



University of Maribor Press

Artificial Intelligence and Environmental Challenges

Research Insights and Emerging Solutions

edited by **Robert Leskovar**



University of Maribor

Faculty of Organizational Sciences

Artificial Intelligence and Environmental Challenges

Research Insights and Emerging Solutions

Editor

Robert Leskovar

June 2026

Title <i>Naslov</i>	Artificial Intelligence and Environmental Challenges <i>Umetna inteligenca in okoljski izzivi</i>
Subtitle <i>Podnaslov</i>	Research Insights and Emerging Solutions <i>Raziskovalni vpogledi in nastajajoče rešitve</i>
Editor <i>Urednik</i>	Robert Leskovar (University of Maribor, Faculty of Organizational Sciences)
Review <i>Recenzija</i>	Björn Paape (RWTH Aachen) Blaž Rodič (Faculty of Information Studies)
Language editing <i>Jezikovni pregled</i>	Meta Arnež
Technical editors <i>Tehnična urednika</i>	Robert Leskovar (University of Maribor, Faculty of Organizational Sciences) Jan Perša (University of Maribor, University of Maribor Press)
Cover designer <i>Oblikovanje ovitka</i>	Jan Perša (University of Maribor, University of Maribor Press)
Cover graphic <i>Grafika na ovitku</i>	Tokyo, photo: LoboStudio Hamburg, uinsplash.com, 2024
Graphic material <i>Grafične priloge</i>	Sources are own unless otherwise noted. Leskovar (editor), 2026
Published by <i>Založnik</i>	University of Maribor University of Maribor Press Slomškovo trg 15, 2000 Maribor, Slovenia https://press.um.si , zalozba@um.si
Issued by <i>Izdajatelj</i>	University of Maribor Faculty of Organizational Sciences Kidričeva cesta 55a, 4000 Kranj, Slovenia https://www.fov.um.si , dekanat.fov@um.si
Edition <i>Izdaja</i>	First edition
Publication type <i>Vrsta publikacije</i>	E-book
Available at <i>Dostopno na</i>	http://press.um.si/index.php/ump/catalog/book/1132
Published at <i>Izdano</i>	Maribor, Slovenia, June 2026
Project name <i>Ime projekta</i>	Artificial Intelligence to Solve Environmental Problems (AI2SEP)
Project number <i>Številka projekta</i>	2023-1-PL01-KA220-HED-000166765
Project financier <i>Financer projekta</i>	European Union



© University of Maribor, University of Maribor Press
/ Univerza v Mariboru, Univerzitetna založba

Text / Besedilo © Authors and Leskovar (editor), 2026

This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. / *To delo je objavljeno pod licenco Creative Commons Priznanje avtorstva-Nekomercialno 4.0 Mednarodna.*

This license allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator. The license allows for commercial use.

Any third-party material in this book is published under the book's Creative Commons licence unless indicated otherwise in the credit line to the material. If you would like to reuse any third-party material not covered by the book's Creative Commons licence, you will need to obtain permission directly from the copyright holder.

<https://creativecommons.org/licenses/by/4.0/>

CIP - Kataložni zapis o publikaciji
Univerzitetna knjižnica Maribor

004.8:502.1(082)(0.034.2)

ARTIFICIAL intelligence and environmental challenges [Elektronski vir] : research insights and emerging solutions / editor Robert Leskovar. - 1st ed. - E-zbornik. - Maribor : University of Maribor, University of Maribor Press, 2026

Način dostopa (URL): <https://press.um.si/index.php/ump/catalog/book/1132>

ISBN 978-961-299-160-9 (PDF)

doi: 10.18690/um.fov.5.2026

COBISS.SI-ID 281943555

ISBN 978-961-299-160-9 (pdf)

DOI <https://doi.org/10.18690/um.fov.5.2026>

Price
Cena Free copy

For publisher Prof. dr. Zdravko Kačič,
Odgovorna oseba založnika rector of University of Maribor

Attribution Leskovar, R. (Ed.). (2026). *Artificial Intelligence and Environmental Challenges: Research*
Citiranje *Insights and Emerging Solutions*. University of Maribor, University of Maribor Press.
doi: 10.18690/um.fov.5.2026



PROFESSOR ANDRZEJ GRZYBOWSKI

This scientific monograph is dedicated to our dear colleague, Professor Andrzej Grzybowski, at the Institute of Physics, University of Silesia. Andrzej gathered five universities and initiated the project "Developing Talents in Artificial Intelligence to Solve Disruptive Environmental Problems (AI2SEP)". He was determined and steadfastly committed to teamwork. Andrzej did not live to see this volume completed, but his influence pervades it. His generosity, enthusiasm, and collaborative ethos continue to inspire us.



University of Maribor Press

Preface

The volume is born of a conviction shared by researchers from five universities across five countries: that artificial intelligence is no longer merely a tool of technological convenience, but an indispensable partner in confronting the environmental crises of our era. Wind energy, solar reliability, air quality, green finance, nanoplastic contamination, and urban heat — each of these challenges demands computational power and analytical sophistication that AI uniquely provides. It is to these intersections that the present monograph is devoted.

The chapters collected here grew out of the collaborative project *Developing Talents in Artificial Intelligence to Solve Disruptive Environmental Problems* (AI2SEP), funded by the European Commission (No. 2023-1-PL01-KA220-HED-000166765) and carried forward by Pavol Jozef Šafárik University in Košice, the University of Silesia, the University of Zagreb Faculty of Organization and Informatics, Vilnius University, and the University of Maribor Faculty of Organizational Sciences. The breadth of that partnership is reflected in the diversity of methodologies represented here, from classical tree-based ensembles to molecular dynamics simulations, from deep learning architectures to stochastic volatility models.

The book is organised around a single guiding question: *Where can AI provide genuine insight into environmental problems, and at what cost?* Each chapter answers that question in its own domain.

The opening contribution, by **Manohar Gowdru Shridhara, Ľubomír Antoni, and Gabriel Semanišin**, benchmarks tree-based machine learning methods — decision trees, random forests, gradient boosting variants (CatBoost, LightGBM), and fuzzy ensembles — on the US Wind Turbine Database. The chapter provides both a technically rigorous treatment of the underlying mathematics and a practically grounded evaluation of predictive performance for wind energy output, offering a valuable reference for practitioners in renewable energy data analysis.

Joachim J. Włodarz then turns to a frontier of hardware-aware AI: the deployment of small language models (SLMs) on edge devices. As the computational footprint of large AI models becomes itself an environmental concern, the design of efficient, resource-constrained inference pipelines grows in importance. This chapter surveys model compression techniques, hardware platforms from single-board computers

to microcontrollers, and the emerging landscape of edge AI, providing a map of possibilities for researchers who must balance capability against energy consumption.

The third chapter, by **Dijana Oreški, Vjeran Strahonja, and Marija Pokos Lukinec**, applies supervised machine learning and deep learning to the prediction of renewable energy output. The authors pay careful attention to hyperparameter optimisation and to the economic and environmental sustainability implications of their findings, demonstrating that improved forecasting translates directly into reduced grid instability and lower carbon emissions.

Eugenijus Gabrielius Ivanauskas, Liepa Urbonaitė, and Saulius Jokubaitis address the financial architecture of the green transition. Their chapter applies sophisticated econometric tools — GJR-GARCH, component GARCH, eGARCH, and vine copula structures — to model the tail risks and return dynamics of green asset classes. As private capital flows are increasingly recognised as essential to meeting climate targets, rigorous quantification of risk in sustainable finance becomes a matter of public as well as scholarly concern.

Fault detection in solar power plants is the subject of the fifth chapter, contributed by **Dominykas Vilčinskas, Lukas Voveris, and Jolita Bernatavičienė**. Drawing on energy production time series, the authors develop outlier-detection and fault-profile methods capable of identifying degraded panel performance before it escalates into costly system failures. The practical value of reliable, automated monitoring for large photovoltaic installations is difficult to overstate.

Ivan Maslov, Agnieszka Głowacka, Bartosz Dziewit, and Paulina Trybek contribute a meticulous review of data challenges in machine learning for air quality and CO₂ emissions. The chapter surveys publicly available datasets, analyses the statistical characteristics of particulate matter (PM_{2.5}, PM₁₀) and carbon dioxide concentrations, and presents illustrative prediction experiments. Its central message — that understanding one's data is a prerequisite for trustworthy modelling — is as simple as it is easily overlooked.

The penultimate chapter moves to the molecular scale. **Jakub Kraus, Piotr Mika, Mateusz Wiśniewski, Paulina Trybek, and Zbigniew Dendzik** explore molecular dynamics (MD) simulations as a tool for investigating how nanoplastics interact with proteins, lipid bilayers, and other biomolecules. As nanoplastic contamination emerges as one of the more insidious environmental threats of our time — pervasive, invisible, and incompletely understood — computational simulation offers a route to mechanistic understanding that laboratory experiment alone cannot yet provide.

The closing chapter is my own contribution on the modelling of urban heat transfer using three-dimensional cellular automata coupled with AI-assisted code development. Urban heat islands represent a measurable, growing public health risk. The chapter describes a physically grounded simulation framework and reflects frankly on the role that human–AI collaboration played in its construction, a methodological transparency that I hope will encourage similar candour elsewhere in the field.

Read together, these chapters reveal both the promise and the limitations of the current moment. AI methods — whether ensemble learners, deep networks, or language models — deliver impressive results when high-quality, domain-appropriate data are available and when the underlying physical or financial mechanisms are at least par-

tially understood. They falter when data are sparse, biased, or poorly curated, a recurrent theme across several chapters. The reader will find no triumphalism here, only honest assessment.

Equally recurring is the question of interpretability. Environmental decision-making is ultimately a human and political process; models whose outputs cannot be explained to non-specialists risk either being ignored or, worse, being accepted uncritically. The preference for tree-based and fuzzy methods in early chapters, and the careful attention to physical plausibility in the cellular automata chapter, reflect a shared commitment to models that can be understood and questioned.

In the role of editor, I wish to thank all contributing authors for their research efforts and patience, the reviewers whose careful reading improved every chapter, and the staff of University Press University of Maribor for their support throughout the production process.

It remains only to invite the reader to engage with this work critically, to test its findings against new data, to extend its methods to new problems, and — above all — to remember that behind every algorithm lies a real environment that does not forgive errors of understanding.

Robert Leskovar
Faculty of Organizational Sciences
University of Maribor



University of Maribor Press

Acknowledgement

This volume gathered researchers from:

- Pavol Jozef Šafárik University in Košice,
- University of Silesia,
- University of Zagreb Faculty of Organization and Informatics,
- Vilnius University, and
- University of Maribor Faculty of Organizational Sciences.

Our institutions created great opportunities for research and publishing. Authors gratefully acknowledge the support and partial funding by:

- VVGS ESGD grant (Early Stage Grants – Pavol Jozef Šafárik University in Košice), led by Ing. Manohar Gowdru Shridhara, ŠPP element OHV040126 / FS 190180 / 2887 – VVGS ESGD – Gowdru, Faculty of Science, within the project “Early Stage Grants – Pavol Jozef Šafárik University in Košice”, code 09I03-03-V05-00008. (M. Gowdru Shridhara)
- European Commission grant AI2SEP project (No. 2023-1-PL01-KA220-HED-000166765)
- Slovenian Research and Innovation Agency, research core funding No. P5-0018

We acknowledge *Oracle Academy* as an associate partner which played an important role in forming the project consortia, promoting and disseminating project results. *Our personal thanks to Mr. Darko Jurekovič, Senior Marketing Program Manager, Oracle Academy (EMEA)* for his continued support of the AI2SEP project.

This publication does not constitute the founders’ endorsement of the content. The content reflects only the views of the authors, and none of the founders can be held responsible for any use which may be made of the information contained therein.

Robert Leskovar on behalf of all authors



University of Maribor Press

List of Contributors

Lubomír Antoni

Faculty of Science, Jesenná 5, Pavol Jozef Šafárik University in Košice, 04001 Košice, Slovakia, e-mail: lubomir.antoni@upjs.sk

Jolita Bernatavičienė

Institute of Data Science and Digital Technologies, Vilnius University, Akademijos str. 4, Vilnius, Lithuania, e-mail: jolita.bernatavicienne@mif.vu.lt

Zbigniew Dendzik

Institute of Physics, University of Silesia, 40-007 Katowice, Poland, e-mail: zbigniew.dendzik@us.edu.pl

Bartosz Dziewit

Institute of Physics, University of Silesia, 40-007 Katowice, Poland, e-mail: bartosz.dziewit@us.edu.pl

Agnieszka Głowacka

Faculty of Science and Technology, University of Silesia, 40-007 Katowice, Poland, e-mail: agnieszka.glowacka@us.edu.pl

Manohar Gowdru Shridhara

Faculty of Science, Pavol Jozef Šafárik University in Košice, Jesenná 5, 04001 Košice, Slovakia, e-mail: manohar.gowdru.shridhara@student.upjs.sk

Eugenijus Gabrielius Ivanauskas

Institute of Applied Mathematics, Vilnius University, Naugarduko str. 24, LT-03225, Vilnius, Lithuania, e-mail: gabrielius.ivanauskas@mif.stud.vu.lt

Saulius Jokubaitis

Institute of Applied Mathematics, Vilnius University, Naugarduko str. 24, LT-03225, Vilnius, Lithuania, e-mail: saulius.jokubaitis@mif.vu.lt

Jakub Kraus

Faculty of Science and Technology, University of Silesia, 40-007 Katowice, Poland, e-mail: jakub.kraus@us.edu.pl

Robert Leskovar

Faculty of Organizational Sciences, University of Maribor, Kidričeva 55a, 4000 Kranj, Slovenia, e-mail: robert.leskovar@um.si

Ivan Maslov

Faculty of Science and Technology, University of Silesia, 40-007 Katowice, Poland, e-mail: ivan.maslov@us.edu.pl

Piotr Mika

Faculty of Science and Technology, University of Silesia, 40-007 Katowice, Poland, e-mail: piotr.mika@us.edu.pl

Dijana Oreški

Faculty of Organization and Informatics, University of Zagreb, Pavlinska 2, 42000 Varaždin, Croatia, e-mail: dijana.oreski@foi.unizg.hr

Marija Pokos Lukinec

Faculty of Organization and Informatics, University of Zagreb, Pavlinska 2, 42000 Varaždin, Croatia, e-mail: mapokos@foi.hr

Gabriel Semanišin

Faculty of Science, Pavol Jozef Šafárik University in Košice, Jesenná 5, 04001 Košice, Slovakia, e-mail: gabriel.semanisin@upjs.sk

Vjeran Strahonja

Faculty of Organization and Informatics, University of Zagreb, Pavlinska 2, 42000 Varaždin, Croatia, e-mail: vjeran.strahonja@foi.unizg.hr

Paulina Trybek

Institute of Physics, University of Silesia, 40-007 Katowice, Poland, e-mail: paulina.trybek@us.edu.pl

Liepa Urbonaitė

Institute of Applied Mathematics, Vilnius University, Naugarduko str. 24, LT-03225, Vilnius, Lithuania, e-mail: liepa.urbonaite@mif.stud.vu.lt

Dominykas Vilčinskas

Institute of Applied Mathematics, Vilnius University, Naugarduko str. 24, LT-03225, Vilnius, Lithuania, e-mail: dominykas.vilcinskas@mif.stud.vu.lt

Lukas Voveris

Institute of Applied Mathematics, Vilnius University, Naugarduko str. 24, LT-03225, Vilnius, Lithuania, e-mail: lukas.voveris@mif.stud.vu.lt

Mateusz Wiśniewski

Institute of Physics, University of Silesia, 40-007 Katowice, Poland, e-mail: mateusz.wisniewski@us.edu.pl

Joachim J. Włodarz

Faculty of Science and Technology, University of Silesia, Bankowa 14, 40-007 Katowice, Poland, e-mail: joachim.wlodarz@us.edu.pl

