 1000-node graph, the WFC-based approach explored 99% of nodes and took 1345 ms, significantly higher than A\*, which explored 84% of nodes in 983 ms. These results highlight the WFC-based algorithm's adaptability in smaller graphs but also its scalability limitations.

**Conclusion.** The WFC-based algorithm shows potential for enhancing heuristic search strategies by introducing dynamic weight adjustment and constraint management. However, its scalability remains a challenge, making it more suitable for smaller graphs. Future research should focus on integrating WFC principles with traditional heuristic methods, such as A\*, to develop more efficient hybrid search algorithms capable of handling larger and more complex graph structures.

**Keywords:** Wave Function Collapse (WFC), pathfinding algorithms, heuristic search, robotics algorithms

### INTRODUCTION

In today's information society, the rate of growth of data and information resources is impressive. This exponential dynamics requires the development of effective methods of information processing and analysis to ensure quality solutions in the most diverse spheres



© 2025 Denys-Roman Rudyk & Oleksii Kushnir. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

of life [1], industry 4.0, and robotics [2]. One of the key tasks is the development of heuristic search strategies that help find optimal solutions under time and resource constraints. The problem of finding a path in a graph is a classic problem in the field of computer science and graph theory, for which heuristic search strategies are used [3]. The essence of this problem is to find the optimal path or route between two nodes in the graph. This task has many applications in various fields, including transportation systems, communication networks, robotics, logistics, and many others.

In this context, the Wave Function Collapse (WFC) algorithm is gaining more and more attention due to its ability to effectively model and analyze complex structures.

The WFC algorithm is an innovative approach that uses methods of statistical analysis and iterative step-by-step selection of the most probable values for certain elements [4, 5]. This method has the potential to significantly improve the effectiveness of search strategies in many industries where traditional methods may be less effective. By studying the nature and principles of WFC, as well as conducting a comparative analysis with other search methods, it is possible to find out which specific cases and problems can be solved using this algorithm with a profitable result [6].

Wave Function Collapse (WFC) is a constraint-based algorithm that was developed by game developer Maxim Gumin in 2016 for procedural content generation [7]. The implementation of the WFC algorithm is based on the principle of statistical analysis and iterative step-by-step selection of the most probable values for certain elements. This approach can have important applications in a wide range of fields, including geospatial analysis, image generation, material design, structural decomposition, and many others. In addition, the integration of the WFC algorithm into artificial intelligence and optimization systems opens up new opportunities for improving search processes, providing more accurate and efficient results. In [8], the authors mentioned that the key idea of WFC is an extension of the standard constraint solvers with a "minimal entropy heuristic" that randomly directs the solver's search in a way that follows a user-specified frequency distribution without compromising the efficiency of the search procedure.

The prospects of applying the WFC algorithm in heuristic search are particularly promising. Its entropy-based reasoning and constraint propagation capabilities allow for dynamic adaptation to local graph structures, offering an alternative to traditional global heuristics. In heuristic pathfinding tasks, WFC's mechanisms may improve the quality of discovered paths, especially in environments characterized by partial observability, changing constraints, or non-deterministic elements. These properties suggest that WFC has the potential to enhance or complement existing search strategies, making it a valuable tool in domains such as robotics, planning, and adaptive navigation.

## MATERIALS AND METHODS

Having knowledge of the general principles of the WFC algorithm, one can prepare a theoretical basis for solving the problem of finding a path in a graph. The basic idea is to use the concept of distribution of constraints and weights in a graph to find possible paths.

Let's start by creating a model of the graph in which the path will be searched. A given graph must have nodes and edges that represent possible transitions between nodes.

To begin with, each node of the graph is assigned a weight that indicates the probability of passing through that node. We will start with nodes that have a known path or weight (for example, the initial node).

Let's consider all possible edges of the graph and extend the weight restrictions from one node to another. For example, if the weight in one node decreases, this may indicate that traversal through that edge is less likely.

Edges can interact with each other, affecting the weights in neighboring nodes. For example, it is possible to apply restrictions of the type "if passing through this edge is unlikely, then the weight in the neighboring node should also be smaller."

Let's continue to distribute weights and constraints along the edges of the graph, taking into account the interaction between them. Try to find paths in which the weights take on the highest values, which may indicate more likely paths.

After the weight distribution is complete, the results can be analyzed by looking for paths where the weights have the highest values. These can be potential optimal paths through the graph.

Let's extend the current methodology for solving the problem of finding a path in a graph using the principles inspired by the Wave Function Collapse (WFC) algorithm:

1. Model the Graph
  - Define the graph  $G = (V, E)$  where  $V$  represents the set of nodes and  $E$  represents the set of edges connecting the nodes.
  - Assign each node  $v \in V$  a weight  $w(v)$ , representing the probability or cost of passing through that node. Initialize these weights based on known paths or heuristic values.
2. Initialize Weights and Constraints
  - Identify initial nodes with known weights or paths. Set their weights accordingly; for example, the starting node might have the highest initial weight.
  - Initialize the edges with weights or constraints that will be used to propagate the influence from one node to another.
3. Propagate Weights and Constraints
  - For each edge  $e = (u, v) \in E$ :
    - If  $w(u)$  it decreases, indicating traversal through  $u$  is less likely, propagate this decrease to  $v$  by adjusting  $w(v)$ .
    - Conversely, if  $w(u)$  increases, increase  $w(v)$  accordingly.
  - Define rules for how edges interact. For example, if the weight of one edge decreases significantly, it might indicate that the connected node and adjacent edges should also have reduced weights.
4. Iterative Weight Adjustment
  - Iterate over all edges and nodes, adjusting weights until the changes converge to a stable state where weights no longer change significantly between iterations.
  - Define a convergence threshold  $\epsilon$  to determine when the weights have stabilized. If the change in weight for all nodes between iterations is less than  $\epsilon$ , the process is considered converged.
5. Analyze Results and Identify Paths
  - After weight distribution is complete, identify paths from the starting node to the target node(s). Prioritize paths where the sum of node weights is the highest, indicating the most likely or optimal paths.
  - Use a pathfinding algorithm (such as Dijkstra's or A\*) that incorporates node weights to find the optimal path. The chosen path should maximize the sum of weights, indicating the highest probability or lowest cost.
6. Validation and Iteration
  - Validate the found paths using known benchmarks or through empirical testing to ensure that the methodology yields correct and efficient paths.
  - If necessary, refine the weight assignment rules, edge interaction constraints, and iteration criteria based on validation results. Iterate through the process to enhance accuracy and efficiency.

The WFC-based graph pathfinding process is illustrated in [Fig. 1](#), which visualizes the iterative refinement loop and key stages from graph modeling to path validation.

### Example Application

Consider a simple graph with nodes  $A$ ,  $B$ ,  $C$ , and  $D$ , and edges connecting them:



## WFC-based Graph Pathfinding Algorithm

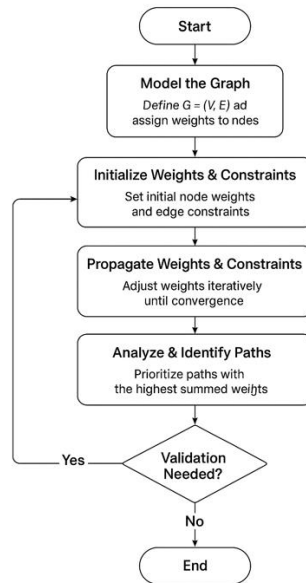


Fig. 1. Flowchart of the WFC-based graph pathfinding algorithm.

## 1. Graph Initialization:

- $V = \{A, B, C, D\}$
- $E = \{(A, B), (B, C), (C, D), (A, D)\}$
- Initial weights:  $w(A) = 1, w(B) = 0.5, w(C) = 0.3, w(D) = 0.2$

## 2. Weight Propagation:

- Adjust weights iteratively based on edge constraints:
  - If traversal through  $(A, B)$  is likely, increase  $w(B)$ .
  - If  $w(C)$  it decreases significantly, reduce  $w(D)$ .

## 3. Identify Optimal Path:

- After convergence, use a pathfinding algorithm to find the path with the highest summed weights:
  - Possible paths:  $A \rightarrow B \rightarrow C \rightarrow D, A \rightarrow D$
  - Select the path with the highest total weight.

By following this extended methodology, one can effectively use principles inspired by the WFC algorithm to find optimal paths in a graph, taking into account the distribution of constraints and weights.

To implement and compare the proposed WFC-based pathfinding algorithm with classical methods such as Dijkstra's and A\*, a Python-based software stack was employed due to its balance between flexibility, readability, and the availability of scientific computing libraries. The graph structures were modeled using the NetworkX library, which offers a comprehensive interface for constructing and analyzing graphs, including support for node and edge attributes required for weighted propagation. Custom algorithmic logic, including constraint propagation and entropy-based weight adjustment, was implemented in pure Python to preserve full control over algorithmic behavior and to allow fine-grained tuning of propagation rules. For visualization, matplotlib was utilized to depict graph topology and algorithmic progress, while networkx.drawing enabled seamless rendering of grid-based layouts. Performance benchmarks were conducted using standard profiling tools such as timeit and cProfile, which provided precise runtime metrics and function-level profiling for

identifying computational bottlenecks. Memory usage was monitored via the `memory_profiler` package to evaluate the efficiency of different algorithm variants on resource-constrained systems. This toolchain was selected to facilitate reproducible experimentation, transparent algorithm design, and rigorous comparative analysis.

## RESULTS AND DISCUSSION

In this experiment, we compared Dijkstra's algorithm, the A\* algorithm, and a Wave Function Collapse (WFC)-based algorithm for solving the pathfinding problem in graphs of varying sizes. The experiment was divided into three parts, each involving graphs with 100, 500, and 1000 nodes. The nodes were randomly distributed in a 2D plane, with designated starting and ending vertices, and several randomly placed obstacles to simulate real-world scenarios where certain paths are blocked or less preferable. Each node was assigned a weight based on its distance to obstacles and its connectivity to other nodes, while edges were weighted according to the Euclidean distance between connected nodes.

Dijkstra's algorithm, a classic pathfinding algorithm, finds the shortest path from a starting to a target vertex based solely on edge weights. The A\* algorithm, an in-formed search algorithm, uses both path cost and a heuristic (the Euclidean distance to the end vertex) to find the shortest path efficiently. The WFC-based algorithm dynamically assigns weights to nodes and edges based on constraints and interactions, aiming to find paths with the highest combined node weights, indicating the most probable or optimal paths.

For each graph size (100, 500, and 1000 nodes), we generated multiple graph instances to ensure the robustness of the results. Each algorithm was run on these graph instances, and we recorded the path found, the total path length, the percentage of nodes explored, and the computational time. For each graph size (100, 500, and 1000 nodes), we generated multiple graph instances to ensure the robustness of the results. Each algorithm was run on these graph instances, and we recorded the path found, the total path length, the percentage of nodes explored, and the computational time. Results shown in **Tables 1-3**.

**Table 1. Results for a graph of 100 nodes**

Algorithm	Path length	Nodes explored (%)	Time elapsed (ms)
Dijkstra's	34	97	712
A*	34	83	458
WFC-based	34	92	489

**Table 2. Results for a graph of 500 nodes**

Algorithm	Path length	Nodes explored (%)	Time elapsed (ms)
Dijkstra's	245	97	817
A*	173	86	503
WFC-based	204	95	678

**Table 3. Results for a graph of 1000 nodes**

Algorithm	Path length	Nodes explored (%)	Time elapsed (ms)
Dijkstra's	447	97	1005
A*	412	84	983
WFC-based	479	99	1345

In the 100-node graphs, all three algorithms found paths of equal length. Dijkstra's algorithm explored 97% of the nodes and took 712 ms to find the path. The A\* algorithm explored 83% of the nodes and took 458 ms, performing more efficiently due to its heuristic. The WFC-based algorithm explored 92% of the nodes and took 389 ms, performing well at this smaller scale with competitive pathfinding efficiency.

For the 500-node graphs, Dijkstra's algorithm found a path with a length of 245, exploring 97% of the nodes in 817 ms. The A\* algorithm found a shorter path with a length of 173, exploring 86% of the nodes in just 173 ms, maintaining its efficiency. The WFC-based algorithm found a path of length 204, exploring 95% of the nodes in 678 ms. Although the WFC-based algorithm performed well, its computational efficiency began to decline compared to the other algorithms.