Contents

Preface	7
Acknowledgement	11
List of Contributors	13
Tree-based Machine Learning Methods for Wind Farm Data	1
Manohar Gowdru Shridhara, Eubomír Antoni, and Gabriel Semanišin	
1 Introduction	2
2 Learning methods of decision trees	3
2.1 Decision trees for regression	3
2.2 Bagging and random forests	4
2.3 Boosting methods	6
2.4 Fuzzy decision trees and fuzzy ensembles	7
3 Dataset description	9
3.1 US Wind turbine database	10
3.2 Data attributes	10
3.3 Data preprocessing	11
4 Methodology	12
4.1 Regression task definition	12
4.2 Implemented models	13
4.3 Evaluation metrics	14
5 Results and discussion	15
5.1 Quantitative results	15
5.2 Model comparison	16
5.3 Feature importance analysis	16
6 Conclusion and future work	17
Data availability	18
References	18
About authors	19

Edge AI: Small Language Models on the Go	21
Joachim J. Włodarz	
1 Introduction	22
2 LLMs and resource constraints	22
3 Small language models	23
4 Model compression and optimization	23
5 Typical applications, use cases and benefits	24
6 Hardware platforms for edge AI with SLMs	26
6.1 Edge AI SBCs	27
6.2 Edge AI TCCs	28
6.3 Edge AI MCUs	29
6.4 Edge AI smartphones	31
7 Future directions	32
8 Conclusions	35
References	35
About author	40
Supervised Machine Learning for Renewable Energy	41
Dijana Oreški, Vjeran Strahonja, and Marija Pokos Lukinec	
1 Introduction	42
2 Materials and methods	43
2.1 Research objectives	43
2.2 Related works	44
2.3 Data description	45
2.4 Deep learning algorithms	46
3 Research results	47
3.1 Model architecture	47
3.2 Hyperparameter optimization	48
3.3 Performance evaluation and comparative analysis	49
4 Discussion	50
4.1 Economic impact	51
4.2 Environmental sustainability impact	51
5 Conclusions	53
References	55
About authors	56
Financing Green Solutions: Asset Returns and Tail Risks	57
Eugenijus Gabrielius Ivanauskas, Liepa Urbonaitė, and Saulius Jokubaitis	
1 Introduction	58
1.1 Regulatory drivers and market growth	58
1.2 Green asset classes	59
1.3 Systemic risk and the decoupling hypothesis	60
2 Methodology	62
2.1 GJR-GARCH	63
2.2 csGARCH model	63
2.3 eGARCH model	64

2.4	Copulas	65
2.5	Vine copula structures	65
2.6	Value at risk and expected shortfall calculation	68
2.7	Moving window approach	69
2.8	Backtesting value at risk	70
2.9	Backtesting expected shortfall	72
3	Data	73
4	Main results	74
	Conclusions	81
	References	83
	About authors	86
Fault Detection in Solar Power Plants Based on Energy Production Data		87
Dominykas Vilčinskis, Lukas Voveris, and Jolita Bernatavičienė		
1	Introduction	88
2	Data	89
3	Methodology	92
3.1	Outlier detection	94
3.2	Fault profile construction	97
4	Experimental results	97
5	Conclusions	102
	References	104
	About authors	105
Machine Learning for Air Quality and CO_2 Emissions: The Role of Data Understanding		107
Ivan Maslov, Agnieszka Głowacka, Bartosz Dziewit, and Paulina Trybek		
1	Introduction	108
2	Analysis of carbon emission data	110
2.1	CO_2 emissions and relevant datasets	112
2.2	Datasets review	114
2.3	Challenges for machine learning applications	116
2.4	Example of prediction of CO_2	117
3	Analysis of $PM_{2.5}$ and PM_{10}	121
3.1	Factors influencing $PM_{2.5}$ and PM_{10} concentrations	121
3.2	$PM_{2.5}$ and PM_{10} data characteristics	121
3.3	OpenAQ platform	122
3.4	Meteostat	122
3.5	Integration of environmental data	123
3.6	Example of PM prediction	123
4	Discussion	125
	References	127
	About authors	129

Nanoplastics and Biostructures: Exploring the Capabilities of MD		
Computer Simulations		131
Jakub Kraus, Piotr Mika, Mateusz Wiśniewski, Paulina Trybek, and Zbigniew Dendzik		
1	Introduction	132
2	Micro and nano plastics in Biological Systems	133
2.1	Sources, sizes, and environmental occurrence	133
2.2	Uptake, translocation, and bioaccumulation	133
2.3	General toxicological endpoints	133
3	Direct effects on proteins and other biomolecules	134
3.1	Adsorption and protein corona formation	134
3.2	Nanoplastics and secondary structure in model peptides	134
3.3	Nanoplastics and membrane proteins	135
3.4	Other biomolecules: nucleic acids and carbohydrates	136
4	Molecular dynamics methodology	136
4.1	General MD framework	136
4.2	Representing nanoplastics	137
4.3	System construction: proteins and membranes	138
4.4	Sampling considerations and enhanced methods	138
4.5	MD simulation workflow	139
5	Example simulations	140
5.1	Nanoplastic-model protein adsorption	140
5.2	Nanoplastic-lipid bilayer simulations	141
6	Physical quantities from MD and their interpretation	143
7	Nanoplastics, oxidative stress, and adverse outcome pathways	146
8	Perspectives, challenges, and future directions	147
	References	148
	About authors	149
Modeling Heat Transfer in an Urban Settlement with 3D Cellular Automata and Artificial Intelligence		151
Robert Leskovar		
1	Introduction	152
2	Methodology	152
2.1	Problem statement and goals	152
2.2	Methods	154
2.3	Assumptions and limitations	154
3	Research	155
3.1	City geometry	155
3.2	Physics of the CA update	157
3.3	Sky and cloud model	160
3.4	Human–AI interaction in code development	161
4	Preliminary Results and Visualisation	162
4.1	Five-panel temperature dashboard	162
4.2	X–Z cross-section heatmaps	163
4.3	Vertical plume profile	164

5	Discussion	165
5.1	Physical plausibility	165
5.2	Human-machine interaction	166
5.3	Limitations and future work	166
	References	167
	About author	168



University of Maribor Press

Tree-based Machine Learning Methods for Wind Farm Data

Manohar Gowdru Shridhara,
Lubomír Antoni, and
Gabriel Semanišin

Abstract

Environmental and energy datasets are typically characterized by nonlinear dependencies and a combination of numerical and categorical variables. Such characteristics require more adaptable computational approaches. In this context, we explore tree-based machine learning methods since they provide a high predictive performance and a high level of interpretability. In this chapter, we present a comparative study of selected tree-based regression models applied to real-world environmental data from the United States Wind Turbine Database. The evaluated methods include a single regression decision tree, a bagging-based Random Forest ensemble, and modern gradient boosting implementations represented by CatBoost and LightGBM. All models are trained within a unified framework using standard regression performance metrics. We demonstrate that ensemble-based approaches substantially outperform a single decision tree in our experimental results. In particular, boosting-based models achieve higher predictive accuracy, with LightGBM providing the best overall performance in terms of squared error metrics and coefficient of determination. Feature importance analysis further highlights the key role of technical turbine characteristics and categorical descriptors. The findings confirm that modern gradient boosting frameworks represent a powerful and effective solution for regression tasks involving large-scale environmental and energy-related datasets.

Keywords: tree-based learning; regression; ensemble methods; gradient boosting; Random Forest; LightGBM; environmental data; wind energy

Manohar Gowdru Shridhara

Pavol Jozef Šafárik University in Košice, Faculty of Science, Jesenná 5, 04001 Košice, Slovakia
e-mail: manohar.gowdru.shridhara@student.upjs.sk

Lubomír Antoni

Pavol Jozef Šafárik University in Košice, Faculty of Science, Jesenná 5, 04001 Košice, Slovakia
e-mail: lubomir.antonip@upjs.sk

Gabriel Semanišin

Pavol Jozef Šafárik University in Košice, Faculty of Science, Jesenná 5, 04001 Košice, Slovakia
e-mail: gabriel.semansin@upjs.sk

1 Introduction

The rapid growth of environmental and energy-related datasets has significantly increased the demand for interpretable machine learning methods capable of modeling nonlinear relationships. In particular, data-driven approaches play an increasingly important role in the analysis and optimization of renewable energy systems, where large-scale heterogeneous datasets are commonly generated. Among various machine learning paradigms, decision tree-based methods have gained an important role due to their flexibility, interpretability, and strong performance in regression and prediction tasks [1, 2].

Decision trees represent a class of supervised learning models that recursively partition the input space in order to minimize prediction error. Their natural ability to capture nonlinear dependencies and interactions among variables makes them particularly appropriate for environmental data, which often contains complex spatial and technical characteristics. However, single decision trees tend to overfit and have high variance, which has motivated the development of ensemble-based approaches such as bagging and boosting [3, 4, 5].

Bagging methods, most notably Random Forests, address the variance issue by constructing an ensemble of decorrelated trees trained on bootstrap samples of the data [4]. Random Forests have proven to be highly effective in a wide range of regression problems, including applications in environmental monitoring and renewable energy. In contrast, boosting techniques build ensembles sequentially, where each subsequent model focuses on correcting the errors of its predecessors. Gradient Boosting and its modern variants have demonstrated strong predictive performance, mainly in structured tabular data [5].

In addition to classical tree-based ensembles, fuzzy decision trees extend the standard framework by incorporating fuzzy logic principles [6]. By allowing soft partitioning of the feature space, fuzzy trees provide a natural mechanism for covering uncertainty and gradual transitions between decision regions, which are common in real-world environmental data. These properties make fuzzy tree-based models attractive not only from a predictive point of view but also in terms of interpretability and robustness.

In this paper, we focus on a theoretical overview and experimental evaluation of selected tree-based machine learning methods for regression tasks in the context of environmental data analysis. The methods considered include classical decision trees, Random Forests, boosting-based ensembles, and fuzzy decision tree approaches. The experimental part of the study is based on real-world data from the United States Wind Turbine Database, which provides detailed technical and geographical information about wind energy installations across the United States. By combining theoretical insights with empirical results, we aim to highlight the strengths and limitations of tree-based learning methods when applied to large-scale environmental datasets. The paper is structured in the following way: In Section 2, we present theoretical foundations and principles of decision tree-based learning methods. In Section 3, we provide a description of the wind farm dataset. In Section 4, we present the methodology and our experimental setup. We summarize the results and discussion in Section 5.

2 Learning methods of decision trees

2.1 Decision trees for regression

Decision trees are hierarchical, non-parametric supervised learning models that recursively partition the input feature space into disjoint regions in order to approximate an underlying target function. In the case of regression tasks, the goal is to predict continuous output values by minimizing an error criterion within each partition. Regression trees are particularly attractive due to their intuitive structure, simplicity of interpretation, and ability to model nonlinear relationships without requiring explicit assumptions about the data distribution [3].

A regression decision tree is typically constructed using a top-down, greedy algorithm. At each internal node, the algorithm selects a splitting attribute and a corresponding threshold that best separates the data according to a predefined impurity or error measure. Commonly used criteria for regression trees include the mean squared error (MSE) or variance reduction, where the optimal split minimizes the weighted sum of variances in the resulting child nodes [7]. The tree-growing process continues recursively until a stopping condition is met, such as a minimum number of samples in a leaf node or a maximum tree depth.

Let d be the number of input features. A regression tree partitions the feature space by selecting, at each internal node, a feature index $j \in \{1, \dots, d\}$ and a threshold $t \in \mathbb{R}$ that minimizes the within-node squared error after the split. Let S denote the set of training indices reaching the current node. The split induces two subsets

$$S_L(j, t) = \{i \in S : x_{ij} \leq t\}, \quad S_R(j, t) = \{i \in S : x_{ij} > t\}.$$

A common objective for regression trees is the sum of squared errors (SSE) in the two children:

$$(j^*, t^*) \in \arg \min_{j, t} \left(\sum_{i \in S_L(j, t)} (y_i - \bar{y}_L)^2 + \sum_{i \in S_R(j, t)} (y_i - \bar{y}_R)^2 \right),$$

where \bar{y}_L and \bar{y}_R are the mean target values in the left and right child, respectively. When a stopping condition is met (e.g., maximum depth, minimum number of samples in a node, or no improvement in the objective), the node becomes a leaf and outputs the constant prediction

$$\hat{y}(x) = \bar{y}_S = \frac{1}{|S|} \sum_{i \in S} y_i.$$

Algorithm 1 summarizes a simplified top-down procedure for constructing a regression tree which can be found e.g. in [3].

Each terminal node, or leaf, represents a local model that outputs a constant prediction, usually computed as the mean of the target values of the samples contained in that leaf. While this piecewise-constant approximation enables regression trees to capture complex nonlinear dependencies and interactions between variables, it also makes them

Algorithm 1 Simplified procedure for building a regression tree [3]

Require: Training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, current node index set S , depth D

Ensure: A regression tree node (either internal split or leaf)

```

1: if  $D \geq D_{\max}$  or  $|S| < n_{\min}$  then
2:   return leaf with prediction  $\bar{y}_S$ 
3: end if
4: Find  $(j^*, t^*)$  minimizing SSE over candidate splits (feature  $j$ , threshold  $t$ )
5: Compute  $S_L \leftarrow \{i \in S : x_{ij^*} \leq t^*\}$  and  $S_R \leftarrow \{i \in S : x_{ij^*} > t^*\}$ 
6: if  $S_L = \emptyset$  or  $S_R = \emptyset$  then
7:   return leaf with prediction  $\bar{y}_S$ 
8: end if
9: Create internal node storing  $(j^*, t^*)$ 
10: Left child  $\leftarrow$  BUILDTREE( $\{(\mathbf{x}_i, y_i)\}, S_L, D + 1$ )
11: Right child  $\leftarrow$  BUILDTREE( $\{(\mathbf{x}_i, y_i)\}, S_R, D + 1$ )
12: return internal node with left and right children
  
```

highly sensitive to small perturbations in the training data. As a result, regression trees are known to suffer from high variance and a tendency to overfit, especially when grown to full depth [3].

Despite these limitations, regression trees provide several important advantages in the context of environmental and energy-related datasets. They naturally handle mixed data types, are robust to outliers, and allow for straightforward assessment of variable importance through split statistics. These properties make decision trees a fundamental building block for more advanced ensemble methods such as Random Forests and boosting algorithms, which aim to improve predictive performance by addressing the instability of individual trees.

2.2 Bagging and random forests

One of the main limitations of single regression trees is their high variance, which results from their sensitivity to small changes in the training data. Ensemble learning methods aim to address this issue by combining multiple base learners in order to obtain more stable and accurate predictions. Bagging, short for bootstrap aggregating, is one of the earliest and most widely used ensemble techniques designed to reduce variance without increasing bias [3, 8].

The bagging approach constructs multiple training datasets by sampling with replacement from the original dataset. A separate regression tree is trained on each bootstrap sample, and the final prediction is obtained by averaging the predictions of all individual trees. Since each tree is trained on a slightly different subset of the data, the ensemble benefits from reduced variance while preserving the expressive power of deep trees. Bagging is particularly effective for unstable learners such as decision trees, making it a natural extension of the regression tree model [3].

Random Forests further extend the bagging principle by introducing an additional layer of randomness during tree construction. In addition to bootstrap sampling of the

training data, Random Forests randomly select a subset of input features at each split node, thereby reducing the correlation between individual trees in the ensemble [4]. This decorrelation effect significantly improves generalization performance, especially in high-dimensional settings or when strong predictor variables dominate the splitting process.

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denote the training dataset. In the bagging framework, B bootstrap samples are generated by sampling with replacement from the original dataset. For each bootstrap sample $b \in \{1, \dots, B\}$, an independent regression tree $f_b(\mathbf{x})$ is trained.