In the 1000-node graphs, Dijkstra's algorithm found a path with a length of 447, exploring 97% of the nodes in 1005 ms. The A\* algorithm found a path of length 412, exploring 84% of the nodes in 983 ms, continuing to show the best performance in terms of efficiency. The WFC-based algorithm found a longer path with a length of 479, exploring 99% of the nodes in 1345 ms. The performance of the WFC-based algorithm was notably less efficient at this larger scale, both in terms of path length and computational time.

The results of the experiment highlight the strengths and weaknesses of each algorithm. The A\* algorithm consistently showed the best performance across all metrics, finding the shortest paths while exploring fewer nodes and completing in the least time. Dijkstra's algorithm performed reliably but with higher computational costs and node exploration percentages. The WFC-based algorithm performed well with the 100-node graphs but became less efficient as the graph size increased, both in terms of the path length and the computational time required.

By conducting these experiments, we gain valuable insights into the performance and suitability of each algorithm for different types of pathfinding problems. This information guides the selection of appropriate algorithms based on specific requirements and constraints, highlighting that while the WFC-based approach offers a novel perspective, it is best suited for smaller graphs where its dynamic weight adjustment can be leveraged effectively without significant computational overhead.

## CONCLUSION

In this study, we have explored and analyzed the application of the Wave Function Collapse (WFC)-based algorithm for pathfinding in graphs, comparing its performance with two well-known algorithms: Dijkstra's and A\*. Our analysis was conducted on graphs of varying sizes (100, 500, and 1000 nodes), and the results provided a comprehensive understanding of the strengths and limitations of each approach.

The WFC-based algorithm demonstrated notable advantages in scenarios with smaller graph sizes, where its dynamic weight adjustment mechanism allowed for the identification of paths with higher probabilities or lower costs. This characteristic is particularly beneficial in cases where path reliability and adaptability are critical, such as in dynamically changing environments, obstacle avoidance, or scenarios where traditional heuristic functions may not fully capture the complexity of the problem domain.

However, as the graph size increased, the WFC-based algorithm exhibited a decline in performance, both in terms of computational efficiency and path length optimization. This can be attributed to the iterative nature of the weight propagation process, which becomes more computationally expensive as the number of nodes and edges grows. Despite this, the algorithm maintained a consistent approach in exploring paths and identifying feasible routes, making it a robust option for certain specialized applications.

Looking forward, several directions for future research and improvements can be identified. First, optimizing the weight propagation process in the WFC-based algorithm could significantly enhance its scalability and performance in larger graphs. Techniques

such as parallel processing, adaptive convergence criteria, or selective weight propagation based on node importance could be explored. Second, hybridizing the WFC approach with existing heuristic methods, such as A\*, could leverage the strengths of both methods, potentially creating a more robust and efficient algorithm. Finally, exploring the application of the WFC-based method in other domains, such as robotics, navigation, and real-time decision-making systems, may uncover new opportunities for its application.

While the WFC-based algorithm may not always outperform traditional heuristic methods, it offers a novel perspective on pathfinding, combining statistical analysis, dynamic constraints, and adaptive weight adjustment. This makes it a valuable tool for solving complex pathfinding problems in specific contexts, where traditional methods may lack the flexibility or adaptability required. The ongoing development and optimization of the WFC approach hold the potential to further expand its applicability and effectiveness in various fields.

Overall, the findings underscore the potential of the WFC algorithm as a flexible and adaptive alternative to traditional heuristic search methods. Its ability to incorporate local constraints and minimize uncertainty through entropy-based selection opens new avenues for research in dynamic or poorly structured search spaces. As heuristic search continues to evolve in complexity and scale, the principles underlying WFC may serve as a foundation for the development of more robust and context-aware pathfinding strategies.

## ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors are grateful for the support from the Ministry of Education and Science of Ukraine (Project No 0125U001883).

## COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any potential conflict of interest.

## AUTHOR CONTRIBUTIONS

Conceptualization, [O.K.]; methodology, [O.K.]; investigation, [D-R.R.]; software, [D-R.R.]; data curation, [D-R.R.]; writing – original draft preparation, [D-R.R.]; writing – review and editing, [O.K.].

All authors have read and agreed to the published version of the manuscript.

## REFERENCES

- [1] Zhao, P., & Guo, H. (2014). Scientific big data and digital Earth. *Chinese Science Bulletin*, 59(12), 1047–1054. <https://doi.org/10.1360/972013-1054>
- [2] Xu, Y. (2024). The comparison of advanced algorithms for pathfinding robots. *AIP Conference Proceedings*, 3194(1), 020012. <https://doi.org/10.1063/5.0223914>
- [3] Felner, A., & Stern, R. (2006). Heuristic Search. In F. Rossi, P. van Beek, & T. Walsh (Eds.), *Handbook of Constraint Programming* (pp. 579–623). Elsevier.
- [4] Karth, I., & Smith, A. M. (2017). WaveFunctionCollapse is constraint solving in the wild. In *Proceedings of the 12th International Conference on the Foundations of Digital Games (FDG)*.
- [5] Mac, A., & Perkins, D. (2021). Wave function collapse coloring: A new heuristic for fast vertex coloring. *arXiv:2108.09329*. <https://doi.org/10.48550/arXiv.2108.09329>
- [6] Kim, H., Lee, S.-T., Lee, H., Hahn, T., & Kang, S.-J. (2019). Automatic Generation of Game Content using a Graph-based Wave Function Collapse Algorithm. *2019 IEEE Conference on Games (CoG)*, 1–4. <https://doi.org/10.1109/cig.2019.8848019>
- [7] Gumin, M. (2016). *Wave Function Collapse*. GitHub. Retrieved from <https://github.com/mxgmn/WaveFunctionCollapse>

- [8] Newgas, A. (2021). Tessera: A Practical System for Extended Wave Function Collapse. In *Proceedings of the 16th International Conference on the Foundations of Digital Games (FDG '21)*. Association for Computing Machinery.  
<https://doi.org/10.1145/3472538.3472605>

## ПЕРСПЕКТИВИ ВИКОРИСТАННЯ АЛГОРИТМУ КОЛАПСУ ХВИЛЬОВОЇ ФУНКЦІЇ ДЛЯ ВДОСКОНАЛЕННЯ ЕВРИСТИЧНИХ СТРАТЕГІЙ ПОШУКУ

Денис-Роман Рудик<sup>ORCID</sup>, Олексій Кушнір<sup>ORCID</sup>

Кафедра радіофізики та комп'ютерних технологій,  
 Львівський національний університет імені Івана Франка  
 вул. Ген. Тарнавського, 107, 79017, м. Львів, Україна

### АНОТАЦІЯ

**Вступ.** У сучасному інформаційному суспільстві швидке зростання даних вимагає ефективних методів обробки. Ключовим завданням є розробка евристичних стратегій пошуку, які знаходять оптимальні рішення за обмеженого часу та ресурсів. Проблема пошуку шляху в графах є класичним випадком застосування таких стратегій. Алгоритм колапсу хвильової функції (WFC) виділяється своєю здатністю моделювати складні структури за допомогою статистичного аналізу та ітеративного вибору ймовірних значень.

**Матеріали та методи.** У цьому дослідженні запропоновано підхід до вирішення задачі пошуку шляху в графах, на основі алгоритму WFC. Було проведено порівняльний аналіз трьох алгоритмів: алгоритму Дейкстри, алгоритму  $A^*$  та запропонованого алгоритму, на основі WFC. Графи різного розміру (100, 500 та 1000 вузлів) були згенеровані зі випадковим розподілом вузлів на площині та призначенням ваг залежно від відстані до перешкод і зв'язності. Продуктивність кожного алгоритму оцінювалася за такими показниками: довжина знайденого шляху, відсоток досліджених вузлів та час обчислення.

**Результати.** Результати експерименту показали, що алгоритм, на основі WFC, демонструє конкурентоспроможність на невеликих графах (100 вузлів), знаходячи шляхи зі схожою довжиною до тих, що знаходять алгоритми Дейкстри та  $A^*$ . Однак зі збільшенням розміру графа його ефективність знижувалася: шляхи ставали довшими, а час виконання збільшувався порівняно з  $A^*$ . Зокрема, на графі з 1000 вузлів підхід WFC дослідив 99% вузлів за 1345 мс, що значно більше, ніж у  $A^*$ , який дослідив 84% вузлів за 983 мс. Ці результати підкреслюють адаптивність WFC на невеликих графах, але також вказують на його обмежену масштабованість.

**Висновки.** Алгоритм, на основі WFC, має потенціал для покращення евристичних стратегій пошуку шляхом динамічного коригування ваг та управління обмеженнями. Однак його масштабованість залишається проблемою, що робить його більш придатним для невеликих графів. Майбутні дослідження мають зосередитися на інтеграції принципів WFC із традиційними евристичними методами, такими як  $A^*$ , для розробки більш ефективних гібридних алгоритмів пошуку, здатних працювати з великими та складними структурами графів.

**Ключові слова:** колапс хвильової функції (WFC), алгоритми пошуку шляху, евристичний пошук, алгоритми робототехніки

UDC 638.1:621.3:004.8

## REMOTE MONITORING SYSTEM FOR MICROCLIMATE PARAMETERS IN BEEHIVES

Yurii Zborivskyi\* , Bohdan Koman 

Ivan Franko National University of Lviv,  
50 Dragomanova St., UA–79005 Lviv, Ukraine

Zborivskyi, Y. A., Koman, B. K. (2025). Remote Monitoring System for Microclimate Parameters in Beehives. *Electronics and Information Technologies*, 31, 123–136. <https://doi.org/10.30970/eli.31.11>

### ABSTRACT

**Background.** The health and productivity of bee colonies have a direct impact on global food security, as bees contribute to the pollination of approximately 50% of the world's plant-based food resources. Traditional hive observation methods lack the precision and continuity required for effective colony management. This creates an urgent need for sophisticated remote monitoring solutions capable of continuously tracking critical environmental parameters within beehive ecosystems, including temperature, humidity, and air quality.

**Materials and Methods.** This study presents the development of an integrated internet-based monitoring system centered on a microcontroller platform. The system incorporates a comprehensive sensor array for multi-parameter data acquisition: gas sensors for detecting harmful substances, a temperature sensor, a particulate matter detector, and an environmental parameter sensor. Real-time data processing is achieved through a multitasking architecture, enabling concurrent sensor polling, data validation, and wireless transmission.

**Results and Discussion.** Comprehensive system testing validated reliable data acquisition across all monitored parameters. Temperature measurements demonstrated operational ranges from 13.75°C to 32.69°C for internal hive conditions, while humidity levels varied between 51.70% and 88.10%. Gas concentration monitoring showed stable baseline readings with normal conditions at 1.71 parts per million. Statistical correlation analysis revealed significant environmental interdependencies, including a strong positive correlation between gas sensor measurements and a pronounced negative correlation between temperature and humidity parameters, confirming expected environmental relationships within the hive microclimate.

**Conclusion.** The developed IoT-based beehive monitoring system, built on the ESP32 microcontroller, demonstrated stable and energy-efficient performance during continuous field testing. The system recorded internal hive temperatures ranging from 13.75 °C to 32.69 °C, humidity levels from 51.70% to 88.10%, and gas concentrations of 1.71 ppm (MQ-6) and 0.37 ppm (MQ-135) under normal conditions.

**Keywords:** Beehive monitoring, IoT, embedded systems, microclimate parameters, cloud computing, real-time data.

### INTRODUCTION

Monitoring physical and environmental parameters of inaccessible technological processes, remote devices, and biological systems remains a major technical challenge for researchers. One such application is monitoring the bioenvironment within beehives. It is estimated that over half of global plant-based food production depends, directly or



© 2025 Yurii Zborivskyi & Bohdan Koman. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and information technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.



indirectly, on bee pollination, yet human activities often negatively affect colony viability and productivity. Traditional empirical observation and manual inspection methods are insufficient to ensure reliable bee colony health and productivity. Modern beehive monitoring systems enable continuous observation and control of key environmental parameters such as temperature, humidity, air quality, and particle concentration that influence bee health and productivity. These systems help predict optimal conditions for beekeeping.