The bagging predictor is defined as the average of individual tree predictions:

$$\hat{f}_{\text{bag}}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B f_b(\mathbf{x}).$$

Random Forests extend the bagging principle by introducing additional randomness during tree construction. At each split node, only a randomly selected subset of features $\mathcal{M} \subset \{1, \dots, d\}$ with $|\mathcal{M}| = m \ll d$ is considered when determining the optimal split. This feature subsampling mechanism reduces correlation among individual trees and improves ensemble generalization performance.

The Random Forest regression predictor is therefore given by

$$\hat{f}_{\text{RF}}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B f_b^{(\mathcal{M})}(\mathbf{x}),$$

where $f_b^{(\mathcal{M})}$ denotes a regression tree trained using random feature selection at each split.

Algorithm 2 Simplified Random Forest regression algorithm [3]

Require: Training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, number of trees B , number of features per split m

Ensure: Random Forest regression model

- 1: **for** $b = 1$ to B **do**
 - 2: Draw a bootstrap sample S_b from the training data
 - 3: Train a regression tree f_b on S_b
 - 4: **for** each split node in f_b **do**
 - 5: Randomly select m features from $\{1, \dots, d\}$
 - 6: Choose the best split using only the selected features
 - 7: **end for**
 - 8: **end for**
 - 9: **return** ensemble predictor $\hat{f}_{\text{RF}}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B f_b(\mathbf{x})$
-

From a regression perspective, Random Forests approximate the target function by averaging predictions across a large number of randomized trees. This averaging mechanism leads to robust performance, strong resistance to overfitting, and stable predictions even in the presence of noise. Moreover, Random Forests provide built-in mechanisms for estimating feature importance, typically based on impurity reduction or permutation-

based measures, which are particularly valuable in environmental applications where understanding the influence of individual variables is essential [3].

Due to their favorable bias–variance trade-off, minimal parameter tuning requirements, and ability to process large heterogeneous datasets, Random Forests have become a standard baseline method in environmental data analysis and renewable energy research.

2.3 Boosting methods

While bagging-based ensembles primarily aim to reduce variance by training base learners independently, boosting methods adopt a different strategy by constructing models sequentially. The central idea of boosting is to iteratively focus on training instances that are difficult to predict, thereby progressively improving the overall performance of the ensemble. This adaptive learning process enables boosting methods to reduce both bias and variance, making them particularly effective for difficult regression tasks [9].

Boosting regression methods construct an additive model by sequentially combining weak learners, typically shallow regression trees. Let M denote the number of iterations. The resulting model can be expressed as

$$f_M(\mathbf{x}) = \sum_{m=1}^M \gamma_m h_m(\mathbf{x}),$$

where each $h_m(\mathbf{x})$ denotes a weak regression tree and γ_m is a scaling parameter controlling its contribution to the ensemble.

In gradient boosting, the learning process is formulated as an optimization problem in function space. Given a differentiable loss function $L(y, f(\mathbf{x}))$, the model is built iteratively by fitting each new base learner to the negative gradient of the loss with respect to the current model predictions [5]. For the commonly used squared error loss,

$$L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2,$$

the negative gradient corresponds to the residual

$$r_{im} = y_i - f_{m-1}(\mathbf{x}_i).$$

At iteration m , a regression tree h_m is fitted to the residuals $\{r_{im}\}_{i=1}^N$, and the ensemble model is updated according to

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \eta h_m(\mathbf{x}),$$

where $\eta \in (0, 1]$ denotes the learning rate that controls the contribution of each newly added tree. This iterative refinement allows boosting models to progressively

reduce prediction error by concentrating learning capacity on previously mispredicted instances.

Algorithm 3 Simplified Gradient Boosting algorithm for regression [3]

Require: Training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, number of iterations M , learning rate η

Ensure: Gradient Boosting regression model

1: Initialize model with constant prediction

$$f_0(\mathbf{x}) = \arg \min_c \sum_{i=1}^N L(y_i, c)$$

2: **for** $m = 1$ to M **do**

3: Compute residuals $r_{im} \leftarrow y_i - f_{m-1}(\mathbf{x}_i)$

4: Fit regression tree h_m to $\{(\mathbf{x}_i, r_{im})\}_{i=1}^N$

5: Update model: $f_m(\mathbf{x}) \leftarrow f_{m-1}(\mathbf{x}) + \eta h_m(\mathbf{x})$

6: **end for**

7: **return** final model $f_M(\mathbf{x})$

Modern boosting frameworks extend the classical gradient boosting paradigm by incorporating advanced regularization strategies, efficient tree-growing mechanisms, and specialized handling of categorical features. Notable examples include CatBoost and LightGBM, which have demonstrated state-of-the-art performance on structured tabular datasets [10, 27].

CatBoost introduces ordered boosting and target-based encoding schemes to reduce bias in predictions caused by many-category features [10]. These properties make CatBoost particularly suitable for real-world datasets containing mixed numerical and categorical variables, as commonly encountered in environmental and energy-related applications.

LightGBM employs histogram-based feature binning and a leaf-wise tree growth strategy that enable efficient modeling of complex feature interactions with reduced computational cost [27]. By choosing splits that most reduce the loss, LightGBM often achieves higher predictive performance compared to level-wise boosting approaches, especially on large-scale datasets.

Despite their strong predictive capabilities, boosting-based models are generally more sensitive to noise and hyperparameter configuration than bagging-based ensembles. Overfitting may occur if there are too many boosting iterations or the individual trees are too complex. Nevertheless, when appropriately regularized and tuned, modern boosting methods frequently outperform Random Forests in regression tasks that require capturing nonlinear relationships and interactions between attributes.

2.4 Fuzzy decision trees and fuzzy ensembles

Classical decision trees rely on crisp, binary splits of the input feature space, which may be insufficient for modeling gradual relationships and uncertainty commonly present in

real-world data. Fuzzy decision trees address this limitation by incorporating concepts from fuzzy logic [6], allowing instances to belong to multiple decision nodes with varying degrees of membership. This soft partitioning of the feature space enables smoother decision boundaries and improved robustness to noise [12].

In fuzzy decision trees, splitting criteria are based on fuzzy membership functions rather than hard thresholds. Continuous attributes are typically represented using linguistic terms, such as *low*, *medium*, and *high*, each associated with a corresponding membership function. During the tree construction process, samples propagate through multiple branches simultaneously, weighted by their membership degrees. Leaf nodes aggregate these contributions to produce final predictions, often through weighted averaging mechanisms [12].

The use of fuzzy logic enhances interpretability by enabling rule-based representations that are similar to human reasoning. Each root-to-leaf path can be interpreted as a fuzzy rule, making fuzzy decision trees particularly attractive in domains where transparency and explainability are essential. Moreover, fuzzy trees have increased tolerance to measurement uncertainty and imprecise attribute values, which are typical characteristics of environmental and energy-related datasets.

Fuzzy ensemble methods extend the principles of fuzzy decision trees by combining multiple fuzzy trees into an ensemble framework. Similar to classical bagging and boosting, fuzzy ensembles aim to improve predictive performance and stability. Approaches such as fuzzy random forests and fuzzy boosting incorporate randomness or sequential learning while preserving fuzzy partitions of the feature space [13]. These hybrid models aim to balance predictive accuracy with interpretability, using the strengths of both ensemble learning and fuzzy reasoning.

In fuzzy decision trees, the partitioning of the feature space is based on fuzzy sets rather than crisp thresholds. Let x_{ij} denote the value of feature j for instance i . Each continuous attribute is associated with a collection of fuzzy sets $\{A_{j1}, \dots, A_{jk}\}$ characterized by membership functions

$$\mu_{jl} : \mathbb{R} \rightarrow [0, 1],$$

which quantify the degree to which x_{ij} belongs to the linguistic term A_{jl} .

At a given internal node, an instance is propagated to multiple child nodes simultaneously, weighted by its membership degrees. For a node corresponding to fuzzy set A_{jl} , the weighted subset of instances is defined as

$$S_{jl} = \{(i, \mu_{jl}(x_{ij})) : i \in S\},$$

where S denotes the set of instances reaching the parent node.

For regression tasks, the prediction at a fuzzy leaf node is typically computed as a weighted average of target values:

$$\hat{y}_A = \frac{\sum_{i \in S} \mu_A(\mathbf{x}_i) y_i}{\sum_{i \in S} \mu_A(\mathbf{x}_i)},$$

where $\mu_A(\mathbf{x}_i)$ represents the aggregated membership degree of instance \mathbf{x}_i in the corresponding fuzzy rule. This formulation allows smooth transitions between decision regions and provides robustness against measurement uncertainty.

Algorithm 4 Simplified construction of a fuzzy regression tree

Require: Training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, fuzzy partitions for each feature

Ensure: Fuzzy regression tree

- 1: **if** stopping criterion satisfied **then**
 - 2: Compute weighted prediction \hat{y} using membership degrees
 - 3: **return** fuzzy leaf node
 - 4: **end if**
 - 5: Select feature j^* and corresponding fuzzy sets $\{A_{j^*l}\}$
 - 6: **for** each fuzzy set A_{j^*l} **do**
 - 7: Compute membership degrees $\mu_{j^*l}(x_{ij^*})$ for all instances
 - 8: Propagate instances to child node with weights μ_{j^*l}
 - 9: Recursively build fuzzy subtree
 - 10: **end for**
 - 11: **return** internal fuzzy node with fuzzy children
-

Fuzzy ensemble models combine multiple fuzzy regression trees by aggregating their predictions. Given an ensemble of B fuzzy trees $\{f_b\}_{b=1}^B$, the final prediction is commonly obtained as

$$\hat{f}_{\text{fuzzy}}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B f_b(\mathbf{x}),$$

analogously to classical ensemble methods, while preserving fuzzy membership propagation within individual trees.

Fuzzy tree-based methods provide a sound alternative for regression tasks characterized by uncertainty, vagueness, and nonlinear interactions. As such, fuzzy decision trees and fuzzy ensembles can represent a valuable extension of classical tree-based learning methods in environmental data analysis.

3 Dataset description

This section describes the dataset used in the experimental part of the study. The focus is placed on the origin of the data, its scope, and its relevance for modeling regression tasks in the context of environmental and renewable energy analysis. Particular attention is given to the structure of the dataset and its suitability for evaluating tree-based machine learning methods.

3.1 US Wind turbine database

The experimental evaluation is based on data obtained from the United States Wind Turbine Database (USWTDB), a comprehensive and publicly available repository maintained by the U.S. Geological Survey in collaboration with national laboratories and governmental agencies [14]. The database provides detailed information on utility-scale wind turbines installed across the United States and represents one of the most authoritative sources of wind energy infrastructure data.

The USWTDB contains records for tens of thousands of wind turbines, covering a wide geographical area and multiple generations of wind energy technologies. Each turbine is described by a rich set of attributes, including geographical coordinates, technical specifications, operational status, and installation metadata. The dataset is continuously updated to reflect newly installed turbines as well as revisions to existing records, ensuring a high level of accuracy and temporal relevance.

Due to its scale and heterogeneity, the USWTDB constitutes a representative example of real-world environmental data characterized by nonlinear relationships, mixed attribute types, and potential measurement uncertainty. These properties make the dataset particularly suitable for evaluating decision tree-based regression models and ensemble learning methods. In this study, a selected subset of numerical and categorical attributes is used to formulate a regression task aimed at modeling relationships between turbine characteristics and target variables of interest.

3.2 Data attributes

The United States Wind Turbine Database provides a comprehensive set of attributes describing individual wind turbines from both technical and geographical perspectives. Attribute definitions and metadata are publicly documented in a structured XML format, which ensures consistency and transparency across dataset versions [15]. In this study, we use a selected subset of these attributes relevant to regression-based modeling of wind turbine characteristics.

From a structural perspective, the attributes can be grouped into several logical categories. The first group consists of geographical attributes, including latitude and longitude coordinates, state identifiers, and county-level information. These variables capture the spatial distribution of wind turbines and indirectly reflect environmental and regulatory conditions that may influence turbine design and deployment.

The second group contains the technical and physical characteristics of wind turbines. These attributes include hub height, rotor diameter, nameplate capacity, and turbine manufacturer information. Such variables are directly related to turbine performance and are therefore particularly relevant for regression tasks aimed at modeling capacity-related or structural properties. The numerical nature of these attributes makes them well-suited for tree-based learning methods, which naturally deal with nonlinear interactions and threshold-based relationships.

Additional attributes describe installation and operational metadata, such as the year of installation, turbine status, and project identifiers. While some of these variables are categorical or temporal in nature, they provide contextual information that may improve predictive performance when appropriately encoded. In the experimental setup, categorical attributes are transformed into numerical representations when required, while attributes with limited analytical relevance are excluded.

Overall, the selected attributes form a heterogeneous feature space that combines spatial, technical, and contextual information. An overview of the selected attributes from the US Wind Turbine Database used in our regression task is shown in Table 1. The dataset contains 70 697 instances.

Table 1: Overview of selected attributes from the US Wind Turbine Database used in our regression task.

Attribute	Type	Description
Latitude	Numerical	Geographic latitude of the turbine location
Longitude	Numerical	Geographic longitude of the turbine location
State	Categorical	U.S. state of installation
County	Categorical	County-level administrative region
Year of Installation	Numerical	Year when the turbine became operational
Hub Height	Numerical	Height of the turbine hub above ground (m)
Rotor Diameter	Numerical	Diameter of the turbine rotor (m)
Nameplate Capacity	Numerical	Rated turbine power capacity (kW)
Turbine Manufacturer	Categorical	Manufacturer of the wind turbine
Turbine Model	Categorical	Model designation of the turbine

3.3 Data preprocessing

Before model training and evaluation, we applied a series of preprocessing steps to ensure data consistency and suitability for tree-based regression models. Given the heterogeneous structure of the United States Wind Turbine Database, particular attention was paid to attribute selection in order to avoid introducing unnecessary bias into the learning process.

We started the preprocessing pipeline by selecting a subset of attributes relevant to the regression task. The target variable was defined as the turbine nameplate capacity, while the input features consisted of selected technical, temporal, and geographical attributes. Specifically, numerical features included hub height, rotor diameter, total turbine height, longitude, latitude, year of installation, and the number of turbines within a plant. In addition, categorical attributes describing turbine state, manufacturer, and model were retained to capture design- and location-specific characteristics.

Rows with missing values in the target variable were removed from the dataset. Subsequently, observations containing missing values in any of the selected input features were also excluded. This conservative strategy was adopted to ensure that all models were trained and evaluated on complete cases only, thereby simplifying the experimental setup and improving result reproducibility.

Categorical attributes were explicitly converted to string representations to guarantee consistent data types and to prevent issues arising from mixed numerical and textual encodings. Numerical attributes were retained in their original scale, as tree-based learning methods are naturally insensitive to feature scaling and monotonic transformations.

After preprocessing, the dataset was randomly partitioned into training and testing subsets using an 80%–20% split with a fixed random seed. The training set was used exclusively for model fitting, while the testing set was reserved for performance evaluation. This separation enables an unbiased assessment of generalization performance and ensures a fair comparison of the evaluated regression models.

Overall, the preprocessing procedure reflects practical conditions commonly encountered in data analysis, prioritizing robustness and transparency over aggressive data transformation. The resulting dataset provides a reliable basis for the experimental comparison of decision tree-based regression methods presented in the following sections.

4 Methodology

This section describes the experimental design used to evaluate the performance of selected tree-based machine learning methods on the wind turbine dataset. The focus is placed on the formulation of the regression task, the definition of input and output variables, and the methodological considerations ensuring a fair and reproducible comparison of different models (Fig. 1).

4.1 Regression task definition

The considered learning problem is formulated as a supervised regression task, where the objective is to model the relationship between a set of input attributes describing wind turbine characteristics and a continuous target variable. Let $\mathbf{x}_i \in \mathbb{R}^d$ denote the feature vector associated with the i -th wind turbine, where d represents the number of selected attributes, and let $y_i \in \mathbb{R}$ denote the corresponding target value.