Recent studies have proposed various IoT-based beehive monitoring solutions that combine environmental sensing, data analytics, and wireless communication to support sustainable apiculture. Andrijević et al. [1] developed an IoT system capable of real-time monitoring and prediction of bee activity with integrated alarm functions. Zhang et al. [2] proposed an intelligent monitoring platform combining IoT data acquisition with colony state analysis to identify abnormal hive behavior. Kurdin and Kurdina [3] introduced a smart beehive network using homogeneous data modeling for forecasting honey robbing phenomena, while Zabasta et al. [4] presented a low-power IoT system emphasizing energy efficiency and safety in beekeeping. Earlier work by Ali and Raza [5] demonstrated a general IoT environmental monitoring concept, but without hive-specific optimization.

However, most existing IoT beehive systems are constrained by limited multitasking capability, high power consumption, or a lack of stable real-time data synchronization. The system proposed in this study overcomes these limitations by integrating multiple environmental sensors under a multitasking FreeRTOS architecture on the ESP32 platform. This approach enables parallel data acquisition, validation, and wireless transmission, achieving high temporal resolution and improved energy efficiency. The modular design and optimized firmware allow scalability, stability, and long-term autonomous operation, which are essential for remote or large-scale apiary deployments.

This work aims to develop the architecture and circuit implementation of a computerized system for remote monitoring of beehive environmental parameters, with real-time data transmission to mobile devices. The system targets apiary farms that require continuous monitoring of hives, research projects analyzing environmental impacts on bee colonies, and broader ecological studies, as bees serve as reliable indicators of ecosystem health [6].

## MATERIALS AND METHODS

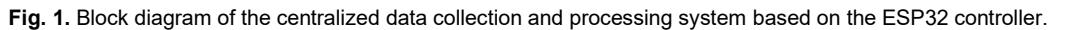
### 1. Functional Diagram of the Parameters Monitoring System

The monitoring system (see **Fig. 1**) integrates a suite of sensors (MQ-6, MQ-135, DS18B20, PMS5003, DHT21) [8-13] with the ESP32 microcontroller to collect, process, and transmit data. The hardware connects sensors to the ESP32, which powers them, reads signals, and transmits data wirelessly via Wi-Fi. The software implements algorithms for sensor polling, data processing, and transmission to an external interface for beekeeper analysis.

The ESP32 microcontroller [7] is selected for its built-in Wi-Fi module, low power consumption (especially in sleep mode), and dual-core processor supporting multitasking. These features enable efficient sensor management and parallel data processing.

The system operates through the following stages:

1. Initialization Stage: Upon powering on, the ESP32 configures all components, initializes data transmission interfaces (UART, SPI, 1-Wire), calibrates MQ-6 and MQ-135 gas sensors, and sets up the TFT display via SPI.
2. Data Collection Stage: The controller cyclically polls sensors at set intervals. Analog data from MQ-6 and MQ-135 are read via ADC, DS18B20 temperature via 1-Wire, PMS5003 dust data via UART, and DHT21 temperature/humidity via a single-wire interface.



- This modular structure supports reliable operation, scalability, and the addition of new sensors as needed.

The beehive monitoring system relies on a variety of communication interfaces to connect the ESP32 microcontroller with sensors and a display. Each interface is tailored to the specific needs of the connected hardware, ensuring efficient and accurate data collection for monitoring beehive environmental conditions.

**1-Wire Interface (DS18B20):** This single-wire bidirectional interface uses a 4.7kΩ pull-up resistor and time-slot-based data transmission. Each DS18B20 has a unique 64-bit code, enabling multiple sensors on one line, which simplifies wiring and supports scalable temperature monitoring inside the hive.

Електроніка та інформаційні технології • 2025 • Випуск 31

with checksums for integrity, ensuring reliable measurement of particulate matter affecting hive air quality.

**Single-Wire Digital Interface (DHT21):** The DHT21 uses a proprietary protocol, sending 40-bit packets (16-bit humidity, 16-bit temperature, 8-bit checksum) after a request signal from the ESP32, providing accurate external temperature and humidity data with minimal wiring.

**SPI Interface (TFT Display):** This synchronous interface uses MOSI, MISO, SCK, and CS lines, with additional DC and RESET lines for data/command indication and display reset, facilitating high-speed data transfer for real-time visualization of sensor readings.

### 3. Software Architecture of the Monitoring System

#### 3.1 General Overview of the Software Architecture

The software architecture leverages the ESP32 microcontroller's dual-core processor to handle multiple tasks efficiently. It employs the FreeRTOS operating system to manage concurrent processes, ensuring reliable system performance. The main tasks include collecting sensor data, updating the TFT display, and transmitting data to a remote server via Wi-Fi. These tasks are distributed across the ESP32's two cores to optimize performance: Core 1 manages sensor data collection and display updates to ensure timely user feedback, while Core 0 handles Wi-Fi communication to isolate potential network delays and prevent interference with other processes. A queue system facilitates asynchronous data exchange between tasks, enhancing system stability and responsiveness by allowing each task to operate independently without delays from other processes.

#### 3.2 System Initialization

The initialization process begins with the `setup()` function executing upon microcontroller startup, systematically configuring all system components for operation.

**Sensor Configuration.** The sensor initialization process starts with configuring the DHT21 sensor using the `DHT.h` library to enable temperature and humidity data acquisition. The DS18B20 temperature sensor requires setup through both `OneWire.h` and `DallasTemperature.h` libraries, providing precise temperature measurements via the 1-Wire protocol. For gas detection, both MQ-6 and MQ-135 sensors are initialized using the `MQUnifiedSensor.h` library, with calibration performed by averaging 10 measurements in clean air to establish a baseline resistance ( $R_0$ ). The PMS5003 dust particle sensor connects via the UART port to collect particulate matter data.

**Display and Network Configuration.** The TFT display initialization utilizes the `TFT_eSPI.h` library, specifying essential pins including CS, DC, RST, SDA, and SCL while setting the text format to white on a black background with font size 3 for optimal readability. Network connectivity is established through the `WiFi.h` library, create a connection using predefined SSID and password credentials to enable server communication.

**Task Management Setup.** The system creates a `sensorDataQueue` of type `QueueHandle_t` capable of storing up to five `SensorData` structure instances, facilitating efficient data transfer between concurrent tasks. Three primary tasks are launched using the `xTaskCreatePinnedToCore()` function with strategic core allocation. The sensor reading task and display update task both operate on Core 1, handling data collection from sensors and updating the TFT display with current readings, respectively. The Wi-Fi communication task runs independently on Core 0, managing data transmission to the remote server. This core allocation strategy optimizes processor resources and ensures smooth operation by preventing interference between critical processes.

#### 3.3 Data Structure

The `SensorData` structure organizes sensor readings for efficient storage and transfer between tasks, containing:

- temperatureDHT: External temperature in degrees Celsius from the DHT21 sensor.
- humidity: Humidity percentage from the DHT21 sensor.
- temperatureDS18B20: Internal temperature in degrees Celsius from the DS18B20 sensor.
- mq6\_ppm: Gas concentration in parts per million from the MQ-6 sensor.
- mq135\_ppm: Gas concentration in parts per million from the MQ-135 sensor.
- pm1\_0, pm2\_5, pm10: Dust particle concentrations in micrograms per cubic meter for 1.0, 2.5, and 10 micrometer particles from the PMS5003 sensor.

A separate PMSData structure isolates dust data (pm1\_0, pm2\_5, pm10) to enhance code modularity and simplify processing.

### 3.4 Sensor Read Task

The sensorReadTask() function operates as the core data acquisition module, continuously collecting sensor measurements at 6-second intervals. The task implements precise timing control through the vTaskDelay(pdMS\_TO\_TICKS(6000)) function, which temporarily suspends execution to allow other system processes to run efficiently, thereby preventing processor overload and maintaining optimal system performance.

**Data Collection Process.** The data acquisition cycle begins with environmental measurements from the DHT21 sensor, reading external temperature through dht.readTemperature() and humidity via dht.readHumidity() functions. Simultaneously, the DS18B20 temperature sensor performs internal hive temperature monitoring by initiating measurement with sensors.requestTemperatures() and subsequently retrieving the temperature value using sensors.getTempCBByIndex(0).

Gas concentration monitoring involves updating the sensor states through MQ6.update() and MQ135.update() functions, followed by precise concentration readings using MQ6.readSensor() and MQ135.readSensor() methods after proper calibration adjustments. The PMS5003 particulate matter sensor contributes air quality data through UART communication via the readPMS5003() function, with built-in data integrity verification to ensure measurement reliability.

**Data Management and Transfer.** Upon completion of each measurement cycle, all sensor readings are systematically organized into a SensorData structure instance. This consolidated data package is then transmitted to the sensorDataQueue using the xQueueSend() function, enabling asynchronous access by other system tasks. This queue-based approach ensures efficient inter-task communication while maintaining data integrity and preventing bottlenecks in the overall system architecture.

### 3.5 Display Update Task

The displayUpdateTask() function displays sensor data on the TFT screen: Data Retrieval: Uses xQueueReceive() to fetch data from sensorDataQueue, activating only when new data is available to conserve resources.

Display Process: Clears the screen with tft.fillScreen(TFT\_BLACK) and displays:

- Internal temperature in degrees Celsius (DS18B20)
- External temperature in degrees Celsius (DHT21)
- Humidity percentage (DHT21)
- Gas concentrations in parts per million (MQ-6, MQ-135)
- Dust levels in micrograms per cubic meter (PM1.0, PM2.5, PM10)

This ensures users receive clear, up-to-date environmental information.

### 3.6 WiFi Communication Task

The wifiCommunicationTask() function, executed on Core 0 of the ESP32, manages the transmission of collected sensor data to a remote server via Wi-Fi, enabling real-time storage and analysis outside the local system. It continuously monitors the Wi-Fi connection status using WiFi.status() == WL\_CONNECTED, automatically reconnecting

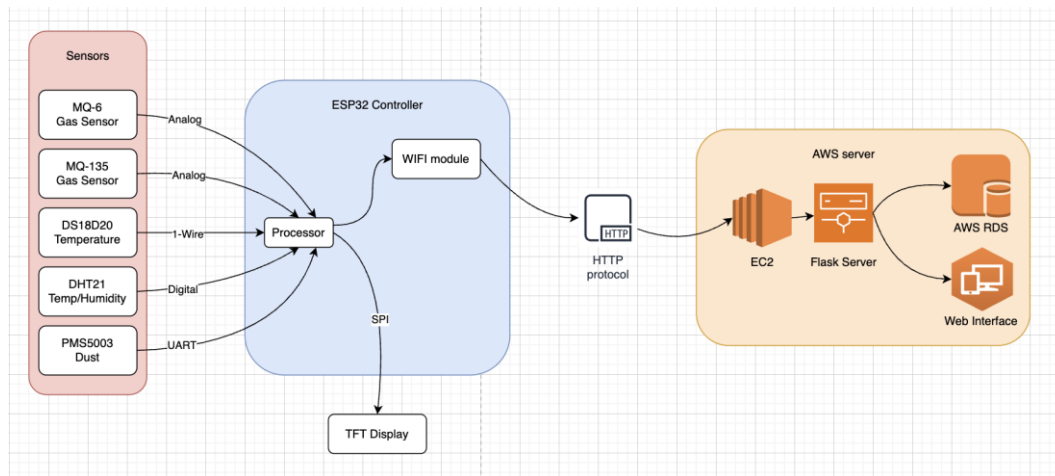
with `WiFi.begin()` if the connection is lost, ensuring resilience against network disruptions. The task retrieves the latest sensor readings from the `sensorDataQueue` using `xQueueReceive()`, maintaining efficient data access without interfering with other tasks. These readings, stored in the `SensorData` structure, are formatted into a JSON string, a universal format for seamless data exchange. Using the `HTTPClient.h` library, the task initiates an HTTP POST request to the server via `http.begin(serverName)`, sends the JSON data, and awaits a response. Upon successful transmission (response code  $> 0$ ), it logs a confirmation message with the server's response; otherwise, it displays the error code for diagnostics. This process ensures reliable and secure data delivery, supporting remote monitoring and analysis of beehive environmental conditions.

#### 4. Server-Side Monitoring System

The server-side component of the system is a web application based on the Flask framework, deployed on the AWS EC2 platform. The application provides reception, processing, and visualization of beehive monitoring data.