Given a training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the goal of the regression model is to learn a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ that approximates the underlying dependency between turbine features and the target variable. The learned function is subsequently used to predict the target values for unseen instances in the test dataset.

In this study, the input feature vector \mathbf{x}_i comprises a combination of geographical attributes (such as latitude and longitude) and technical parameters (including hub height, rotor diameter). The target variable (turbine capacity) is selected to represent a contin-

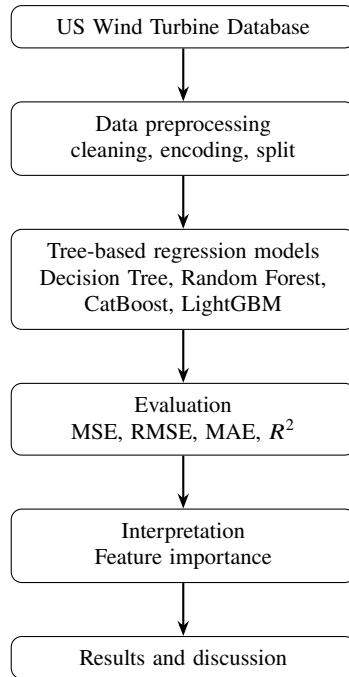


Fig. 1: Simplified compact workflow of the proposed approach.

uous turbine-related characteristic of practical relevance in the context of wind energy analysis. This formulation allows the evaluation of model performance in capturing non-linear interactions between spatial and technical factors.

The regression task is designed to reflect realistic modeling conditions encountered in environmental datasets, where the relationships between variables are often complex and influenced by multiple interacting factors.

4.2 Implemented models

To evaluate the effectiveness of decision tree-based learning approaches for environmental regression tasks, several representative models were implemented and compared within a unified experimental framework. The selected models cover both classical tree learners and modern ensemble-based methods that are widely adopted in contemporary machine learning practice for tabular data.

As a baseline model, a single regression decision tree was employed. This model serves as a reference point for assessing the benefits of ensemble learning techniques. The regression tree was trained using variance-based splitting criteria and constrained by regularization parameters, such as a minimum of 5 samples per leaf to reduce overfitting and ensure meaningful generalization.

Bagging-based ensemble learning was represented by the Random Forest regressor. In this approach, 300 regression trees are trained independently on bootstrap samples of the training data, with additional randomization introduced through feature subsampling at each split. The final prediction is obtained by averaging the outputs of all individual trees. Random Forests are known for their robustness, stability, and relatively low sensitivity to hyperparameter tuning, making them a strong baseline for heterogeneous environmental datasets.

Boosting-based ensemble methods represent the third category of evaluated models. In particular, two modern gradient boosting implementations were employed. The first is CatBoost (3000 iterations, learning rate of 0.05, depth of 8, and RMSE loss function), which constructs ensembles of decision trees in a sequential manner and is specifically designed to handle categorical attributes effectively. By employing ordered boosting and target-based encoding, CatBoost reduces prediction bias and mitigates overfitting when working with high-cardinality categorical features commonly found in real-world datasets.

The second boosting-based model is LightGBM, a highly efficient gradient boosting framework that employs histogram-based feature binning and leaf-wise tree growth strategies. This design enables LightGBM to capture complex nonlinear relationships and feature interactions with high computational efficiency. Model complexity was controlled through parameters such as 5000 boosting iterations, a learning rate of 0.03, a minimum of 30 child samples, and regularization terms to balance predictive accuracy and generalization.

All models were trained and evaluated using identical training and testing splits to ensure a fair comparison. Hyperparameters were selected based on recommended practices for each algorithm and validated through preliminary experimentation.

4.3 Evaluation metrics

The predictive performance of the implemented regression models was assessed using several commonly adopted evaluation metrics for continuous-valued targets. These metrics quantify different aspects of prediction error and together provide a comprehensive view of model accuracy and robustness.

The primary evaluation metric used in this study is the mean squared error (MSE), defined as

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2,$$

where y_i denotes the true target value and \hat{y}_i represents the corresponding model prediction. MSE penalizes larger errors more strongly due to the squared term and is therefore sensitive to outliers.

To provide an error measure expressed in the same units as the target variable, the root mean squared error (RMSE) is also reported. RMSE is obtained as the square root of MSE and is often easier to interpret in practical applications. Additionally, the mean

absolute error (MAE) is included as a complementary metric that measures the average magnitude of prediction errors without emphasizing extreme deviations.

Finally, the coefficient of determination (R^2) is used to evaluate the proportion of variance in the target variable that is explained by the model. The R^2 score provides a normalized measure of goodness-of-fit and enables comparison across different models and datasets.

The combination of these metrics allows for balanced evaluation of model performance, capturing both overall accuracy and sensitivity to large prediction errors. All evaluation measures are computed on the test dataset, which remains unseen during model training and parameter selection.

5 Results and discussion

In this section, we present and discuss the experimental results obtained from applying selected decision tree-based regression models to the United States Wind Turbine Database. The evaluated models include a single regression tree, a Random Forest ensemble, a CatBoost regressor, and a LightGBM regressor. Model performance is assessed using the evaluation metrics defined in Section 4.3, and interpretability is examined through feature importance analysis.

5.1 Quantitative results

Table 2 summarizes the predictive performance of all evaluated models in terms of mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and the coefficient of determination (R^2). All results are reported on the held-out test dataset.

Table 2: Regression performance of tree-based models on the USWTDB dataset.

Model	MSE	RMSE	MAE	R^2
Decision Tree	141.25	11.88	0.72	0.99978
Random Forest	105.19	10.26	0.71	0.99983
CatBoost	83.34	9.13	1.33	0.99987
LightGBM	77.50	8.80	0.74	0.99988

The results clearly demonstrate the advantages of ensemble-based learning methods over a single regression tree. While the baseline decision tree achieves relatively high predictive accuracy, its performance is consistently worse than that of ensemble models across all error-based metrics except for MAE, where the Decision Tree (0.72) outperforms CatBoost (1.33).

Among the evaluated approaches, LightGBM achieves the best overall performance, obtaining the lowest MSE and RMSE as well as the highest R^2 value. CatBoost also performs strongly, outperforming both the single tree and the Random Forest ensemble in terms of squared error metrics. These results indicate that boosting-based methods are particularly effective in capturing complex nonlinear relationships present in the wind turbine dataset.

Interestingly, the Random Forest model, while robust and competitive, is outperformed by both boosting-based approaches in this experimental setting in RMSE and R^2 value. This observation suggests that the sequential error-correction mechanism used in boosting methods provides an advantage over variance reduction alone when modeling structured tabular data with strong dominant predictors.

5.2 Model comparison

A more detailed comparison of the evaluated models highlights the practical implications of different ensemble learning strategies. The Random Forest regressor exhibits stable and reliable performance, benefiting from bootstrap aggregation and feature-level randomization. Its relatively low MAE indicates strong robustness to moderate prediction errors, making it suitable for applications where stability and interpretability are prioritized.

In contrast, boosting-based models, particularly LightGBM, achieve higher performance in terms of squared error metrics. By constructing trees in a sequential manner and optimizing leaf-wise growth, LightGBM is able to model fine-grained interactions between numerical and categorical attributes more effectively than bagging-based ensembles. The improved performance of boosting methods reflects their ability to focus learning on hard examples and reduce residual errors iteratively.

While Random Forests remain a strong baseline for heterogeneous environmental datasets, modern gradient boosting implementations such as LightGBM can offer strong performance when properly configured.

5.3 Feature importance analysis

We conducted the feature importance to provide model interpretability for the Random Forest and LightGBM models. It should be noted that feature importance values are derived differently for each model.

For the Random Forest model, feature importance is based on impurity reduction aggregated across all trees in the ensemble. We found that rotor diameter is by far the most influential predictor, accounting for the majority of explained variance. This result is consistent with domain knowledge in wind energy systems, where the rotor diameter affects the energy capture potential of a turbine. Additional contributions are observed from the turbine model and manufacturer, reflecting design-specific characteristics, while temporal and geographical attributes play a secondary role.

In contrast, LightGBM computes feature importance based on split frequency and gain across leaf-wise tree growth. The resulting importance distribution highlights a broader set of influential features, including geographical coordinates, turbine identifiers, and temporal variables. This indicates that LightGBM exploits more complex interactions between spatial, technical, and contextual attributes when forming predictions.

Despite differences in importance ranking mechanisms, both models consistently identify turbine-specific technical attributes and categorical descriptors as key drivers of predictive performance. Overall, the results confirm that modern boosting-based methods, particularly LightGBM, offer state-of-the-art performance for regression tasks involving large-scale environmental and energy-related datasets.

6 Conclusion and future work

In this chapter, we present a systematic study of selected decision-tree-based machine learning methods applied to regression tasks in the context of environmental data analysis. The theoretical background of classical decision trees, ensemble learning techniques based on bagging and boosting, as well as fuzzy decision tree approaches, was complemented by an experimental evaluation using real-world data from the United States Wind Turbine Database.

The experimental results demonstrate that ensemble-based methods significantly outperform a single regression tree in terms of predictive accuracy and robustness. Random Forests provide a balance between performance and stability, while boosting-based models achieve higher accuracy by capturing fine-grained nonlinear relationships among turbine characteristics. Feature importance analysis further highlights the interpretability of tree-based models and confirms the dominant influence of technical turbine attributes on model predictions, in agreement with domain knowledge from wind energy systems.

Despite the strong performance of the evaluated models, we acknowledge several limitations. The analysis focuses on a static snapshot of turbine data and does not explicitly account for temporal dynamics or spatial dependencies. Moreover, the experimental setup prioritizes methodological comparison over exhaustive hyperparameter optimization, which may further improve predictive performance.

In our future work, we aim to explore extensions of the presented approach in several directions. These include the incorporation of fuzzy ensemble methods to better handle uncertainty and vagueness in environmental data, the integration of spatial and temporal modeling techniques, and the application of advanced explainability tools to further enhance model transparency. Additionally, the proposed methodology may be extended to other types of renewable energy datasets and environmental monitoring applications.

Data availability

Data are available from U.S. Wind Turbine Database [16], provided by the U.S. Geological Survey, American Clean Power Association, and Lawrence Berkeley National Laboratory via <https://energy.usgs.gov/uswtodb>

Acknowledgement

This work was supported by the VVGS ESGD grant (Early Stage Grants – Pavol Jozef Šafárik University in Košice), led by Ing. Manohar Gowdru Shridhara, ŠPP element 0HV040126 / FS 190180 / 2887 – VVGS ESGD – Gowdru, Faculty of Science, within the project “Early Stage Grants – Pavol Jozef Šafárik University in Košice”, code 09I03-03-V05-00008. (M. Gowdru Shridhara).

References

1. Zhang, Y., Wang, J. X., & Wang, X. F. (2014). Review on probabilistic forecasting of wind power generation. *Renewable and Sustainable Energy Reviews*, 32, 255–270. <https://doi.org/10.1016/j.rser.2014.01.033>
2. Abisoye, B. O., Sun, Y., & Wang, Z. (2024). A survey of artificial intelligence methods for renewable energy forecasting: Methodologies and insights. *Renewable Energy*, 221, Article 100529. <https://doi.org/10.1016/j.renene.2023.100529>
3. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer. <https://doi.org/10.1007/978-0-387-84858-7>
4. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
5. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
6. 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)
7. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth International Group.
8. Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140. <https://doi.org/10.1007/BF00058655>
9. Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5(2), 197–227. <https://doi.org/10.1007/BF00116037>
10. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31, 6638–6648.
11. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 3146–3154.
12. Yuan, Y., & Shaw, M. J. (1995). Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, 69(2), 125–139. [https://doi.org/10.1016/0165-0114\(94\)00229-Z](https://doi.org/10.1016/0165-0114(94)00229-Z)
13. Pal, S. K., & Mitra, P. (2004). Case generation using rough sets with fuzzy representation. *IEEE Transactions on Knowledge and Data Engineering*, 16(3), 292–300. <https://doi.org/10.1109/TKDE.2003.1262181>

14. U.S. Geological Survey. (2024). *United States wind turbine database (USWTDB)* (Version 7.1) [Data set]. <https://eerscmapp.usgs.gov/uswtodb/>
15. U.S. Geological Survey. (2024). *United States wind turbine database – Attribute definitions (XML metadata)* (Version 7.1) [Data set]. <https://data.usgs.gov/datacatalog/metadata/USGS.6001e327d34e592d8671fae0.xml>
16. Hoen, B. D., Diffendorfer, J. E., Rand, J. T., Kramer, L. A., Garrity, C. P., & Hunt, H. E. (2025). *United States wind turbine database (USWTDB)* (Version 8.2, December 10, 2025) [Data set]. U.S. Geological Survey, American Clean Power Association, & Lawrence Berkeley National Laboratory.

About authors

Manohar Gowdru Shridhara is a PhD student at the Faculty of Science, Pavol Jozef Šafárik University in Košice. His research interests include machine learning and optimization techniques, mainly in the fields of energetics and wind farms.

Eubomír Antoni is an associate professor at the Institute of Computer Science, Faculty of Science, Pavol Jozef Šafárik University in Košice. His research interests include artificial intelligence, fuzzy systems, data mining, and applied machine learning.

Gabriel Semanišín is a professor of Computer Science at Faculty of Science, Pavol Jozef Šafárik University in Košice. As part of his research activities, he focuses mainly on algorithmic graph theory and its application in various areas of theoretical and applied informatics. He is a co-guarantor of the study programs Applied Informatics, Data Analysis and Artificial Intelligence, and Computer Science. He was a supervisor of six PhD students in the study programs Computer Science, Discrete Mathematics and Theory of Teaching Informatics.



University of Maribor Press

Edge AI: Small Language Models on the Go

Joachim J. Włodarz

Abstract

The proliferation of edge devices, ranging from smartphones and various wearable devices, up to industrial sensors or autonomous vehicles, gives an opportunity to leverage the power of AI-based methods directly at the point where data is acquired or generated. However, deploying traditional Large Language Models (LLMs) on resource-constrained edge devices becomes impractical due to substantial computational and memory requirements. In this contribution, the rapidly evolving field of Edge AI is explored, specifically focusing on the development and deployment of Small Language Models (SLMs), optimized for edge environments. The various challenges and opportunities associated with SLMs are indicated, together with a review of the current techniques for model compression and optimization. An outline of future research and development is also given.

Keywords: edge AI; small language model; model compression; quantization; pruning and federated learning; on-device AI; resource-constrained devices

1 Introduction

The proliferation of Internet of Things (IoT) devices [1], coupled with the growing demand for real-time and personalized user experiences, has significantly accelerated the advancement of edge computing. Edge AI, which involves executing artificial intelligence applications directly on edge devices, offers notable advantages when compared to traditional networked or cloud-based solutions [2]. These advantages include reduced latency, enhanced data privacy and improved reliability - especially in environments with limited connectivity - as well as a decrease in bandwidth consumption. It also makes it possible to work completely offline, without access to any network.

Historically, the computational requirements of advanced AI models, particularly Large Language Models (LLMs) [3], have constrained their deployment on edge devices. However, recent breakthroughs in model compression techniques and the emergence of Small Language Models (SLMs) [4] have stimulated a major shift within this landscape.

Ultra-compact AI models are increasingly deployed on edge devices with minimal resources, even with just a few kilobytes of memory onboard [5]. It makes it possible to process data locally on commodity devices such as sensors, wearables, and home appliances. This innovative approach yields considerable benefits in energy efficiency, enabling such devices to function for years on a single battery. Additionally, it offers near-instantaneous response times and enhanced privacy, as data processing occurs on-device without the need to transmit information externally.

In this contribution, it will be shown that SLMs are a vital enabler for Edge AI, facilitating the incorporation of robust language capabilities across a myriad of applications that were once considered impractical. The specific challenges and opportunities inherent in this field will be explored, offering a comprehensive overview of current solutions and highlighting potential future directions.