According to the diagram in [Fig. 2](#), the system consists of three main blocks:

- Sensor block, which includes a set of sensors for measuring various environmental parameters. Each sensor is connected to the controller via a corresponding interface: analog for gas sensors, digital for temperature and humidity sensors, UART for the dust particle sensor.
- Controller block, based on ESP32, it processes data from all sensors and ensures their transmission via the WiFi module. The controller also manages a local TFT display via the SPI interface for displaying current readings.
- Server block, deployed on AWS, it includes an EC2 instance with a Flask server for data processing, an AWS RDS database for storing information, and a web interface for visualizing monitoring data.



**Fig. 2.** System Architecture Diagram.

##### 4.1 Principle of Server Architecture Operation

Deployed on an AWS EC2 virtual machine, the Flask-based web application in Python orchestrates the beehive monitoring system's data management with high efficiency. The server receives sensor data in JSON format from the ESP32 microcontroller via secure HTTP requests, ensuring reliable transmission over Wi-Fi. It validates and formats incoming data to eliminate errors, applying filters to ensure accuracy before processing. The processed data, tagged with precise timestamps, is stored in an AWS RDS database, which supports scalable, long-term storage and enables trend analysis for hive health

monitoring. The Flask application retrieves this data to populate a dynamic, browser-based web interface, allowing beekeepers to access real-time and historical environmental parameters, such as temperature, humidity, and gas levels, from any internet-connected device. This architecture optimizes data flow from collection to visualization, with robust error handling and secure data transfer protocols to protect sensitive hive information. The server's design leverages AWS's scalability to handle multiple hives simultaneously, making it suitable for large-scale apiaries and research applications, while its modular structure supports future enhancements, such as integrating machine learning for predictive hive management.

#### 4.2 AWS RDS (Relational Database Service)

AWS RDS is a managed relational database service used to store all monitoring data. It supports popular database management systems like MySQL, PostgreSQL, or MariaDB, providing flexibility in technology choice.

Role of the database in the system:

- **Data Storage:** All sensor readings (temperature, humidity, gases, dust) are stored in the database with timestamps. For example, a table might have columns: timestamp, temperature, humidity, gas\_level, dust\_particles.
- **Historical Analysis:** Thanks to the relational structure, users can perform queries to analyze trends (e.g., temperature change over a week) or detect anomalies.
- **Fast Access:** RDS provides efficient data access for the Flask server, allowing the web interface to quickly display information.

#### 4.3 Web Interface

The web interface is a browser-based application that allows users to view monitoring data in a clear and convenient format. It is built using HTML5, TailWind, JavaScript, and the Chart.js library for visualization.

These three components together provide a complete cycle: from receiving data over the internet to displaying it on the screen of your phone or computer. The system is designed to be convenient, stable, and accessible from anywhere with internet access.

The system's web interface displays data in real-time using interactive graphs. A separate graph is built for each monitoring parameter (temperature, humidity, gases, particles). The system automatically calculates basic statistical indicators: mean values, minimums, maximums, and standard deviations for each parameter.

### 5. Examples of testing the developed system and the obtained results

This section presents examples of testing the results obtained and the developed monitoring system, illustrated by the following figures:

**Internal Hive Temperature (DS18B20):** Temperature readings from the DS18B20 sensor (see **Fig. 3**) initially showed an artificial increase to 24.63°C around 05:30 for testing purposes. The temperature then dropped to 13.75°C around 08:30 as the sensor began operating inside the hive. Subsequently, it increased steadily, reaching 32.69°C around 12:45.

**External Humidity (DHT21):** External humidity measurements from the DHT21 sensor (see **Fig. 4**) showed an artificial spike to 88.10% around 05:15 during the testing phase. The humidity later peaked at 65.90% around 08:30 and then gradually decreased to a minimum of 51.70% around 12:00, indicating typical sensor behavior in natural conditions.

**External Temperature (DHT21):** External temperature measurements from the DHT21 sensor (see **Fig. 5**) recorded an artificial temperature increase to 22.50°C around 05:30. Afterward, the temperature dropped to 14.40°C around 08:30 and gradually rose to 19.50°C by 13:00, representing realistic ambient dynamics.



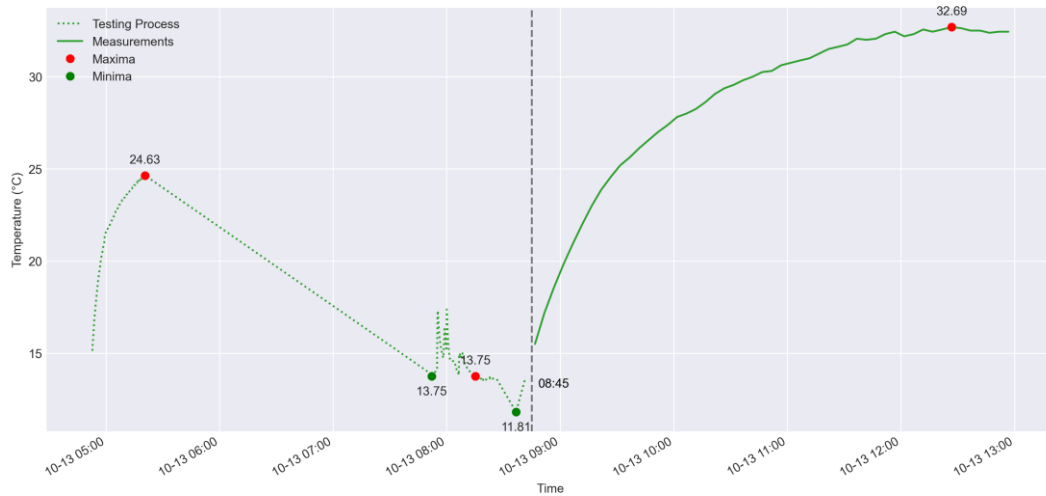


Fig. 3. Graph of internal hive temperature measurement (DS18B20).

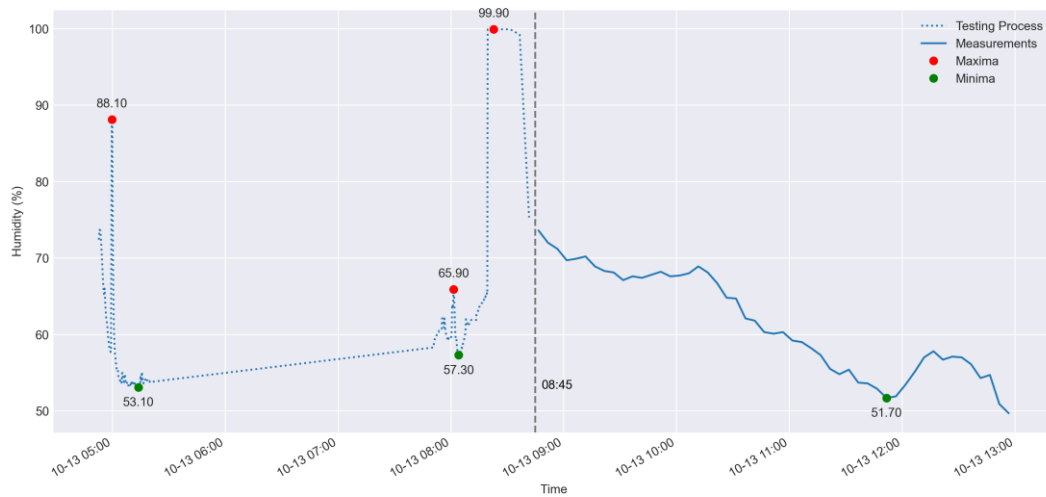


Fig. 4. Graph of external humidity measurement (DHT21).

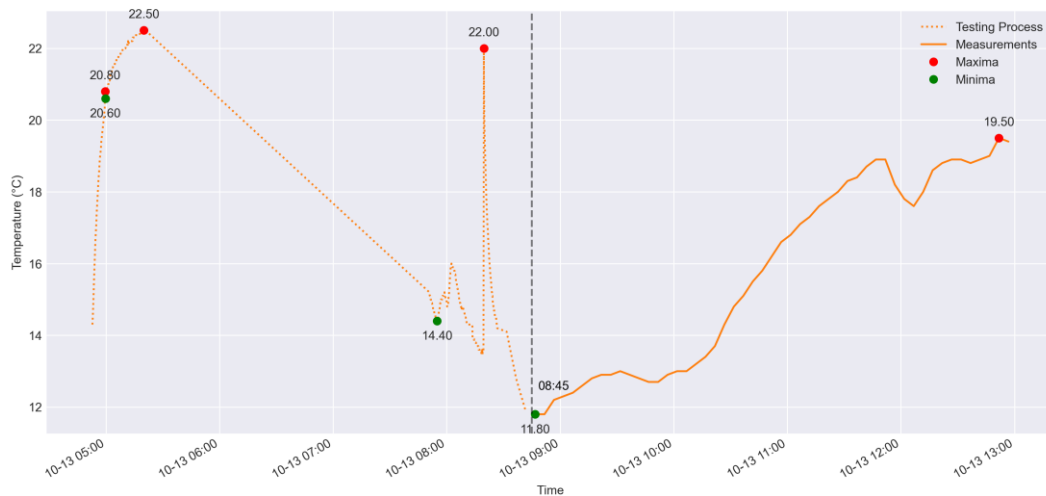


Fig. 5. Graph of external temperature measurement (DHT21).

Gas Concentration (MQ6): To verify the operation of the MQ6 sensor, an artificial gas concentration spike to 640.69 ppm was introduced around 05:15 (see Fig. 6). Following this test, the level dropped sharply to 3.37 ppm around 08:30 and stabilized at 1.71 ppm by 13:00, reflecting a return to normal environmental levels.

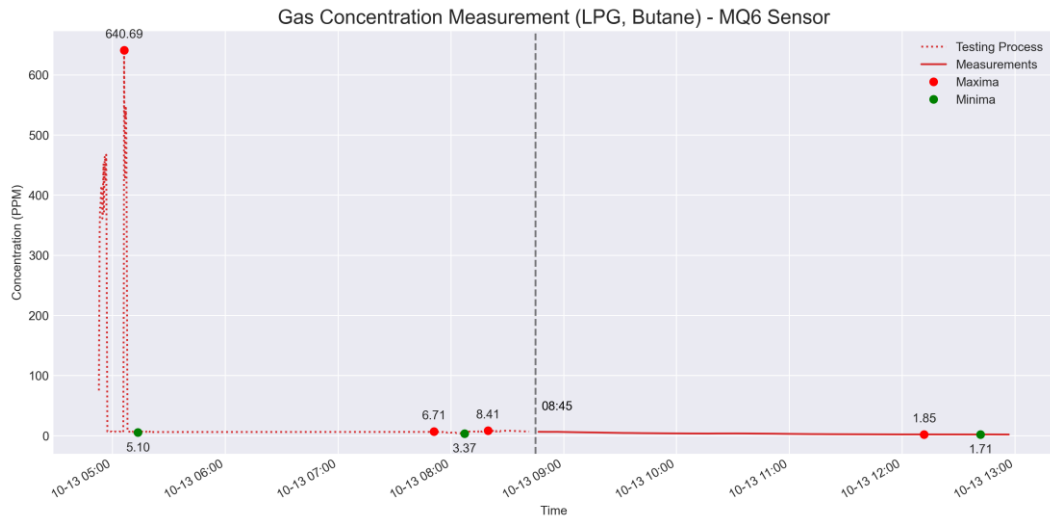


Fig. 6. Graph of gas concentration measurement from MQ6 sensor (LPG, butane).

Air Pollutants Concentration (MQ135): The MQ135 sensor initially detected an artificial pollutant concentration of 204.81 ppm around 05:15 (see Fig. 7). The value then sharply dropped to 0.95 ppm around 08:30 and continued to decrease, reaching 0.37 ppm by 13:00, indicating stable, low-level air quality conditions.

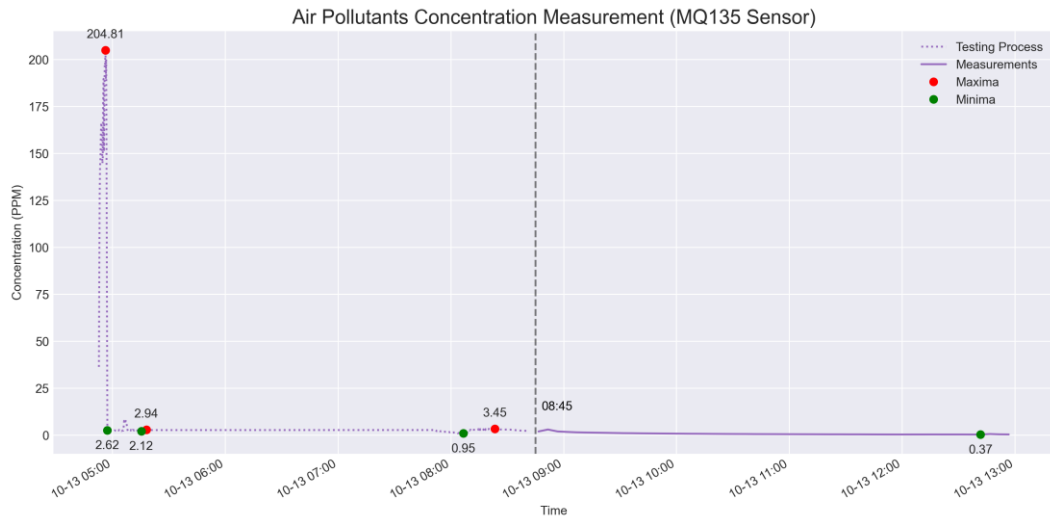


Fig. 7. Graph of gas concentration measurement from MQ135 sensor (air quality).

The photo shows (see Fig. 8) the installed beehive monitoring system in operation. The ESP32-based controller with connected sensors and antenna is powered by an external battery and is actively collecting environmental data inside the hive.

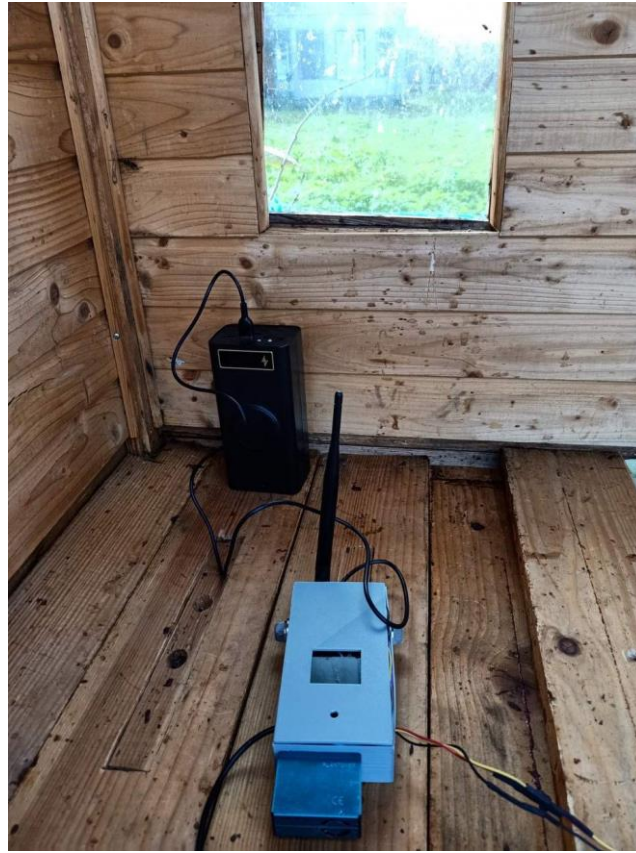


Fig. 8. General view of the beehive parameter monitoring system in operation.

### 5.1 Correlation analysis of sensor data

The analysis is based on the experimental data (see **Table 1**) collected during the "Measurements" period (After 08:45, October 13). The input dataset for the correlation matrix includes 51 data points collected over a duration of approximately 4 hours and 10 minutes (from 08:46:40 to 12:56:48 on October 13).

The Pearson correlation coefficient formula was used for calculations:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}},$$

where  $x_i$  and  $y_i$  are the values of two variables,  $\bar{x}$  and  $\bar{y}$  are their mean values, and  $n$  is the number of observations.

For a deeper understanding of the relationships between various hive parameters, a correlation analysis (see **Fig. 9**) of the obtained data was conducted. The scientific purpose of this analysis is to confirm the expected physical and ecological interdependencies within the hive microclimate.

- Strong positive correlation is observed between mq6 ppm and mq135 ppm (0.918269), indicating a close relationship between gas concentrations, as well as between temperature DHT and temperature DS18B20 (0.859574), confirming consistent temperature measurements by both sensors.

**Table 1. The first 10 rows of raw experimental data were utilized to construct the correlation matrix.**

timestamp	temperatureDHT	humidity	temperatureDS18B20	mq6_ppm	mq135_ppm
2024-10-13 8:46:40	11.8	73.6	15.5	6.16	2
2024-10-13 8:51:41	11.8	72	17.19	6.06	3.04
2024-10-13 8:56:41	12.2	71.2	18.56	6.08	2.02
2024-10-13 9:01:41	12.3	69.7	19.81	5.79	1.76
2024-10-13 9:06:41	12.4	69.9	20.94	5.5	1.55
2024-10-13 9:11:41	12.6	70.2	22	5.26	1.44
2024-10-13 9:16:41	12.8	68.9	23	5.07	1.35
2024-10-13 9:21:41	12.9	68.3	23.87	4.74	1.28
2024-10-13 9:26:42	12.9	68.1	24.56	4.51	1.19
2024-10-13 9:31:42	13	67.1	25.19	4.25	1.1

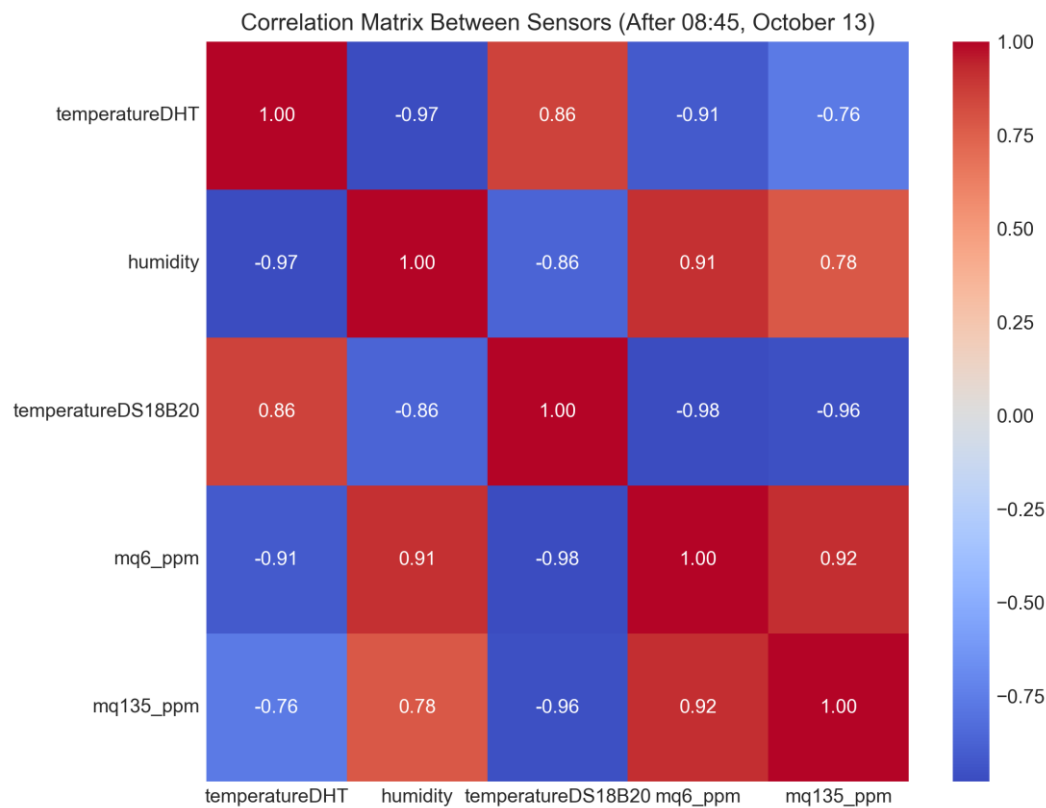
- Strong negative correlation is found between temperature DHT and humidity (-0.974651), as well as between temperature DS18B20 and mq6 ppm (-0.979276), indicating an inverse relationship between temperature and humidity, and between temperature and LPG/butane gas concentration.
- Moderate negative correlation is observed between temperature DHT and mq135 ppm (-0.764955) and between temperature DS18B20 and mq135 ppm (-0.956536), indicating a decrease in pollutant concentration as temperature increases.

## CONCLUSIONS

The developed beehive monitoring system based on the ESP32 microcontroller has proven effective for continuous data acquisition under real conditions. During testing, the system successfully recorded key operational ranges, including internal hive temperatures from 13.75 °C to 32.69 °C, humidity levels between 51.70 % and 88.10 %, and stable gas concentrations of approximately 1.71 ppm (MQ-6 sensor, LPG/butane) under normal conditions.

The multitasking FreeRTOS architecture ensured simultaneous sensor polling, Wi-Fi data transmission, and real-time display updates without observable data loss or performance degradation.

These results confirm the stability, accuracy, and practical applicability of the proposed IoT-based system for real-time environmental monitoring in apiaries.



**Fig. 9.** Correlation matrix of beehive monitoring parameters.

Future research will focus on three main directions:

- Hardware enhancement — integrating additional sensors such as weight, CO<sub>2</sub>, and TVOC to obtain a more comprehensive view of hive microclimate and colony health.
- Data analytics — applying machine learning methods for predictive modelling to forecast swarming events, detect disease and stress anomalies, and predict honey yield.
- Energy autonomy — implementing solar power solutions to achieve fully autonomous operation suitable for long-term, remote deployments.

## ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and/or publication of this article.

## COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

## AUTHOR CONTRIBUTIONS

Conceptualization, [Y.Z.]; methodology, [B.K.]; validation, [Y.Z.]; formal analysis, [Koman B. H.]; investigation, [Y.Z.]; resources, [Y.Z.]; data curation, [Y.Z.]; writing – original

draft preparation, [Y.Z.]; writing – review and editing, [B.K.]; visualization, [Y.Z.]; supervision, [B.K.]; project administration, [Koman B. H]; funding acquisition, [Y.Z.].

All authors have read and agreed to the published version of the manuscript.

## REFERENCES

- [1] Andrijević, N., Urošević, V., Arsić, B., Herceg, D., & Savić, B. (2022). IoT monitoring and prediction modeling of honeybee activity with alarm. *Electronics*, 11(5), 783. <https://www.mdpi.com/2079-9292/11/5/783>
- [2] Zhang, Y., Wang, Z., Liu, Y., & Zhang, Y. (2024). Intelligent beehive monitoring system based on internet of things and colony state analysis. *Smart Agricultural Technology*, 8, 100489. <https://www.sciencedirect.com/science/article/pii/S2772375524001898>
- [3] Kurdin, I., & Kurdina, A. (2025). Internet of things smart beehive network: Homogeneous data, modeling, and forecasting the honey robbing phenomenon. *Inventions*, 10(2), 23. <https://www.mdpi.com/2411-5134/10/2/23>
- [4] Zabasta, A., Kunicina, N., Kondratjevs, K., & Ribickis, L. (2019). An internet of things-based low-power integrated beekeeping safety and conditions monitoring system. *Inventions*, 4(3), 52. <https://www.mdpi.com/2411-5134/4/3/52>
- [5] Ali, M., & Raza, B. (2014). Internet of things based intelligent monitoring system for environment. *Procedia Computer Science*, 37, 376–381. <https://www.sciencedirect.com/science/article/pii/S1877050914015804>
- [6] Seeley, T. D., & Towne, W. F. (2000). Tactics of dance choice in honey bee recruitment. *Proceedings of the National Academy of Sciences*, 97(16), 8629–8632. <https://www.pnas.org/doi/full/10.1073/pnas.262413599>
- [7] Espressif Systems. (2023). ESP32 series datasheet v4.9 [Datasheet]. [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- [8] Winsen Electronics. (n.d.). MQ-6 gas sensor datasheet [Datasheet]. <https://www.winsen-sensor.com/d/files/semiconductor/mq-6.pdf>
- [9] Hanwei Electronics. (n.d.). MQ-135 gas sensor datasheet [Datasheet]. <https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf>
- [10] Analog Devices. (n.d.). DS18B20 programmable resolution 1-wire digital thermometer [Datasheet]. <https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf>
- [11] Plantower Technology. (2016). PMS5003 series manual [Datasheet]. [https://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual\\_v2-3.pdf](https://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual_v2-3.pdf)
- [12] Aosong Electronics. (n.d.). DHT21 (AM2301) digital temperature and humidity sensor [Datasheet]. <https://mikroshop.ch/pdf/DHT21.pdf>
- [13] Principles of construction of hybrid microsystems for biomedical applications. Dzundza, B.S., Kohut, I.T., Holota, V.I., Turovska, L.V., Deichakivskyi, M.V. – *Physics and Chemistry of Solid State*, 2022, 23 (4), pp.776-784. <https://doi.org/10.15330/pcss.23.4.776-784>



## СИСТЕМА ДИСТАНЦІЙНОГО МОНІТОРИНГУ ПАРАМЕТРІВ МІКРОКЛІМАТУ У ВУЛИКАХ

Юрій Зборівський , Богдан Коман 

Львівський національний університет імені Івана Франка  
вул. Драгоманова, 50, Львів, Україна

### АНОТАЦІЯ

**Вступ.** Здоров'я та продуктивність бджолиних колоній безпосередньо впливають на глобальну продовольчу безпеку, оскільки бджоли сприяють запиленню приблизно 50% рослинних харчових ресурсів у всьому світі. Традиційні методи спостереження за вуликами не мають точності та безперервності, необхідної для ефективного управління колоніями, що створює нагальну потребу в досконалих системах дистанційного моніторингу, здатних безперервно відстежувати критичні параметри навколишнього середовища, включаючи температуру, вологість та якість повітря в екосистемах вуликів.