2 LLMs and resource constraints

Traditional LLMs, like the GPT model series introduced by OpenAI [6], have demonstrated impressive capabilities, but they are also characterized by immense size – hundreds of billions of parameters - and huge computational complexity. Deploying these models on edge devices presents several significant hurdles. Namely, their memory footprint often exceeds by orders of magnitude the capacity of edge devices. Moreover, the computational cost of inference demands significant processing power, leading to high energy consumption and resulting in slow response times on resource-constrained hardware. Furthermore, transferring data between the edge devices and the cloud can be bandwidth-intensive and unreliable, not to mention that sending sensitive data to the cloud for processing raises privacy and security concerns.

The recent shortage of RAM chips has significantly altered the Edge AI landscape. This scarcity is prompting a crucial architectural shift, as developers increasingly adopt “DRAM-less” hardware designs, such as the Hailo-8 and Hailo-8L accelerators, which

enable the execution of AI models entirely on-chip. In addition to hardware adaptations, this shortage has made model optimization a necessity, accelerating further the transition from LLMs to more efficient SLMs. The availability of currently used LPDDR5X and DDR5 chips for edge systems is anticipated to remain critically constrained at least through the next year. In consequence, the AI scene becomes effectively bifurcated into the “high end” part enjoying priority, and the “everyone else adapts” part, where repurposed or even refurbished hardware and also various microcontroller based designs become more and more attractive for Edge AI projects.

3 Small language models

Small Language Models (SLMs) are typically defined as models with only a few billion parameters, offering a compelling alternative to traditional LLMs for Edge AI applications. While they may exhibit reduced performance on certain tasks compared to their larger counterparts, their smaller size and much lower computational requirements make them significantly more suitable for deployment on edge devices.

Several SLMs have emerged recently, each demonstrating a balance between performance and efficiency. They are the primary driver for Edge AI, enabling “intelligence” on devices like smartphones, IoT sensors, or medical equipment. The following models are widely used for on-device tasks:

- Google Gemma 3 / 3n [7]: A multimodal family (text, image, audio) with variants like Gemma 3n 1B that can reach speeds over 2,500 tokens/second on mobile GPUs.
- Microsoft Phi-4 Mini / Phi-4 [8, 9]: The 3.8B parameter Mini is highly optimized for reasoning and coding, while Phi-4 (14B) pushes the upper limit of what “small” entails.
- Meta Llama 3.2 (1B & 3B) [10]: Specifically designed for mobile and edge performance with high accuracy-to-size ratios.
- Qwen 2.5 (1.5B) [11]: An Alibaba-developed model popular for multilingual edge applications.
- TinyLlama 1.1B [12]: Compact and fast open-source model.
- Shakti Family (100M–500M) [13]: Specialized ultra-compact models designed for domain-specific tasks like legal or medical analysis on tiny hardware.
- SmoILM2 (135M & 360M & 1.7B) [14]: A new state-of-the-art family of SLMs.

4 Model compression and optimization

To further enhance the suitability of SLMs for Edge AI, various model compression and optimization techniques are being employed. For example, quantization reduces the precision of model weights and activations, e.g. from 32-bit to 4-bit integers, significantly lowering the memory footprint and improving inference speed [15, 17]. Instead of applying one precision level to the whole model, Mixed-Precision Quantization or

Activation-Aware Quantization methods could be used [17, 18, 19]. These methods assign higher precision to “sensitive” layers that impact accuracy the most, and lower the precision (e.g., 2-bit or 4-bit) to others. AWQ is a preferred method for SLMs because it protects the most important weights during quantization, maintaining at the same time a higher reasoning accuracy than the older methods like GPTQ. Other techniques like Post-Training Quantization (PTQ) or Quantization-Aware Training (QAT) are also commonly used [20].

The model size and computational complexity could be substantially reduced by pruning, the removal of redundant or less important connections in the neural network [21]. Structured pruning is often preferred for more effective hardware acceleration. Rather than removing individual weights, which can be expensive for hardware to optimize, structured pruning removes entire blocks or layers [22]. Algorithms like SparseGPT [23] and Wanda [24] are usually used to prune models after training, allowing a model to retain its “intelligence” while shrinking its footprint by 20–50%.

Knowledge distillation, where a smaller “student” model mimics the behavior of a larger “teacher” model, allows the developed model to inherit the knowledge of the teacher model while maintaining a significantly smaller size [25]. Some models use iterative refinement [26, 27], a “self-distillation” procedure to “teach themselves”, to improve performance without requiring an external teacher model and a separate distillation procedure.

Low-rank factorization [28, 29] decomposes weight matrices into lower-rank approximations, reducing the number of parameters, and hardware-aware optimization tailors model architectures and optimization techniques to the specific hardware capabilities of the edge device.

Speculative decoding (SD) is another inference acceleration technique that uses a tiny “draft” model alongside the primary SLM [30]. The tiny draft model quickly predicts several future tokens. The primary SLM verifies then these tokens in a single parallel step, rather than generating them one by one. Frameworks like SLED (Speculative LLM Decoding) [31] can increase system throughput by up to 2.8x on popular SMB hardware like Raspberry Pi 5 or NVIDIA Jetson.

These techniques are often combined into a single pipeline to fit models into memory-constrained environments. Modern SLMs are no longer just “shrunk” versions of their “big” counterparts, being rather designed specifically for the respective edge hardware (Hardware-Aware Architecture) [32].

5 Typical applications, use cases and benefits

Recently, Small Language Models (SLMs) and their applications have transitioned from experimental pilot designs to industry standards, with roughly 80% of AI inference now occurring locally on edge devices according to industry estimates [34]. Their ability to provide real-time, private, and offline intelligence resulted in diverse use cases.

In healthcare, portable and wearable devices use SLMs to analyze heart rhythms and lung sounds locally, alerting emergency services when a verified anomaly is detected [35]. Devices like digital stethoscopes transcribe sounds and flag potential abnor-

malities instantly, enabling clinical workflow acceleration. Direct on-device diagnostic [36, 37] eliminates concerns about patient data confidentiality by processing sensitive information locally within the hospital’s secure network.

In industrial environments, the use of Edge AI enables predictive maintenance, e.g. sensors can analyze vibrations and temperatures in real-time, predicting imminent equipment failures before they occur and reducing unplanned downtime significantly [38]. High-speed vision systems using local SLMs could be used for inspection on production lines in real time, ensuring immediate rejection of faulty items when necessary.

Millisecond-level navigation decisions are crucial in the case of autonomous vehicles [39]. Edge AI enables dynamic traffic management, where edge-native AI analyzes local sensor data at intersections to adjust signals in real-time, reducing congestion and emissions. Autonomous vehicles process terabytes of sensor data locally to support safety features, such as pedestrian detection and truck platooning, for which cloud-access-related latency could be fatal.

The environmental sector faces immense challenges, requiring constant monitoring, analysis, and informed decision-making. Traditional cloud-based AI solutions often struggle to meet the specific needs of environmental applications due to latency, bandwidth limitations, and power constraints. Edge AI, particularly leveraging SLMs, offers a substantially improved approach [40]. Acoustic monitoring in wildlife reserves, for instance, can utilize SLMs on low-power devices to identify species presence, or detect illegal activities without transmitting large audio files. Similarly, camera trap analysis can be performed locally, automatically identifying the animals and detecting unusual behaviors [41].

Precision agriculture could benefit from SLMs analyzing crop health in real-time, enabling targeted action and reducing pesticide use [42]. Air quality monitoring stations, equipped with sensors and SLMs, can analyze data and provide alerts when pollutant levels exceed safe thresholds. The ability to perform these analyses on-device eliminates the need for constant cloud connectivity and reduces power consumption, making Edge AI with SLMs a powerful tool for sustainable environmental management. Such devices may also be used for real-time monitoring of air and water quality, triggering instant alerts when spikes of hazardous concentrations are detected.

In the case of consumer electronics, SLMs enable “always-on” virtual assistance, that handle tasks like document summarization, email drafting, and real-time translation entirely offline [43]. Smart Home devices like thermostats can use local SLMs to learn user behavior and habits, optimizing energy usage without sending personal routines to the cloud. On a larger scale, SLMs could be also seen as enablers of locally tuned and democratically aligned intelligence that can better serve urban equity and efficiency goals [44].

By processing data at the source, Edge AI provides a “Privacy-by-Design” architecture [45], essential for meeting the data regulations in many countries. Sensitive information, such as medical records or financial transactions, never leaves the device, eliminating the risk of interception during cloud transit. Since data is processed locally, there is no centralized data center for hackers to target, significantly lowering the impact of large-scale breaches. Local processing by SLMs also eliminates the 50–200ms “round-trip” delay typical of cloud servers. Usually SLMs can deliver sub-5ms latency,

which is very important for time-sensitive applications. Additionally, the performance then remains stable, regardless of network congestion or server load.

Shifting tasks from cloud LLMs to local SLMs can reduce the inference costs by up to 90% (from ~\$0.50 in the cloud to ~\$0.05 on-device) [46, 47]. Local processing can also drastically reduce the volume of raw data sent over networks, lowering recurring telecommunication and data storage expenses. It could also ensure more reliability in environments where internet connectivity is impossible or unreliable, e.g. within coal mines.

Specialized SLMs could be optimized for standard hardware, allowing companies to scale AI without expensive and power hungry GPU clusters. Local inference reduces the massive energy demands and carbon footprints associated with running large-scale data centers. SLMs consume much less power than un-optimized much bigger models, also allowing battery-powered IoT devices to operate for days rather than hours. It is especially important in field operations, enabling robust performance in remote areas, e.g. for emergency responders in disaster zones, or industrial workers on factory floors with spotty Wi-Fi. Smart home devices and medical wearables could therefore continue to function during internet outages, maintaining essential safety and monitoring services.

6 Hardware platforms for edge AI with SLMs

While the advancements in software and in the algorithms for SLMs are undoubtedly noteworthy, it is essential to recognize that the hardware platforms supporting these technologies hold equal, if not greater, importance. This is particularly true given the rapid progression in hardware capabilities, which often outpaces significantly the improvements made in software [48]. The choice of a device depends on various factors such as computational requirements, power constraints, cost, and environmental conditions, with options ranging from very low-power microcontrollers to high-performance GPUs and specialized NPUs. The decentralized model of operation eliminates the necessity for continuous cloud connectivity, thereby providing advantages such as reduced latency, improved privacy, and decreased bandwidth expenses. Single-Board Computers (SBCs), hosting different CPUs and sometimes equipped with GPUs or NPUs, often serve as Edge AI platforms, facilitating the execution of artificial intelligence algorithms on compact and energy-efficient devices. Another choice could be repurposed Thin-Client Computers (TCCs), equipped often with surprisingly efficient hardware at a fraction of the cost of specialized industrial buildups. Refurbished older models typically cost only a fraction of the price of similar new equipment.

In resource-constrained embedded AI systems, microcontrollers (MCUs) are increasingly employed. Quite often they are enhanced with hardware accelerators such as dedicated Neural Processing Units (NPUs). These components enable the local execution of complex tasks, including vision and voice recognition. Unlike conventional MCUs, AI-optimized variants maintain a delicate balance between low power consumption and high computational capability, facilitating the operation of lightweight models such as TinyML [5]. Typically, MCUs exhibit a higher level of specialization, making

them more "tailored" for specific applications when compared to single-board computers (SBCs) or Thin-Client Computers (TCCs).

6.1 Edge AI SBCs

The Raspberry Pi SBCs [49], particularly the Raspberry Pi 4 and Raspberry Pi 5, together with the respective Computing Modules (CM), have emerged as a popular choice for Edge AI buildups due to their affordability, versatility, and extensive community support [50]. These devices can easily serve as a cornerstone for accessible Edge AI, ranging from simple hobbyist projects to platforms capable of handling generative AI workloads. It is worth mentioning that the dot product (DotProd) and half-precision floating point (FP16) arithmetic instructions, introduced in the ARMv8.2 CPU family ISAs [51], could provide the mathematical efficiency needed for modern transformers running e.g. on Raspberry Pi 5. The usage of FP16 allows the CPU to process twice as many data points per clock cycle when compared to FP32, while using the same SIMD (Single Instruction, Multiple Data) registers. Moreover, since FP16 weights are half the size of FP32, one can fit twice as many weights into the CPU cache, drastically reducing the time CPU spends waiting for data to be processed. Unlike 8-bit integers, FP16 maintains enough dynamic range to run models like Llama 3.2 or Gemma 3 with virtually zero accuracy loss when compared to their cloud-based versions. The DotProd instruction performs a 4-way 8-bit integer dot product and accumulates the result into a 32-bit integer in a single instruction. For quantized models (INT8), DotProd provides a theoretical 4x speedup over standard integer math. This is the primary reason why a Raspberry Pi 5 (ARMv8.2) is significantly faster than a Raspberry Pi 4 (ARMv8.0) for AI tasks at similar clock speeds. Because the hardware executes these operations in fewer cycles, it also reduces the "energy-per-token," which is especially important for battery-operated edge devices. For example, this allows the Raspberry Pi 5 to run 4-bit quantized SLMs like Phi-3.5 Mini at 10–15 tokens per second, making the difference between "laggy" text (~2 tps) and "real-time" text (10+ tps) generation.

While early SLM use on Raspberry Pi relied solely on the CPU, the platform can now utilize dedicated Neural Processing Units (NPUs) to achieve better performance. The newest Raspberry Pi AI HAT+ 2, launched in January 2026, features the Hailo-10H chip, delivering up to 40 TOPS of performance [52]. Unlike previous versions, the AI HAT+ 2 includes 8GB of dedicated RAM, allowing it to handle SLMs and Vision-Language Models (VLMs) independently of the main Raspberry Pi board. However, intensive SLM workloads would require an active cooler to prevent thermal throttling.

The NVIDIA Jetson Nano SBC offers a significant step up in processing power, particularly for GPU-accelerated tasks, though at a higher cost and power consumption [53, 54]. The NVIDIA Jetson Nano is considered a foundational, but nowadays legacy platform for edge AI. While it still offers valuable learning opportunities for computer vision, its limited memory and outdated software stack make it impractical for running modern SLMs effectively. Even when highly optimized and quantized, they require nowadays more memory for the model weights, tokenizer, and KV cache, often leading to system crashes or severe performance degradation (swap thrashing). The NVIDIA

Jetson Nano SBC uses a 128-core NVIDIA Maxwell™ GPU architecture from 2019. This architecture lacks the Tensor Cores found in newer generations (like Orin or Thor) that are essential for accelerating the matrix multiplication operations that dominate SLM inference. NVIDIA Jetson Orin Nano / Orin Nano Super are the official successors, offering significantly more AI performance and memory options (8GB/16GB). Through a “Super Mode” software update in JetPack 6.2, it delivers up to 67 TOPS (up from 40 TOPS) and a 1.7x performance boost for SLMs [55].

NVIDIA Jetson AGX Thor is the current high-end platform for advanced robotics, providing massive compute power for complex agentic AI workflows. Launched in late 2025/early 2026, it is now the premier platform for robotics and humanoid AI. Powered by the NVIDIA Blackwell™ GPU architecture, it delivers up to 2070 FP4 TFLOPS of AI compute over 7.5x the performance of the previous Orin generation [56], but at the cost of a disproportionate price increase.

6.2 Edge AI TCCs

Repurposing PC terminal hardware, including thin client computers (TCCs) and point-of-sale systems, is increasingly recognized as a sustainable and cost-effective alternative to single-board computers (SBCs) or specialized industrial computers [57]. These devices often feature surprisingly capable processors and substantial amounts of RAM, making them well-suited for Edge AI deployments. This approach not only extends the devices’ lifespan but also contributes to reducing electronic waste.

The trend of utilizing such devices as “Edge AI Gateways” has gained traction, particularly as they are commonly available in corporate surplus markets. Typically fanless and durable, these systems offer greater performance than microcontrollers while remaining more affordable than NVIDIA Jetson kits. For an extensive overview, including the devices indicated below, please refer to the ParkyTowers online service [58].

Thin clients from reputable brands such as HP (t-series), Dell (Wyse), or Lenovo (e.g., the ThinkCentre M-series) are exceptionally well-suited for the requirements of SLMs. It is advisable to opt for newer models that feature either the AMD Ryzen Embedded or Intel Elkhart Lake or newer chip families, as these processors incorporate advanced instruction sets, including AVX2, which significantly enhance the matrix calculations necessary for SLM applications. Unlike many Single Board Computers (SBCs), thin clients generally include SODIMM slots, allowing for straightforward expansion up to 16GB or even 32GB of RAM. This capability facilitates the execution of more substantial 7B or 14B models, such as Phi-4 or Llama 3.2 3B, which may otherwise fail to run on platforms like the Raspberry Pi 5. Additionally, thin clients typically offer a wider array of I/O interfaces compared to SBCs, further enhancing their versatility and functionality in professional settings.

For instance, the HP Pro t640 Thin Client is powered by the AMD Ryzen™ Embedded R1505G processor, which boasts 2 cores and 4 threads with a maximum boost frequency of 3.3 GHz, and is also equipped with Radeon™ Vega 3 Graphics. It offers exceptional performance, particularly when upgraded to 16GB or 32GB of RAM (DDR4). This compact, fanless device is designed for continuous operation, provid-

ing enterprise-grade durability. Furthermore, the default M.2 flash memory can be upgraded to a higher-capacity NVMe SSD, allowing for the storage of multiple models and vector databases. The integrated AMD Radeon™ Vega 3 Graphics also facilitates light GPU acceleration for AI applications, enhancing overall processing capabilities. Two of the four integrated USB ports support speeds up to 10 Gbps (USB 3.2 Gen 2).

The newer HP Pro t550 Thin Client from the same family features an Intel Celeron™ J6412 processor, a part of the Intel Elkhart Lake family, which includes four cores and four threads, capable of reaching a burst frequency of up to 2.6 GHz. This model is enhanced by the integration of Intel GNA 2.0 (Gaussian & Neural Accelerator), an ultra-low-power AI co-processor specifically designed for continuous, “always-on” background operations [59].

Distinct from traditional GPUs and CPUs, the GNA is optimized for low-precision integer arithmetic, rendering it highly efficient for particular Small Language Model components. In the context of the HP Pro t550, the GNA functions primarily as a “gatekeeper” or “pre-processor”, effectively conserving energy and CPU resources. It is capable of monitoring for specific trigger phrases or recognizing human speech patterns with minimal power consumption, utilizing microwatts. The more power-intensive Celeron™ CPU cores are then only activated when a valid command is identified.

Additionally, the GNA can serve as a neural filtering hardware device, adept at eliminating background noise, such as that generated by fans or traffic, to deliver a “clean” signal to the model, subsequently enhancing accuracy. Another potential application includes continuous biometric monitoring, where the GNA can execute compact neural networks for tasks like speaker identification or heart-rate analysis directly from sensors without placing undue stress on the main system. It is important to note that the GNA 2.0 requires models to be quantized to INT8 format. The Intel OpenVINO™ Neural Network Compression Framework (NNCF) can be utilized to convert models like Phi-3 Mini or TinyLlama into this optimal format [60].

6.3 Edge AI MCUs

Microcontrollers have emerged as essential components in the realm of Edge AI, facilitating the execution of artificial intelligence algorithms on low-power devices [5]. This capability for localized processing enables immediate decision-making with minimal latency, enhancing user privacy and lowering energy consumption in comparison to traditional cloud-based solutions.

In practical applications, these devices are proficient in managing specialized functions such as predictive maintenance. For example, by analyzing data related to vibrations or temperature, they can identify potential equipment failures in industrial settings before they escalate. Furthermore, microcontrollers support computer vision applications, enabling gesture recognition and defect detection through the utilization of optimized models like MobileNet [62]. In the consumer sector, they are an integral part of voice-activated keyword spotting in smart home devices and the monitoring of vital signs in healthcare wearables, which can detect various irregularities, such as heart arrhythmias. Agriculture also reaps the benefits of these technologies, as microcontrollers

facilitate the monitoring of soil quality and the identification of pests via field-deployed cameras.

The hardware supporting these diverse applications varies based on performance requirements and also budget considerations. Products like the STMicro STM32 series are designed for high-performance image processing and demanding machine learning tasks, typically incorporating dedicated hardware accelerators [63]. For more economically sensitive IoT initiatives, the ESP32-S3 [64] provides AI-extended instructions at a more accessible price point, while Texas Instruments (TI) chips prioritize low-latency performance for high-speed controls [65].

To effectively bridge the divide between AI models and MCU hardware, specialized development tools are essential. Frameworks such as LiteRT for Microcontrollers [67], known also as TensorFlow Lite for Microcontrollers, empower developers to compress and quantize models, ensuring compatibility with the limited memory constraints of microcontrollers. Additionally, manufacturer-specific utilities such as the STM32Cube AI Studio [66], facilitate the direct conversion of neural networks into optimized C code, simplifying the deployment of advanced intelligence even on the most compact hardware setups.

When choosing the right hardware for AI on microcontrollers, the decision largely depends on the specific requirements of the project. Factors such as the need for high-performance vision capabilities, integrated wireless connectivity, or ultra-low power consumption for extended battery life usually play pivotal roles here.

For tasks demanding intensive processing, such as real-time image recognition and video processing, the STMicroelectronics STM32N6 [68] stands out as a premier option due to its dedicated Neural-ART Accelerator, with an impressive 600 GOPS of computing power. Likewise, the Renesas RA8M85, equipped with a Cortex-M85 core and Helium extensions, efficiently manages complex mathematical operations inherent to machine learning, surpassing older conventional designs [69]. For projects with a particular focus on gesture recognition or low-power vision applications, the Grove Vision AI Module V2, which integrates an ARM Cortex M55 alongside a micro NPU [70], excels in delivering high frame rates while consuming minimal power.

In the case of applications to smart home technology or mutually connected IoT devices, the Espressif ESP32-S3 presents a commendable balance of cost-effectiveness and AI vector instructions, making it a preferred choice within the Arduino community. More specialized IoT solutions, such as the Silicon Labs EFR32MG24, come equipped with an integrated Matrix Vector Processor tailored for AI workloads and seamless compatibility with modern smart home protocols like Matter [71]. For high-performance wearables and audio processing applications, the Nordic nRF54 Series [72] offers a robust multi-core architecture capable of handling demanding AI tasks without sacrificing stable Bluetooth connectivity.

In scenarios where prototyping or the development of battery-operated industrial sensors is required, the Raspberry Pi RP2350 MCU [73], utilized in the Pico 2 board, provides extensive community support and a low-cost entry point for exploring TinyML. For more specialized industrial or medical applications where devices need to function for years on a single charge, the STM32U5 Series remains the benchmark for ultra-low power consumption while delivering the necessary performance for fundamental anomaly detection.

The STM32 family and the Raspberry Pi Pico series MCUs exemplify two distinct tiers of embedded AI development. The STM32 ecosystem is expansive, encompassing a range of products from low-power microcontrollers to high-performance processors such as the STM32H7. Many of these higher-end models are equipped with dedicated Neural Processing Units and hardware floating-point units designed specifically to enhance the performance of complex inference tasks. ST Microelectronics complements this robust hardware with advanced software tools, that automate the conversion of standard AI models into optimized code suitable for production-grade applications.

In contrast, the Raspberry Pi Pico MCU series is aimed at creators and educational audiences, focusing on simplicity and affordability. The original Pico's capabilities are somewhat limited for AI applications, primarily due to its RP2040 chip lacking a hardware floating-point unit. As a result, it must rely on slower software emulation for math-intensive AI tasks. However, the introduction of the RP2350 MCU chip in the newer Pico 2 board significantly addresses this limitation by incorporating hardware floating-point support and digital signal processing instructions, allowing for much more efficient handling of machine learning tasks compared to its predecessor.

In summary, professionals engaged in high-speed vision or industrial sensor development will find the STM32 to be a more suitable choice due to its extensive hardware scalability and comprehensive development tools. Conversely, the Pico and Pico 2 serve as excellent platforms for rapid prototyping, hobbyist projects, and educational contexts, where ease of use through MicroPython and cost-effectiveness are prioritized over maximum processing performance.

6.4 Edge AI smartphones

Last but not least, the market is witnessing a growing availability of AI-native smartphones [61], intended for the consumer sector. These devices have evolved from traditional flagship models with supplementary smart features to those where artificial intelligence serves as the core architecture. This transformation is marked by the seamless integration of specialized hardware, including Neural Processing Units (NPUs), which enable sophisticated generative models to operate directly on the device. By processing data locally at the "edge", rather than depending on cloud servers, these smartphones deliver significantly enhanced performance and improved privacy. This capability also facilitates offline functionality, proving invaluable in situations where network connectivity is limited or unavailable.

7 Future directions

The future of Edge AI is defined by the ongoing shift from “connected devices” to “autonomous devices”. The convergence of 2-bit quantization, specialized NPU hardware, and agentic workflows is gradually moving the majority of AI workloads from massive data centers to the palm of the user’s hand. Recent breakthroughs allow SLMs to run with significantly less memory. This makes it possible to run a high-reasoning parameter model on a device with little RAM, like a smartphone or a mid-range IoT gateway. These hyper-efficient models allow for “always-on” AI that consumes less power than a standard LED bulb, enabling solar-powered environmental sensors to “think” for months without needing a recharge.

It has to be emphasized that computer architecture related issues play a crucial role in the effectiveness of Edge AI, by harmonizing rapid processing capabilities with the inherent limitations of local hardware. Given that, these devices do not possess the extensive resources available to cloud servers, but only specialized components such as Neural Processing Units (NPUs) or Application-Specific Integrated Circuits (ASICs) that are designed to perform complex AI computations in parallel. This design significantly reduces latency and enables real-time decision-making in applications such as robotics and medical sensors.

In cases when energy efficiency is paramount, architectural innovations like Computation-in-Memory (CiM) technology [74] could lower power consumption, by decreasing the energy-intensive transfer of data between processors and memory, which enhances the battery longevity in portable electronics. In addition to speed and power considerations, these architectural designs address the “memory wall” challenge by incorporating localized caches and high-bandwidth memory solutions that facilitate seamless data transfer without overloading the device. This emphasis on local execution also provides a security advantage, safeguarding sensitive information from potential exposure, by ensuring it remains on the device, thereby mitigating vulnerabilities to external cyber threats. Contemporary trends, such as heterogeneous computing and the adoption of flexible instruction sets like RISC-V [75], empower engineers to customize hardware for specific applications. This ensures that whether the device is a smart camera or an industrial sensor, its architecture would be optimized to meet the demands of its specific AI workload.

The primary benefit of RISC-V in the artificial intelligence sector is its modular instruction set architecture. This design empowers developers to eliminate unnecessary complexity and incorporate custom instructions specifically optimized for tensor operations and matrix multiplication. Given that modern AI workloads demand substantial resources, traditional fixed architectures often result in power wastage due to general-purpose features that do not contribute to deep learning. RISC-V addresses this inefficiency through a “building blocks” approach, allowing developers to combine a fundamental integer set with the RISC-V Vector Extension to achieve remarkable parallel throughput, all without the licensing limitations and stringent design constraints inherent in proprietary solutions.

In high-performance data centers, RISC-V frequently functions as the management layer within expansive AI accelerators, coordinating data flow between memory and

specialized computing units. Companies such as NVIDIA leverage these cores to execute complex scheduling tasks, ensuring that their primary processing units remain fully operational. Concurrently, at the edge of the network, RISC-V is slowly becoming the standard for efficient, low-power inference in devices like smart sensors and wearables. In these applications, integrating specific AI algorithms directly into the silicon delivers real-time image recognition and voice processing capabilities while maintaining a minimal power budget that larger architectures struggle to match.

The transition toward RISC-V is also driven by a quest for architectural independence, enabling global technology firms and research institutions to innovate in AI hardware development without dependence on a single vendor's direction. This shift has resulted in a rapidly growing ecosystem, in which software frameworks like TensorFlow Lite or PyTorch are being fine-tuned also for RISC-V instructions. Nowadays, we are witnessing the advent of the first RISC-V AI PCs, designed to compete with established players by offering comparable tops-per-watt performance for executing large language models locally.

It is worth mentioning that also the dataflow architecture proposed more than fifty years ago by Jack Dennis [76] seems to be a significant advancement in hardware designs intended to accelerate AI workloads. This architecture is characterized by computations that are initiated by the availability of data rather than a predetermined sequence of instructions [77]. In contrast to traditional processors that operate according to a linear program counter, dataflow systems “conceptualize” in a sense the AI models as directed graphs in which operations are executed automatically upon the arrival of the respective inputs. This architecture provides remarkable efficiency, particularly for neural networks, as it facilitates the simultaneous processing of data across multiple layers of a model through extensive parallelization. By maintaining a continuous flow of data through a pre-configured pipeline, this approach effectively mitigates the performance bottlenecks and excessive energy consumption associated with the well-known “memory wall” phenomenon prevalent in conventional CPUs and GPUs. The Hailo AI accelerators mentioned above in the context of Raspberry Pi SBCs are the prominent real-world implementation of a successful structure-driven dataflow architecture designed specifically for Edge AI [78, 79].

The revival of analog computing [80, 81] could provide a critical evolution in hardware designed to overcome the energy and data-transfer limitations of traditional digital systems [82]. Unlike standard digital processors, which must constantly move information between separate memory and logic units, analog devices utilize analog signals related to physical phenomena which are used to model the problem being solved. This approach effectively eliminates the “von Neumann bottleneck”, allowing for the processing of artificial intelligence tasks with much better energy efficiency and significantly lower latency than conventional GPUs or CPUs. It is also possible to use various metamaterials to build analog computing devices [84], even as reconfigurable metastructures able to perform complex calculations [85]. While digital systems remain superior for high-precision tasks, modern hybrid analog/digital AI chips have reached a level of accuracy sufficient for complex pattern recognition [83], making them a cornerstone of the decentralized AI movement where 80% of inference is projected to happen locally on devices. This shift not only promises to preserve battery life but also enhances

data privacy by ensuring that sensitive information is processed at the source rather than being transmitted to the cloud.

While significant progress has been made, several challenges and opportunities still remain. Improving SLM performance through novel architectures and training techniques is paramount, as is developing federated learning approaches for training SLMs collaboratively across multiple edge devices while preserving privacy [86]. As privacy concerns grow, the way AI models are trained is changing. Instead of sending user data to the cloud, devices perform “on-device training” to personalize the model. Then the device sends only the mathematical updates (gradients) back to improve the global model, ensuring Zero-Knowledge Privacy.

Running language models on microcontrollers necessitates a focus on highly compressed TinyML architectures instead of general-purpose SLMs. The most effective strategy seems to be here to utilize hardware specialized models, like the STM32 Model Zoo [87], or develop custom models using frameworks like TensorFlow Lite, which can subsequently be converted into optimized C code via specialized toolchains, like the STM32Cube.AI toolchain in the case of STM32 hardware. For applications involving voice and text processing, developers often employ grammar-based models or specialized audio event detection families that are suitable for the constrained SRAM available on these chips. High-performance models, such as the STM32H7 series, are usually favored for their speed here, while the more recent STM32N6 series features a dedicated hardware NPU that enhances the efficiency of neural network execution. Additionally, tools like NanoEdge AI Studio [88] facilitate on-device learning and anomaly detection, empowering users without requiring extensive data science expertise.