**Матеріали та методи.** Дане дослідження представляє розробку інтегрованої IoT системи моніторингу, що базується на платформі мікроконтролера ESP32. Система включає комплексний масив датчиків, що складається з газових сенсорів MQ-6 та MQ-135, датчика температури DS18B20, детектора твердих частинок PMS5003 та екологічного сенсора DHT21 для багатопараметричного збору даних. Обробка даних в режимі реального часу досягається завдяки багатозадачній архітектурі на основі FreeRTOS, що забезпечує одночасне опитування сенсорів, перевірку даних та бездротову передачу.

**Результати.** Комплексне тестування системи підтвердило надійний збір даних по всіх контрольованих параметрах. Вимірювання температури продемонстрували робочі діапазони від 13,75°C до 32,69°C для внутрішніх умов вулика, тоді як рівні вологості варіювалися між 51,70% та 88,10%. Моніторинг концентрації газу показав стабільні базові показники, при цьому датчики MQ-6 реєстрували 1,71 ppm за нормальних умов. Статистичний кореляційний аналіз виявив значущі взаємозалежності навколишнього середовища, включаючи сильну позитивну кореляцію ( $r = 0,918269$ ) між вимірюваннями газу MQ-6 та MQ-135 і виражену негативну кореляцію ( $r = -0,974651$ ) між параметрами температури та вологості, підтверджуючи очікувані екологічні взаємозв'язки в мікрокліматі вулика.

**Висновки.** Розроблена IoT-система моніторингу вуликів на базі мікроконтролера ESP32 продемонструвала стабільну та енергоефективну роботу під час безперервних польових випробувань. Система зафіксувала внутрішню температуру вулика в діапазоні від 13,75 °C до 32,69 °C, рівень вологості від 51,70 % до 88,10 %, а також концентрації газів 1,71 ppm (MQ-6) і 0,37 ppm (MQ-135) за нормальних умов.

**Ключові слова:** моніторинг вуликів, IoT, вбудовані системи, параметри мікроклімату, хмарні обчислення, дані реального часу.

UDC: 538.9

## ELECTRONIC PROPERTIES OF Ga-DOPED As-Se-Te GLASSES

Yaroslav Shpotyuk<sup>1</sup><sup>\*</sup>, Adam Ingram<sup>2</sup>, Andriy Luchechko<sup>1</sup>,  
Dmytro Slobodzyan<sup>1</sup>, Markiyan Kushlyk<sup>1</sup>, Oleh Kravets<sup>1</sup>,  
Mykhaylo Shpotyuk<sup>1</sup>, Roman Golovchak<sup>3</sup>

<sup>1</sup>Department of Sensor and Semiconductor Electronics,  
Ivan Franko National University of Lviv,  
107, Tarnavskoho Str., 79017 Lviv, Ukraine

<sup>2</sup>Department of Physics, Opole University of Technology,  
75, Ozimska str., Opole, 45370, Poland

<sup>3</sup>Department of Physics, Engineering and Astronomy, Austin Peay State University,  
Clarksville, TN 37044, USA

\*Corresponding author e-mail: [yaroslav.shpotyuk@lnu.edu.ua](mailto:yaroslav.shpotyuk@lnu.edu.ua)

Shpotyuk, Y., Ingram, A., Luchechko, A., et al. (2025). Electronic Properties of Ga-doped As-Se-Te Glasses. *Electronics and Information Technologies*, 31, 137–144. <https://doi.org/10.30970/eli.31.12>

### ABSTRACT

**Introduction.** The As-Se-Te system attracts significant attention since these chalcogenide glasses are endowed with unique semiconductor and optical characteristics. The wide infrared transparency, non-linearity, and ease of molding capability in these glasses render them promising candidates for electronic and photonic applications. However, the solubility of rare earth ions in chalcogenide matrices is generally low, restricting their applications. The addition of Ga is known to improve apparent solubility, but the intrinsic electrical characteristics of Ga-doped As-Se-Te glasses remain unexplored.

**Materials and Methods.** Chalcogenide glasses of the As-Se-Te system doped with Ga were explored. Samples were obtained from high-purity precursor using the melt-quenching method. Optical, structural, and electronic properties of the studied samples were investigated using high-resolution X-ray photoelectron spectroscopy (XPS), impedance spectroscopy, and optical spectroscopy.

**Results.** The incorporation of both Ga and Te leads to a decrease in the optical bandgap compared to binary As<sub>2</sub>Se<sub>3</sub> glass. The valence band XPS spectra of the studied glasses reveal characteristic features similar to other binary and ternary chalcogenides, reflecting contributions from Se, Te, As, and Ga electronic states. These results indicate that the electronic structure is strongly influenced by chalcogen–As(Ga) bonding, which affects the valence band density of states and associated defect-related phenomena. The temperature-dependent DC conductivity demonstrates multiple conduction mechanisms. Incorporation of Ga lowers the high-temperature activation energy slightly, indicating modifications of the conduction process, while Te-containing samples exhibit even higher activation energies, suggesting contributions from hopping mechanisms and defect-related states.

**Conclusions.** The influence of Ga and Te on the optical and electronic properties of As<sub>2</sub>Se<sub>3</sub>-based chalcogenide glasses is studied using optical and impedance spectroscopies. Electronic properties of Ga-modified As-Se-Te glasses are shown to be important for their applications in optoelectronic integrated platforms. The obtained results are correlated to the valence band structure of these materials determined through XPS.

**Keywords:** chalcogenide glass, DC conductivity, bandgap, temperature dependence.



© 2025 Yaroslav Shpotyuk et al. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

## INTRODUCTION

Arsenic-selenium-tellurium glasses are important functional materials for active and passive applications in modern photonic and electronic integrated platforms thanks to their high infrared (IR) transparency, excellent fiber drawing and molding capabilities, simple thin-film technology, large optical nonlinearities, and presence of semiconductor properties. The common examples include far-IR optics, IR optical waveguides for space telecommunication, chemical and biological sensors, phase-change devices, and Ovonic switches [1-8]. More applications are possible if these glasses are doped with rare-earth elements, such as Ce, Pr, Eu, Tb, Dy, or Er, enabling active media for lasers, optical amplifiers, and broadband sources in the mid-IR spectral range [3,9,10]. The bottleneck of such applications is a concentration of rare-earth ions, whose solubility in chalcogenide matrices is generally low, because at higher concentrations, the rare-earth atoms clamp into metal nanoparticles. Ga is known to improve the number of rare-earth atoms converted into the ionic form, increasing their apparent solubility [11,12].

Recently, the influence of Ga on the structure of arsenic selenides and arsenic-tellurium selenides has been investigated [13,14], whereas the basic electrical properties of these glasses remain unexplored. So, in the present work, we have performed optical and electrical characterization of As-Se-Te glasses containing 2 at.% of Ga, and binary As-Se glasses with a variable concentration of Ga.

## MATERIALS AND METHODS

A conventional melt-quench method was employed for the synthesis of  $\text{Ga}_x(\text{As}_{0.4}\text{Se}_{0.6})_{100-x}$  ( $x=0,1,2,3,4,5$ ) and  $\text{Ga}_2(\text{As}_{0.4}\text{Se}_{0.6})_{98-y}\text{Te}_y$  ( $y=0,10,15,20,30$ ) glasses using high-purity (at least 5N) chemical precursors: Ga, As, Se, and Te. Melting was performed in a vacuum-sealed silica tube at 900 °C for 10 hours using a rocking furnace. After quenching into water from 750 °C, the samples were annealed for 6 h, at ~10 °C below the corresponding glass transition temperatures.

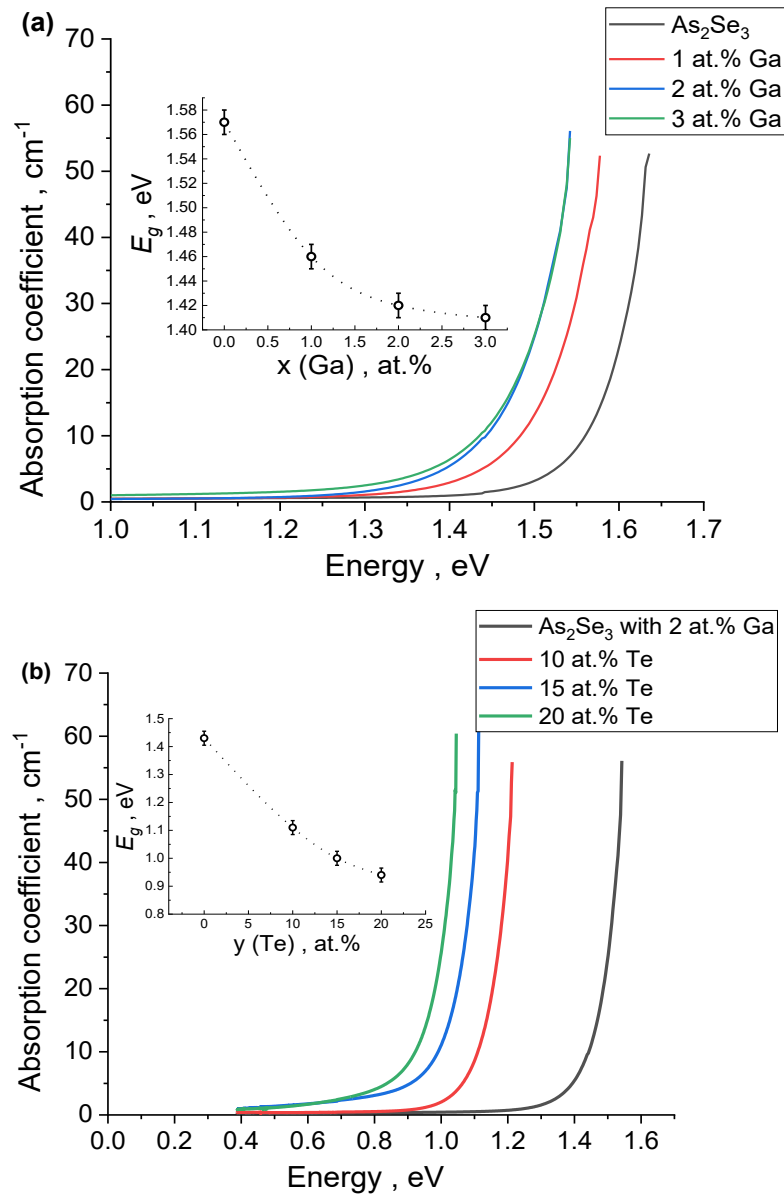
High resolution XPS valence band spectra were recorded with a Scienta ESCA-300 spectrometer (monochromatic Al  $K_\alpha$  X-rays) on the samples fractured *in situ* in the spectrometer's measurement chamber under a vacuum of  $2 \times 10^{-8}$  Torr or better. For all measurements, the angle between the surface and detector was 90°. The instrument was operated in a mode that yielded a Fermi-level width of 0.4 eV for Ag metal and at a full width at half maximum of 0.54 eV for the Ag  $3d_{5/2}$  core level peak. The energy scale was calibrated using the Fermi level of clean Ag. Surface charging from the photoelectron emission was neutralized using a low-energy (<10 eV) electron flood gun. The experimental positions of the valence band spectra were adjusted by referencing to the position of the 1s core level peak (284.6 eV) of adventitious carbon [15]. XPS data were analyzed with the standard CASA-XPS software package.