The future also involves devices talking to each other to solve problems. If an edge device doesn't have the compute power for a complex task, it can “lease” NPU cycles from a nearby AI PC or a smart vehicle via ultra-low-latency 6G or Wi-Fi 7 connections. Groups of federated devices can then share their local SLM insights to form a high-resolution “global view” or “collaborative edge”. It enables “swarm sensing” by a group of sensors covering a large area - both from above (drones) and from stationary installed sensors, and then to develop a “swarm intelligence” to deal e.g. with problems in a massive industrial complex [89]. Allowing SLMs to adapt to new data and tasks without forgetting previously learned knowledge represents also a significant frontier.

Explainable AI (XAI) for Edge SLMs is crucial for fostering trust and accountability, and hardware-software co-design can further improve performance and energy efficiency [90]. Nowadays, AI is having a tremendous impact on many aspects of our life, including healthcare and engineering, where intelligent systems cannot be considered as “black boxes” or “infallible oracles”. Explainability could be also treated as an edge-AI system service [91], in contrast to being a specific model property implemented locally.

8 Conclusions

Edge AI powered by SLMs represents a transformative shift in the landscape of artificial intelligence. By bringing language understanding and generation capabilities directly to the edge, we can unlock a new era of personalized, real-time, and privacy-preserving applications. The potential for sustainable environmental monitoring and action, coupled with the growing feasibility of repurposing existing hardware, underscores the immense promise of this field. While challenges remain, ongoing research and development efforts are steadily pushing the boundaries of what is possible, paving the way for a future where intelligent devices are seamlessly integrated into our daily lives.

References

1. Domínguez-Bolaño, T., Campos, O., Barral, V., Escudero, C. J., & García-Naya, J. A. (2022). An overview of IoT architectures, technologies, and existing open-source projects. *Internet of Things*, 20, Article 100626. <https://doi.org/10.1016/j.iot.2022.100626>
2. Gauttam, H., Nain, G., Pattanaik, K. K., & Mendes, P. (2026). Edge-AI: A systematic review on architectures, applications, and challenges. *Journal of Network and Computer Applications*, 245, Article 104375. <https://doi.org/10.1016/j.jnca.2025.104375>
3. Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2025). A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5), Article 106. <https://doi.org/10.1145/3744746>
4. Wang, F., Zhang, Z., Zhang, X., Wu, Z., Mo, T., Lu, Q., Wang, W., Li, R., Xu, J., Tang, X., He, Q., Ma, Y., Huang, M., & Wang, S. (2025). A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with LLMs, and trustworthiness. *ACM Transactions on Intelligent Systems and Technology*, 16(6), Article 145. <https://doi.org/10.1145/3768165>
5. Tsoukas, V., Gkogkidis, A., Boumba, E., & Kakarountas, A. (2024). A review on the emerging technology of TinyML. *ACM Computing Surveys*, 56(10), Article 259. <https://doi.org/10.1145/3661820>
6. OpenAI. (2026). *Open models by OpenAI* [Website]. Retrieved February 15, 2026, from <https://openai.com/open-models/>
7. Gemma Team. (2025). Gemma 3 technical report. arXiv. <https://arxiv.org/abs/2503.19786>
8. Microsoft. (2024). *Phi-4 technical report*. Microsoft Research. <https://www.microsoft.com/en-us/research/wp-content/uploads/2024/12/P4TechReport.pdf>
9. Microsoft. (2025). Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-LoRAs. arXiv. <https://arxiv.org/abs/2503.01743>
10. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Srivankumar, A., Korenev, A., Hinsvark, A., ... Schrijver, R. (2024). The Llama 3 herd of models. arXiv. <https://arxiv.org/abs/2407.21783>
11. Qwen Team. (2025). Qwen2.5 technical report. arXiv. <https://arxiv.org/abs/2412.15115>
12. Zhang, P., Zeng, G., Wang, T., & Lu, W. (2024). TinyLlama: An open-source small language model. arXiv. <https://arxiv.org/abs/2401.02385>
13. Aralimatti, R., Shakhadri, S. A. G., Kruthika, K. R., & Angadi, K. B. (2025). Fine-tuning small language models for domain-specific AI: An edge AI perspective. In K. Arai (Ed.), *Intelligent systems and applications: Proceedings of the IntelliSys 2025 conference* (Lecture Notes in Networks and Systems, Vol. 1554, pp. [xx-xx]). Springer. https://doi.org/10.1007/978-3-031-99965-9_31

14. Allal, L. B., Lozhkov, A., Bakouch, E., von Werra, L., & Wolf, T. (2025). SmolLM2: When smol goes big – data-centric training of a small language model. arXiv. <https://arxiv.org/abs/2502.02737>
15. Li, S., Nguyen, H., Zheng, B., Nguyen, H.-T., Yao, Y., Zhou, Y., Qin, Z., Zhang, H., Han, X., Hu, S., Chen, W., & [remaining authors]. (2024). Evaluating quantized large language models. In *Proceedings of the 41st International Conference on Machine Learning* (Vol. 235, pp. 28480–28524). JMLR.org. <https://doi.org/10.5555/3692070.3693214>
16. Wang, Y., Huang, L., Zhang, J., & [remaining authors]. (2024). Art and science of quantizing large-scale models: A comprehensive overview. arXiv. <https://arxiv.org/abs/2409.11650>
17. Rakka, M., Fouda, M. E., Khargonekar, P., & Kurdahi, F. (2024). A review of state-of-the-art mixed-precision neural network frameworks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12), 7793–7812. <https://doi.org/10.1109/TPAMI.2024.3394390>
18. Qin, T., Luo, J., Cheng, C., & [remaining authors]. (2025). Mixed-precision quantization based on information entropy. *Scientific Reports*, 15, Article 12974. <https://doi.org/10.1038/s41598-025-91684-8>
19. Lin, J., Tang, J., Tang, H., Yang, S., Xiao, G., & Han, S. (2025). AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. *GetMobile: Mobile Computing and Communications*, 28(4), 12–17. <https://doi.org/10.1145/3714983.3714987>
20. Zhao, X., Xu, R., & Guo, X. (2023). Post-training quantization or quantization-aware training? That is the question. In *Proceedings of the 2023 China Semiconductor Technology International Conference (CSTIC)* (pp. 1–3). IEEE. <https://doi.org/10.1109/CSTIC58779.2023.10219214>
21. Hou, B., Wu, Q., Hao, Y., & [remaining authors]. (2025). Instruction-following pruning for large language models. arXiv. <https://arxiv.org/abs/2501.02086>
22. Guo, J., Chen, X., Tang, Y., & Wang, Y. (2025). SlimLLM: Accurate structured pruning for large language models. arXiv. <https://arxiv.org/abs/2505.22689>
23. Frantar, E., & Alistarh, D. (2023). SparseGPT: Massive language models can be accurately pruned in one shot. In *Proceedings of the 40th International Conference on Machine Learning* (Vol. 202, pp. 10323–10337). JMLR.org. <https://doi.org/10.5555/3618408.3618822>
24. Yu, P., Wang, J., Sui, X., Ling, N., Wang, W., & Jiang, W. (2026). Efficient post-training pruning of large language models with statistical correction. arXiv. <https://arxiv.org/abs/2602.07375>
25. Moslemi, A., Briskina, A., Dang, Z., & Li, J. (2024). A survey on knowledge distillation: Recent advancements. *Machine Learning with Applications*, 18, Article 100605. <https://doi.org/10.1016/j.mlwa.2024.100605>
26. Zhang, L., Bao, C., & Ma, K. (2022). Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8), 4388–4403. <https://doi.org/10.1109/TPAMI.2021.3067100>
27. Shenfeld, I., Damani, M., Hübötter, J., & Agrawal, P. (2026). Self-distillation enables continual learning. arXiv. <https://arxiv.org/abs/2601.19897>
28. Sainath, T. N., Kingsbury, B., Sindhvani, V., Arisoy, E., & Ramabhadran, B. (2013). Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)* (pp. 6655–6659). IEEE. <https://doi.org/10.1109/ICASSP.2013.6638949>
29. Hsu, Y.-C., Ting, H., Chang, S., Lou, Q., Shen, Y., & Jin, H. (2022). Language model compression with weighted low-rank factorization. arXiv. <https://arxiv.org/abs/2207.00112>
30. Leviathan, Y., Kalman, M., & Matias, Y. (2023). Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning* (Vol. 202, pp. 19274–19286). JMLR.org. <https://doi.org/10.5555/3618408.3619203>
31. Yan, M., Agarwal, S., & Venkataraman, S. (2025). Decoding speculative decoding. arXiv. <https://arxiv.org/abs/2402.01528>
32. Marculescu, D., Stamoulis, D., & Cai, E. (2018). Hardware-aware machine learning: Modeling and optimization. In *Proceedings of the 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD '18)* (pp. 1–8). ACM. <https://doi.org/10.1145/3240765.3243479>

33. Rhea, S., Dohan, D., Firat, O., & [remaining authors]. (2024). HW-GPT-Bench: Hardware-aware architecture benchmark for language models. *Advances in Neural Information Processing Systems*, 37, 60776–60834. <https://doi.org/10.52202/079017-1944>
34. Korolov, M. (2026, January 8). CES 2026: AI compute sees a shift from training to inference. *Computerworld*. Retrieved February 15, 2026, from <https://www.computerworld.com/article/4114579/ces-2026-ai-compute-sees-a-shift-from-training-to-inference.html>
35. Gupta, S., & Chaudhary, A. (Eds.). (2026). *Artificial intelligence in healthcare: Trends, applications, and future directions*. Apple Academic Press.
36. Nguyen, M. H., Shen, Y., Liao, J., & [remaining authors]. (2025). On-device diagnostic recommendation with heterogeneous federated BlockNets. *Science China Information Sciences*, 68, Article 140102. <https://doi.org/10.1007/s11432-024-4162-2>
37. Villalobos-Quesada, M., Ho, K., Chavannes, N. H., & Talboom-Kamp, E. P. (2023). Direct-to-patient digital diagnostics in primary care: Opportunities, challenges, and conditions necessary for responsible digital diagnostics. *European Journal of General Practice*, 29(1), Article 2273615. <https://doi.org/10.1080/13814788.2023.2273615>
38. Artiushenko, V., Lang, S., Lerez, C., Reggelin, T., & Hackert-Oschätzchen, M. (2024). Resource-efficient edge AI solution for predictive maintenance. *Procedia Computer Science*, 232, 348–357. <https://doi.org/10.1016/j.procs.2024.01.034>
39. Uprety, I., & Zhao, X. (2025). Edge-deployable LLMs for autonomous vehicle intelligence. In *Proceedings of the Tenth ACM/IEEE Symposium on Edge Computing (SEC '25)* (pp. 1–7). ACM. <https://doi.org/10.1145/3769102.3774639>
40. Rahman, M. A., Dewan, M. A. A., Hasan, M., & [remaining authors]. (2025). A scalable framework for deploying AI-powered wildlife monitoring in resource-limited field environments. *IEEE Access*, 13, 145023–145041. <https://doi.org/10.1109/ACCESS.2025.3598927>
41. Boscoe, B., Johnson, S., Osborn, A., Campbell, C., & Mager, K. (2025). GreenCrossingAI: A camera-trap/computer-vision pipeline for environmental science research groups. In *Proceedings of the Practice and Experience in Advanced Research Computing 2025 (PEARC '25)* (pp. 1–8). ACM. <https://doi.org/10.1145/3708035.3736003>
42. Chen, S., Li, D., Liu, J., & Wei, R. (2024). Raspberry Pi-based intelligent greenhouse IoT platform. In *Proceedings of the 2023 5th International Conference on Internet of Things, Automation and Artificial Intelligence (IoTAAI '23)* (pp. 787–791). ACM. <https://doi.org/10.1145/3653081.3653213>
43. Dai, X., & Yao, W. (2025). Research on development strategies for edge-AI-based smart-home devices. In *Proceedings of the 2nd Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence (DEAI '25)* (pp. 1474–1481). ACM. <https://doi.org/10.1145/3745238.3745469>
44. Tiwari, A. (2026). A proposal to localise urban AI: A conceptual shift from generalist LLMs to task-specific SLMs. *Computational Urban Science*, 6, Article 11. <https://doi.org/10.1007/s43762-026-00241-0>
45. Sachdev, R. (2020). Towards security and privacy for edge AI in IoT/IoE-based digital marketing environments. In *Proceedings of the 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC 2020)* (pp. 341–346). IEEE. <https://doi.org/10.1109/FMEC49853.2020.9144755>
46. Mathai, A. (2026, January 17). Cut AI costs by 90%: Why smart companies are downsizing to small language models (SLMs). *Mindster*. Retrieved February 15, 2026, from <https://mindster.com/mindster-blogs/small-language-models-slm-cost-efficiency/>
47. Dong, Z., Sharma, H., O’Toole, E., Champati, J. P., & Wu, K. (2026). Pay for hints, not answers: LLM shepherding for cost-efficient inference. arXiv. <https://arxiv.org/abs/2601.22132>
48. Scott, K. (2001). *On Proebsting’s law* (Technical Report). University of Virginia, Department of Computer Science. <https://doi.org/10.18130/V33Z0W>
49. Raspberry Pi Ltd. (2026). *Raspberry Pi computer hardware*. Retrieved February 15, 2026, from <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>
50. Włodarz, J. (2024). Computer chips and social economy: The impact of affordable computing. In Z. Wittine, S. Franc, & A. Barišić (Eds.), *International Scientific Conference “Empowering*

- Change: Fostering Social Entrepreneurship for a Sustainable Future*” (pp. 50–56). University of Zagreb. https://doi.org/10.1007/978-3-031-12345-6_6
51. ARM Ltd. (2025, December). *The Armv8.2 architecture extension*. ARM Developer. Retrieved February 15, 2026, from https://developer.arm.com/documentation/109697/2025_12/Feature-descriptions/The-Armv8-2-architecture-extension
 52. Raspberry Pi Ltd. (2026). *Raspberry Pi AI HAT+ 2 product brief*. Retrieved February 15, 2026, from <https://pip-assets.raspberrypi.com/categories/1319-raspberry-pi-ai-hat-2/documents/RP-009655-MM-4-raspberry-pi-ai-hat-plus-2-product-brief.pdf>
 53. Chavan, S. R., Gavande, P., & Mhaske, M. D. (2025). A comprehensive review of Nvidia Jetson Nano module. *International Research Journal of Modernization in Engineering Technology and Science*, 7(3), 1378–1386. <https://doi.org/10.56726/IRJMETS84923>
 54. Maslekar, A., Suryavanshi, A., & Gavande, P. (2025). Catalyst for intelligent edge: A comprehensive analysis of the Nvidia Jetson Nano – architecture, performance, benchmarking and comparative standing in the AIoT landscape. *International Research Journal of Modernization in Engineering Technology and Science*, 7(10), 2999–3008. <https://doi.org/10.56726/IRJMETS84050>
 55. Maheshwari, S., & Su, C. (2025, January 16). NVIDIA JetPack 6.2 brings super mode to NVIDIA Jetson Orin Nano and Jetson Orin NX modules. *NVIDIA Developer Blog*. Retrieved February 15, 2026, from <https://developer.nvidia.com/blog/nvidia-jetpack-6-2-brings-super-mode-to-nvidia-jetson-orin-nano-and-jetson-orin-nx-modules/>
 56. NVIDIA Corp. (2026). *NVIDIA Jetson Thor: The ultimate platform for physical AI and robotics*. Retrieved February 15, 2026, from <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-thor/>
 57. Caruso, J. E. (2025, December 10). AI at the edge – Intelligent systems operating where data is generated. *Syslog Technical Articles*. Retrieved February 15, 2026, from <https://www.syslog.com/blog/ai-at-the-edge>
 58. ParkyTowers. (2026). *Thin clients*. Retrieved February 15, 2026, from <https://www.parkytowers.me.uk/thin/>
 59. Fatkina, A., Kozlov, A., Shevelev, A., & [remaining authors]. (2019). GNA: New framework for statistical data analysis. *EPJ Web of Conferences*, 214, Article 05024. <https://doi.org/10.1051/epjconf/201921405024>
 60. Intel Corp. (2026). *OpenVINO toolkit documentation*. Retrieved February 15, 2026, from <https://docs.openvino.ai>
 61. Hirsch, M., Mateos, C., & Majchrzak, T. A. (2025). Exploring smartphone-based edge AI inferences using real testbeds. *Sensors*, 25(9), Article 2875. <https://doi.org/10.3390/s25092875>
 62. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv. <https://arxiv.org/abs/1704.04861>
 63. Hentati, M., Hentati, R., Gouiaa, Y., & Abid, M. (2025). Edge AI-based classification: A case study on STM32 microcontroller. In *Proceedings of the 2025 International Conference on Advanced Systems and Emergent Technologies (IC_ASET 2025)* (pp. 1–6). IEEE. https://doi.org/10.1109/IC_ASET65966.2025.11231782
 64. Espressif Systems Co., Ltd. (2026). *ESP32-S3 technical reference manual*. Retrieved February 15, 2026, from https://www.espressif.com/documentation/esp32-s3_technical_reference_manual_en.pdf
 65. May, A. (2026). *Increasing intelligence at the edge with embedded processors* [White paper]. Texas Instruments. Retrieved February 15, 2026, from <https://www.ti.com/lit/wp/spry349a/spry349a.pdf>
 66. STMicroelectronics NV. (2022). *STM32Cube ecosystem overview: Making STM32 development easier* [Product presentation]. Retrieved February 15, 2026, from https://www.st.com/resource/en/product_presentation/stm32cube_ecosystem_overview.pdf
 67. Google AI for Developers Community. (2026). *LiteRT for microcontrollers*. Retrieved February 15, 2026, from <https://ai.google.dev/edge/litert/microcontrollers/overview>