Temperature-dependent Direct Current (DC) conductivity measurements were performed in vacuum using an HIOKI LCR impedance analyzer with a bias voltage of 1 V. Carbon paste electrodes were deposited on the opposite sides of 1 mm thick sample disks to create an electrical ohmic contact.

Optical transmission spectra were recorded at room temperature by Agilent Technologies Cary-5000 UV/Vis/NIR spectrometer in the 200–3200 nm spectral range with 2 nm resolution.

## RESULTS AND DISCUSSION

The fundamental optical absorption edges of the investigated  $\text{Ga}_x(\text{As}_{0.4}\text{Se}_{0.6})_{100-x}$  and  $\text{Ga}_2(\text{As}_{0.4}\text{Se}_{0.6})_{98-y}\text{Te}_y$  glasses are shown in **Fig. 1**, where the absorption coefficient ( $\alpha$ ) was calculated from the transmission data of bulk 2 mm thick samples with the aid of PARAV software [16].



**Fig. 1.** Optical absorption coefficients of  $\text{Ga}_x(\text{As}_{0.4}\text{Se}_{0.6})_{100-x}$  (a) and  $\text{Ga}_2(\text{As}_{0.4}\text{Se}_{0.6})_{98-y}\text{Te}_y$  (b) bulk glasses calculated from room-temperature transmission spectra. The inserts show estimated optical bandgap compositional dependences.

The edge follows the expected Urbach exponential behaviour proper to most chalcogenide glasses [17,18]. The inserts show optical bandgap ( $E_g$ ) values and their compositional dependence, estimated using Tauc plots for the indirect electron transitions in PARAV [16].

$$\alpha h\nu = B(h\nu - E_g)^2 \quad (1)$$

The room-temperature optical bandgap values drop from  $\sim 1.6 (\pm 0.05)$  eV for  $\text{As}_2\text{Se}_3$  glass to  $\sim 1.4 (\pm 0.1)$  eV for Ga-modified  $\text{As}_2\text{Se}_3$ , and to  $\sim 1.0 (\pm 0.1)$  eV with Te addition. So,

incorporation of both Ga and Te leads to a decrease in the optical bandgap compared to binary  $\text{As}_2\text{Se}_3$  glass, the effect of Te being more significant.

The classic DC conductivity ( $\sigma_{DC}$ ) vs  $1/T$  dependence of chalcogenide glasses shows several distinct regions [17-19]. The high-temperature one is associated with charge carriers excited directly into the non-localized states of the conduction band ( $E_c$ ). It is followed by the region where charge carriers are excited into localized states ( $E_A$ ) near the edge of the conduction band or localized states ( $E_B$ ) near the edge of the valence band ( $E_v$ ). They usually contribute to a hopping mechanism of charge transport. There could also be a region associated with temperature-activated hopping of charge carriers localized in the states at the Fermi level ( $E_F$ ), if present. All these mechanisms follow  $\exp(-E_a/kT)$  dependence, where activation energy  $E_a$  depends on  $E_A$ ,  $E_B$ ,  $E_c$ ,  $E_v$ ,  $E_F$  and activation energies of charge hopping ( $W_i$ ) between relevant localized states [18]. Finally, the low-temperature region usually follows the well-known  $T^{-1/4}$  law and is ascribed to a temperature-activated hopping of charge carriers with variable jump lengths (variable range hopping) [18].

The slopes of the linear fits to the high-temperature region of  $\ln(\sigma_{DC})$  vs  $1/T$  dependences (Fig. 2) were used to determine the relevant activation energies. Calculated high-temperature value of  $E_a = 0.85 (\pm 0.05)$  eV for  $\text{As}_2\text{Se}_3$  bulk sample roughly corresponds to a half of the optical bandgap (or slightly higher), which allows us to assume (if one accepts that Fermi level in chalcogenide glasses is pinned near the middle of bandgap) [17,18] that this conduction mechanism is associated with direct excitation of charge carriers into the non-localized states of conduction band with  $E_a = E_c - E_F$ . Incorporation of Ga into  $\text{As}_2\text{Se}_3$  bulk glass decreases the value of  $E_a = 0.75 (\pm 0.02)$  eV while still being close to half of the corresponding optical bandgap of Ga-modified  $\text{As}_2\text{Se}_3$  samples. Nevertheless, these values are consistently higher than one half of the corresponding optical bandgap values, which might indicate a slight shift of the Fermi level towards the valence band in full agreement with the fact that most chalcogenides are generally considered as *p*-type semiconductors [17,20]. This difference is even more pronounced in Te-containing samples, where doubled  $E_a$  values ( $E_a$  decreases from 0.67 to 0.60 eV with Te addition) are noticeably higher than  $E_g$  values ( $1.0 \pm 0.1$  eV, see insert to Fig. 1b). The higher activation energies than the half of the optical bandgap values can also originate from the hopping mechanism of conduction involving localized states with  $E_a = E_{A,B} - E_F + W_i$ , where  $W_i$  is the hopping activation energy [18]. The latter mechanism can also be a reason for a kink in conductivity dependences like the one visible after  $1000/T \sim 3$  for some of the investigated glass compositions, including  $\text{As}_2\text{Se}_3$  (Fig. 2). The hopping mechanism relies on a higher concentration of defects, like  $\text{D}^+ \cdot \text{D}^-$  topological

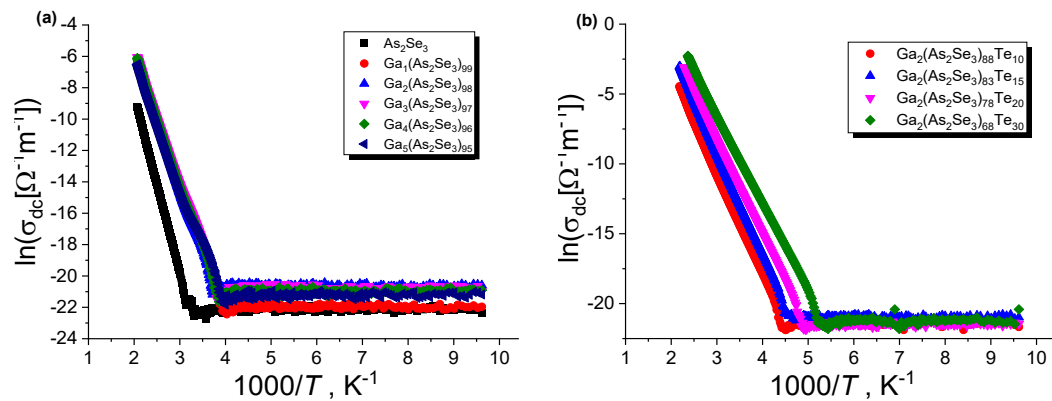
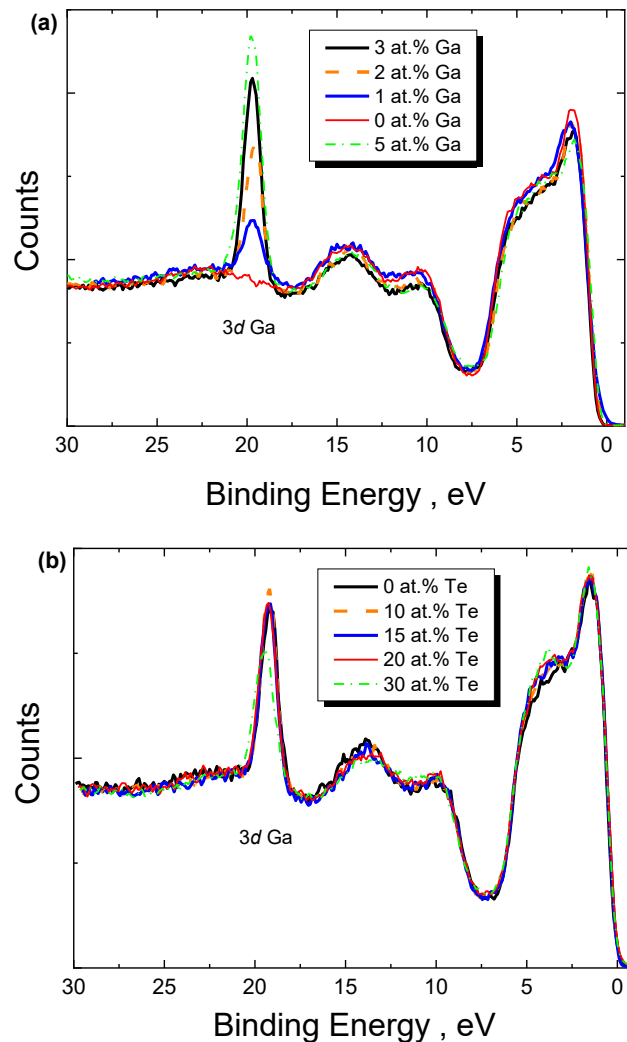


Fig. 2. Temperature dependence of DC conductivity for  $\text{Ga}_x(\text{As}_{0.4}\text{Se}_{0.6})_{100-x}$  (a) and  $\text{Ga}_2(\text{As}_{0.4}\text{Se}_{0.6})_{98-y}\text{Te}_y$  (b).

coordination centres, used to explain many phenomena in vitreous chalcogenides [17,18]. At the same time, the low-temperature region of  $T^{-1/4}$  dependence is difficult to assess in the samples investigated due to very low conductivity values, which are on the verge of equipment sensitivity.

The valence band density of states is crucial for the understanding of electronic properties, defect structure, and various induced phenomena occurring in chalcogenide glasses [17-21]. Valence band XPS spectra of the investigated samples (Fig. 3) show similar features to the valence bands of other binary and ternary chalcogenides [22]. The well-observed feature at about 20 eV is attributed to the lone pair Se 4p and Te 5p (in Te-containing samples) electrons, whereas the peak at about 5 eV is contributed by 4p and 5p bonding states of Se and Te (if present in the composition), respectively. The observed broad band at 7-16 eV is due to the overlap of signals from Ga 4s, As 4s, Se 4s and Te 5s electrons, whereas XPS signal at 18 eV is caused by Ga 3d electrons (Fig. 3). The valley at ~3 eV, which is well observed in Se-rich arsenic selenides [22], disappears in the investigated glasses due to the broadening of Se 4p and Te 5p XPS peaks by As 4p and Ga 4p bonding states from a prevailing concentration of chalcogen-As(Ga) bonds.



**Fig. 3.** Valence band XPS spectra recorded for a fractured in vacuum  $\text{Ga}_x(\text{As}_{0.4}\text{Se}_{0.6})_{100-x}$  (a) and  $\text{Ga}_2(\text{As}_{0.4}\text{Se}_{0.6})_{98-y}\text{Te}_y$  (b) bulk glasses.



## CONCLUSION

Incorporation of Ga into As<sub>2</sub>Se<sub>3</sub> glass matrix leads to a decrease in its optical bandgap and activation energy of high-temperature DC conductivity. Further addition of Te into the composition leads to a more significant decrease in the optical bandgap and activation energy of DC electrical conductivity. Valence band XPS spectra correlate well with the electrical and optical properties of the investigated materials, showing typical features of chalcogenide vitreous semiconductors.

## ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was performed within the framework of the IMPRESS-U project supported by the NSF (Grant # OISE-2106457), NAWA (Grant # BPN/NSF/2023/1/00002/U/00001), and NAS (STCU Grant # 7112). This work is part of the research performed within projects No. 0123U100655 and 0125U000544, the subject of the Program funded by the Ministry of Education and Science of Ukraine.

## COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any potential conflict of interest.

## AUTHOR CONTRIBUTIONS

Conceptualization, [Y.S., A.I., D.S., R.G.]; methodology, [Y.S., A.L., M.S., R.G.]; validation, [Y.S., A.I., R.G.]; formal analysis, [D.S., M.K.]; investigation, [Y.S., A.I., O.K., R.G.]; resources, [Y.S., M.S., R.G.]; data curation, [Y.S., A.I., D.S., R.G.]; writing – original draft preparation, [Y.S., R.G.]; writing – review and editing, [A.L., M.K., M.S.]; visualization, [Y.S., O.K., R.G.]; supervision, [Y.S., R.G.]; project administration, [Y.S., R.G.].

## REFERENCES

- [1] Seddon, A.B., Farries, M.C., Nunes J.J., Xiao, B., Furniss D., Barney, E., Phang, S., Chahal, S., Kalfagiannis, N., Sojka, Ł. & Sujecki, S. (2024). Short review and prospective: chalcogenide glass mid-infrared fibre lasers. *Eur. Phys. J. Plus*, 139, 142. <https://doi.org/10.1140/epjp/s13360-023-04841-1>
- [2] Musgraves, J.D. (2024) Chalcogenide glasses: Engineering in the infrared spectrum. *American Ceramic Society Bulletin*, 103(4), 22. <https://bulletin.ceramics.org/article/chalcogenide-glasses-engineering-in-the-infrared-spectrum/>
- [3] Adam, J-L. & Zhang X. (2014). Chalcogenide Glasses: Preparation, properties and application. *Oxford, Cambridge: Woodhead Publishing series in Electronic and Optical Materials*, 704. ISBN 0857093568, 9780857093561
- [4] Golovchak, R., Plummer, J., Kovalskiy, A., Holovchak, Y., Ignatova, T., Trofe, A., Mahlovanyi, B., Cebulski, J., Krzeminski, P., Shpotyuk, Y., Boussard-Pledel, C. & Bureau, B. (2023). Phase-change materials based on amorphous equichalcogenides. *Scientific Reports*, 13, 2881. <https://doi.org/10.1038/s41598-023-30160-7>
- [5] Fritzsche, H. (2007). Why are chalcogenide glasses the materials of choice for Ovonic switching devices? *Journal of Physics and Chemistry of Solids*, 68, 878–882. <https://doi.org/10.1016/j.jpcs.2007.01.017>
- [6] Cui, S., Chahala, R., Shpotyuk, Y., Boussard, C., Lucas, J., Charpentier, F., Tariel, H., Loréal, O., Nazabal, V., Sire, O., Monbet, V., Yang, Z., Lucas, P. & Bureau, B. (2014). Selenide and telluride glasses for mid-infrared bio-sensing. *Proceedings of SPIE*, 8938, 893805. <https://doi.org/10.1117/12.2036734>
- [7] Eggleton, B.J., Luther-Davies, B. & Richardson, K. (2011). Chalcogenide photonics. *Nature Photonics*, 5, 141–148. <https://doi.org/10.1038/nphoton.2011.309>

- [8] Yu, Q., Wang, Sh., Fu, Y., Tan, L., Gao, Ch., Lin, Ch. & Kang, Sh. (2024). Highly Er<sup>3+</sup>-doped chalcogenide glass with enhanced mid-infrared emission and superior mechanical property. *Optics Express*, 32(26), 46131-46139. <https://doi.org/10.1364/OE.544914>
- [9] Blanc, W., Choi, Y.G., Zhang, X., Nalin, M., Richardson, K.A., Righini, G.C., Ferrari, M., Jha, A., Massera, J., Jiang, S., Ballato, J. & Petit, L. (2023). The past, present and future of photonic glasses: A review in homage to the United Nations International Year of Glass 2022. *Progress in Materials Science*, 134, 101084. <https://doi.org/10.1016/j.pmatsci.2023.101084>
- [10] Seznec, V., Ma, H., Zhang, X., Nazabal, V., Adam, J.-L., Qiao, X.S. & Fan, X.P. (2006). Spectroscopic properties of Er<sup>3+</sup>-doped chalcogenide glass ceramics. *Proceedings of SPIE*, 6116, 61160B–1–9. <https://doi.org/10.1117/12.645909>
- [11] Aitken, B.G., Ponader, C.W., & Quimby, R.S. (2002). Clustering of rare earths in Ge–As sulfide glass. *Comptes Rendus Chimie*, 5(12), 865-872. [https://doi.org/10.1016/S1631-0748\(02\)01458-3](https://doi.org/10.1016/S1631-0748(02)01458-3)
- [12] Choi, Y.G., Kim, K.H., Lee, B.J., Shin, Y.B., Kim, Y.S. & Heo, J. (2000). Emission properties of the Er<sup>3+</sup>:<sup>4</sup>I<sub>11/2</sub>→<sup>4</sup>I<sub>13/2</sub> transition in Er<sup>3+</sup>- and Er<sup>3+</sup>/Tm<sup>3+</sup>-doped Ge–Ga–As–S glasses. *Journal of Non-Crystalline Solids*, 278, 137-144. [https://doi.org/10.1016/S0022-3093\(00\)00331-8](https://doi.org/10.1016/S0022-3093(00)00331-8)
- [13] Shpotyuk, Y., Bureau, B., Boussard-Pledel, C., Nazabal, V., Golovchak, R., Demchenko, P. & Polovynko, I. (2014). Effect of Ga incorporation in the As<sub>30</sub>Se<sub>50</sub>Te<sub>20</sub> glass. *Journal of Non-Crystalline Solids*, 398–399, 19–25. <https://doi.org/10.1016/j.jnoncrysol.2014.04.021>
- [14] Shpotyuk, Y., Boussard-Pledel, C., Nazabal, V., Chahal, R., Ari, J., Pavlyk, B., Cebulski, J., Doualan, J.L. & Bureau, B. (2015). Ga-modified As<sub>2</sub>Se<sub>3</sub>–Te glasses for active applications in IR photonics. *Optical Materials*, 46, 228–232. <https://doi.org/10.1016/j.optmat.2015.04.024>
- [15] Moulder, J.F., Stickler, W.F., Sobol, P.E. & Bomben, K.D. (1992). *Handbook of X-ray photoelectron spectroscopy* (J. Chastein, Ed.). Eden Prairie, MN: Perkin-Elmer Corp., Physical Electronics Division. Retrieved from <https://www.phl.com/>
- [16] Ganjoo, A. & Golovchak, R. (2008). Computer program PARAV for calculating optical constants of thin films and bulk materials: Case study of amorphous semiconductors. *Journal of Optoelectronics and Advanced Materials*, 10, 1328–1332. Retrieved from <http://joam.inoe.ro/>
- [17] Feltz, A. (1993). *Amorphous inorganic materials and glasses*. Weinheim, Germany: VCH, 477. ISBN 3527284214, 9783527284214.
- [18] Mott, N.F. & Davis, E.A. (1979). *Electronic processes in non-crystalline materials*. Oxford, England: Clarendon Press.
- [19] Golovchak, R., Plummer, J., Kovalskiy, A., Ingram, A., Kozdras, A., Ignatova, T., Trofe, A., Boussard-Pledel, C. & Bureau, B. (2024). Effect of Bi additive on the physical properties of Ge<sub>2</sub>Se<sub>3</sub>-based equichalcogenide glasses and thin films. *ACS Applied Electronic Materials*, 6(4), 2720–2727. <https://doi.org/10.1021/acsaelm.4c00263>
- [20] Popescu, M. (2000). *Non-crystalline chalcogenides*. New York, NY: Kluwer Academic Publishers. <https://doi.org/10.1007/0-306-47129-9>
- [21] Golovchak, R., Plummer, J., Kovalskiy, A., Holovchak, Y., Ignatova, T., Nowlin, K., Trofe, A., Shpotyuk, Y., Boussard-Pledel, C. & Bureau, B. (2022). Broadband photosensitive medium based on amorphous equichalcogenides. *ACS Applied Electronic Materials*, 4(11), 5397–5405. <https://doi.org/10.1021/acsaelm.2c01075>
- [22] Golovchak, R., Shpotyuk, O. & Kovalskiy, A. (2023). High-resolution XPS for determining the chemical order in chalcogenide network glasses. *Journal of Non-Crystalline Solids: X*, 18, 100188. <https://doi.org/10.1016/j.nocx.2023.100188>

## ЕЛЕКТРОННІ ВЛАСТИВОСТІ СКЛА As-Se-Te, ЛЕГОВАНОГО Ga

Ярослав Шпотюк<sup>1,\*</sup>, Адам Інграм<sup>2</sup>, Андрій Лучечко<sup>1</sup>,  
Дмитро Слободзян<sup>1</sup>, Маркіян Кушлик<sup>1</sup>, Олег Кравець<sup>1</sup>,  
Михайло Шпотюк<sup>1</sup>, Роман Головчак<sup>3</sup>

<sup>1</sup>Кафедра сенсорної та напівпровідникової електроніки,  
Львівський національний університет імені Івана Франка,  
вул. Тарнавського, 107, м. Львів, 79017, Україна

<sup>2</sup>Кафедра фізики, Політехніка Опольська,  
вул. Озімська, 75, м. Ополе, 45370, Польща

<sup>3</sup>Кафедра фізики, інженерії та астрономії, Державний університет Остіна Пі,  
м. Кларксвілл, TN 37044, США

## АНОТАЦІЯ

**Вступ.** Система As-Se-Te привертає значну увагу, оскільки ці сорти халькогенідного скла наділені унікальними напівпровідниковими та оптичними властивостями. Широка прозорість в інфрачервоному діапазоні, нелінійність та простота формування роблять ці матеріали перспективними кандидатами для електронних і фотонних застосувань. Однак розчинність рідкісноземельних іонів у халькогенідних матрицях загалом є низькою, що обмежує їх практичне використання. Додавання Ga, як відомо, покращує розчинність рідкісноземельних іонів, проте власні електричні характеристики скла As-Se-Te, легованого Ga, залишаються переважно недослідженими.

**Матеріали та методи.** Досліджено композиції халькогенідного скла системи As-Se-Te леговані Ga. Зразки були отримані з високочистих прекурсорів методом плавлення з подальшим охолодженням. Оптичні, структурні та електронні властивості досліджених зразків були вивчені за допомогою високороздільної X-променевої фотоелектронної спектроскопії (XPS), імпедансної спектроскопії та оптичної спектроскопії.

**Результати.** Введення як Ga, так і Te призводить до зменшення ширини забороненої зони порівняно з бінарним склом As<sub>2</sub>Se<sub>3</sub>. Спектри валентної зони, отримані методом рентгенівської фотоелектронної спектроскопії для досліджених матеріалів демонструють характерні особливості, схожі на інші бінарні та тернарні халькогеніди, що відображає внесок електронних станів Se, Te, As та Ga. Ці результати вказують на те, що електронна структура сильно визначається зв'язками халькоген–As(Ga), що впливає на густину станів валентної зони та пов'язані з дефектами. Температурно-залежна електропровідність постійного струму демонструє кілька механізмів провідності. Введення Ga знижує енергію активації при високих температурах через модифікацію процесу провідності, тоді як зразки, що містять Te, демонструють ще вищі значення енергії активації, що вказує на внесок механізмів стрибкоподібної провідності та дефектних станів.

**Висновки.** Вплив Ga та Te на оптичні та електронні властивості халькогенідного скла на основі As<sub>2</sub>Se<sub>3</sub> досліджувався за допомогою оптичної та імпедансної спектроскопії. Показано, що електронні властивості композицій скла As-Se-Te, модифікованого Ga мають важливе значення для їх потенційного використання в інтегрованих оптоелектронних платформах. Отримані результати корелюють зі структурою валентної зони цих матеріалів, визначеною за допомогою рентгенівської фотоелектронної спектроскопії.

**Ключові слова:** халькогенідне скло, електропровідність постійного струму, заборонена зона, температурна залежність.

Збірник наукових праць

## **Електроніка та інформаційні технології**

## **Electronics and information technologies**

Випуск 31

2025

Підп. до друку 31.10.2025. Формат 70x100,16. Папір друк.  
Друк на різогр. Гарнітура Times New Roman. Умовн. друк. арк. .  
Тираж 100 прим. Зам. № .

Львівський національний університет імені Івана Франка.  
79000 Львів, вул. Університетська, 1.

Свідоцтво про внесення суб'єкта видавничої справи до Державного  
реєстру видавців, виготівників і розповсюджувачів видавничої  
продукції. Серія ДК № 3059 від 13.12.2007 р.