68. STMicroelectronics NV. (2026). *STM32N6x5xx STM32N6x7xx datasheet*. Retrieved February 15, 2026, from <https://www.st.com/resource/en/datasheet/stm32n657a0.pdf>
69. Renesas Electronics Co. (2023). *An introduction to Renesas Advanced (RA) MCU kits* [Presentation]. Retrieved February 15, 2026, from <https://www.renesas.com/en/document/ppt/introduction-renesas-advanced-ra-mcu-kits>
70. Seeed Studio, Inc. (2026). *Grove Vision AI module V2* [Product wiki]. Retrieved February 15, 2026, from https://wiki.seeedstudio.com/grove_vision_ai_v2/
71. Silicon Laboratories, Inc. (2026). *EFR32MG24 wireless SoC family datasheet*. Retrieved February 15, 2026, from <https://www.silabs.com/documents/public/data-sheets/efr32mg24-datasheet.pdf>
72. Nordic Semiconductor ASA. (2026). *nRF54L series technical documentation*. Retrieved February 15, 2026, from <https://docs.nordicsemi.com/category/nrf-54L-series>
73. Raspberry Pi Ltd. (2025). *RP2350 datasheet: A microcontroller by Raspberry Pi*. Retrieved February 15, 2026, from <https://pip-assets.raspberrypi.com/categories/1214-rp2350/documents/RP-008373-DS-2-rp2350-datasheet.pdf>
74. Sun, Z., Kvatinsky, S., Si, X., Alhaji, R., Kang, J., & [remaining authors]. (2023). A full spectrum of computing-in-memory technologies. *Nature Electronics*, 6, 823–835. <https://doi.org/10.1038/s41928-023-01053-4>
75. Borade, S. A., Bansod, S., Hati, A. J., & Singh, S. K. (2025). AI edge processor using RISC-V instruction set architecture design. In *Proceedings of the 2025 Global Conference in Emerging Technology (GINOTECH 2025)* (pp. 1–8). IEEE. <https://doi.org/10.1109/GINOTECH63460.2025.11076631>
76. Dennis, J. B. (1974). First version of a data flow procedure language. In G. Goos & J. Hartmanis (Eds.), *Programming symposium: Proceedings, colloque sur la programmation* (Lecture Notes in Computer Science, Vol. 19, pp. [xx–xx]). Springer.
77. Veen, A. H. (1986). Dataflow machine architecture. *ACM Computing Surveys*, 18(4), 365–396. <https://doi.org/10.1145/27633.28055>
78. Hailo Technologies Ltd. (2025). *Bringing generative AI to the edge: LLM on Hailo-10H*. Retrieved February 15, 2026, from <https://hailo.ai/blog/bringing-generative-ai-to-the-edge-llm-on-hailo-10h/>
79. Krispin-Avraham, I., Orfaig, R., & Bobrovsky, B.-Z. (2024). Real-time 3D object detection using InnovizOne LiDAR and low-power Hailo-8 AI accelerator. arXiv. <https://doi.org/10.48550/arXiv.2412.05594>
80. Ulmann, B. (2022). *Analog computing*. De Gruyter Oldenbourg. <https://doi.org/10.1515/9783110787740>
81. Ulmann, B. (2026). *Analog computing: Development, programming, applications, and future directions*. De Gruyter. <https://doi.org/10.1515/9783112210178>
82. Haensch, W., Gokmen, T., & Puri, R. (2019). The next generation of deep learning hardware: Analog computing. *Proceedings of the IEEE*, 107(1), 108–122. <https://doi.org/10.1109/JPROC.2018.2871057>
83. Guo, N., Huang, Y., Mai, T., & [remaining authors]. (2016). Energy-efficient hybrid analog/digital approximate computation in continuous time. *IEEE Journal of Solid-State Circuits*, 51(7), 1514–1524. <https://doi.org/10.1109/JSSC.2016.2543729>
84. Włodarz, J. (2019). Analog computing and (meta)materials. In *Proceedings of the 1st Polish-Chinese Conference “From Molecular Modeling to Nano- and Biotechnology”* (pp. 1–46).
85. Tzarouchis, D. C., Edwards, B., & Engheta, N. (2025). Programmable wave-based analog computing machine: A metastructure that designs metastructures. *Nature Communications*, 16, Article 908. <https://doi.org/10.1038/s41467-025-56019-1>
86. Wang, Z., Wu, F., Yu, F., Zhou, Y., Hu, J., & Min, G. (2024). Federated continual learning for edge-AI: A comprehensive survey. arXiv. <https://arxiv.org/abs/2411.13740>
87. STMicroelectronics NV. (2026). *AI model zoo for STM32 devices* [Software repository]. Retrieved February 15, 2026, from <https://github.com/STMicroelectronics/stm32ai-modelzoo>
88. STMicroelectronics NV. (2026). *NanoEdge AI Studio: Automated machine learning (ML) tool for STM32 developers*. Retrieved February 15, 2026, from <https://www.st.com/en/development-tools/nanoedgeaistudio.html>

89. Sah, D. K., Vahabi, M., & Fotouhi, H. (2025). Federated learning at the edge in industrial internet of things: A review. *Sustainable Computing: Informatics and Systems*, *46*, Article 101087. <https://doi.org/10.1016/j.suscom.2025.101087>
90. Górriz, J. M., Ramírez, J., Ortíz, A., & [remaining authors]. (2023). Computational approaches to explainable artificial intelligence: Advances in theory, applications and trends. *Information Fusion*, *100*, Article 101945. <https://doi.org/10.1016/j.inffus.2023.101945>
91. Singh, S. K., & Roy, J. (2026). Scalable explainability-as-a-service (XaaS) for edge AI systems. arXiv. <https://arxiv.org/abs/2602.04120>

About author

Joachim J. Włodarz has been active in academia since the early 1980s, primarily in the fields of quantum chemistry/physics and computer science. He is currently a university professor at the Faculty of Science and Technology, University of Silesia in Katowice, Poland.

Supervised Machine Learning for Renewable Energy

Dijana Oreški,
Vjeran Strahonja, and
Marija Pokos Lukinec

Abstract

Accurate renewable energy forecasting is important for optimizing grid integration and advancing environmental sustainability. This chapter develops predictive models based on supervised machine learning for solar energy consumption using historical data from solar power plants, integrating various data sources: historical energy consumption, actual weather conditions (including temperature, insolation, and wind speed), and historical weather forecasts. Advanced artificial intelligence and machine learning algorithms including deep learning were trained on a multi-source dataset to identify complex temporal patterns and weather-energy patterns. The models achieved high precision, demonstrating robustness against meteorological variability. Accurate predictive models enable utilities to reduce fossil-fuel-based reserve capacity, minimize grid inefficiencies, and enhance renewable energy utilization. For environmental sustainability, these models directly support decarbonization goals by enabling larger solar integration, reducing associated carbon emissions from backup generation, and promoting resource-efficient energy planning. By facilitating the reliable and efficient integration of solar power, this approach represents a small step toward achieving zero net emissions in the energy sector.

Keywords: renewable energy; power plant; machine learning; forecasting; environmental sustainability

Dijana Oreški
University of Zagreb, Faculty of Organization and Informatics, Pavlinska 2, 42000 Varaždin, Croatia
e-mail: dijana.oreski@foi.unizg.hr
Vjeran Strahonja
University of Zagreb, Faculty of Organization and Informatics, Pavlinska 2, 42000 Varaždin, Croatia
e-mail: vjeran.strahonja@foi.unizg.hr
Marija Pokos Lukinec
University of Zagreb, Faculty of Organization and Informatics, Pavlinska 2, 42000 Varaždin, Croatia
e-mail: mapokos@foi.hr

1 Introduction

The rapid global transition toward renewable energy sources is not only a strategic imperative for mitigating climate change and reducing greenhouse gas emissions but also a cornerstone for achieving long-term environmental and economic sustainability. Solar, wind, and hydro power technologies provide clean energy, yet their variability and uncertainty introduce operational challenges for modern power systems. Solar energy, in particular, has emerged as one of the fastest-growing and most promising resources due to its modularity, scalability, and declining costs. However, its inherently changeable nature - caused by meteorological fluctuations, and seasonal variability - complicates reliable grid integration and real-time operational planning. These challenges are further amplified by the accelerating global demand for electricity, driven not only by population growth and electrification of transport and industry but also by the proliferation of energy-intensive digital technologies such as large language models (LLMs) and AI-driven data centers. As the share of variable renewable energy sources rises, the ability to accurately forecast solar generation or consumption and optimize system flexibility becomes increasingly important for reducing reliance on fossil-fuel-based backup capacity, minimizing grid inefficiencies, and supporting the broader goal of deep decarbonization.

Accurate forecasting of solar consumption is therefore critical to ensure stable electricity supply, reduce reliance on fossil-fuel-based backup systems, and optimize resource allocation. Short-term and mid-term forecasts allow grid operators to better schedule generation, minimize balancing costs, and improve the efficiency of energy markets. However, accurate prediction is challenged by the fact that renewable energy sources are inherently dependent on external, often unpredictable, factors such as weather conditions. Similarly, energy consumption patterns are influenced by user behavior, time of day, seasonality, and socio-economic factors. Traditional forecasting methods often struggle to capture these nonlinear and dynamic relationships.

Recent advances in artificial intelligence (AI) and machine learning (ML) have opened new opportunities for developing highly accurate and adaptive forecasting models in the energy sector. Unlike traditional rule-based or statistical methods, which rely on fixed assumptions and limited analytical flexibility, ML algorithms can autonomously learn complex patterns from large volumes of historical data and continuously adapt to emerging trends. This makes them suitable for both energy generation and consumption prediction, where nonlinear interactions and temporal dependencies play a critical role. Deep learning architectures, for example, can capture intrinsic relationships among diverse variables such as solar irradiance, ambient temperature, wind speed, and historical consumption profiles, providing predictive capabilities far beyond those of conventional approaches [1]. Furthermore, the integration of multi-source datasets - combining measured solar output, real-time meteorological data, and weather forecast information - significantly improves model robustness against meteorological variability. By enabling data-driven decision-making for grid stability, load balancing, and energy efficiency, ML-based forecasting methods provide a foundation for optimizing renewable energy integration and advancing sustainability objectives.

This chapter develops and evaluates advanced predictive models based on machine and deep learning (DL) for solar energy consumption with the aim of improving grid reliability and supporting environmental sustainability objectives. By leveraging DL and other state-of-the-art ML techniques on a rich, multi-source data set, we demonstrate how powerful AI algorithms in combination with quality data can develop accurate predictive models.

This chapter is organized as follows. Section 2 reviews the relevant literature and theoretical background and presents the methodology, detailing the Long Short-Term Memory (LSTM) and Time Series Transformer (TST) models along with a description of the heterogeneous data sources. Section 3 discusses the research results, while Section 4 highlights their implications for integration into battery optimization systems, along with a discussion of financial and environmental impacts. Finally, Section 5 concludes the research and outlines potential directions for future work.

2 Materials and methods

This section outlines the study objectives, summarizes related work to position our approach within the existing literature, and describes the methodology and datasets used to develop and evaluate the predictive models.

2.1 Research objectives

The research problem is motivated by a real-life problem faced by the solar energy company which installs solar power plants. These plants rely on sunlight, which is highly variable and influenced by local weather patterns. The company's goal is to maximize electricity generation while ensuring a reliable supply to the power grid. One major challenge is accurately predicting how much energy will be produced each day or hour. Bad forecasting can lead to overproduction or underproduction, which affects grid stability and profitability. The company needs accurate short-term and long-term forecasts to plan operations and schedule maintenance effectively. Because solar power is not always available when demand peaks, companies also rely on battery storage systems. They must decide when to store energy and when to sell it back to the grid - in a dynamic pricing environment.

Motivated by this real-life problem, our research aims to:

1. Develop accurate models for predicting renewable energy consumption. The first objective is to build machine learning models that can forecast energy consumption from renewable sources. These models need to account for variables like weather, season, location, and historical output. We develop next-day and multi-period energy-consumption forecasts by training on plant-level historical datasets that integrate three information streams: (i) historical consumption profiles, (ii) realized meteorological conditions: air temperature, solar irradiance/insolation, and wind speed and

- (iii) historical weather forecasts for the same variables. Incorporating both realized weather and time-aligned forecast fields enables the models to learn from operationally available signals and to better approximate real-world forecasting conditions.
2. Evaluate and compare different machine learning algorithms. A variety of models can be used and this objective involves testing two algorithms to see which offers the best accuracy and generalization.

Ultimately, predictions should be actionable. The goal of such models is to use ML outputs to inform real-world decisions, such as when to charge/discharge batteries or how to best feed energy into the grid to reduce costs and improve system stability. This is discussed in discussion section.

2.2 Related works

The rapid growth of AI and ML applications in predictive modeling has resulted in an extensive and diverse body of research. Understanding the current state of knowledge requires examining not only the technical methods used but also the types of data on which these methods rely and the performance of the resulting models. This literature review is organized into three main areas. First, it investigates what data sources have been used to develop predictive models. Second, it examines what AI and ML algorithms have been applied, evaluating trends in algorithm selection and implementation. Finally, it analyzes the quality of the resulting predictive models, considering evaluation metrics. Together, these three perspectives provide an overview of how predictive modeling approaches have evolved and where research opportunities for improvement remain.

Recent literature highlights that diverse and multi-layered datasets are critical for developing accurate predictive models for solar energy forecasting and battery storage optimization [2]. Measurements such as irradiance, temperature, and energy production have shown the best performance. Models trained on these datasets achieve a low MAPE and high R^2 . Additional categories of data include weather sensor readings, smart grid data, and energy consumption/load profiles, which are often integrated into models for battery energy storage system (BESS) optimization [3, 4]. Existing studies, however, typically rely on a single type of data or, at best, integrate two distinct data sources. No work has been identified that combines more than two heterogeneous sources of information to build predictive models in this domain. This limitation reduces the potential to capture complex relationships and richer contextual factors that could enhance predictive accuracy and robustness. Addressing this research gap represents one of the motivations for the present study.

Several studies have demonstrated that machine learning techniques can effectively model photovoltaic (PV) energy generation by leveraging meteorological variables. For example, [5] explores how algorithms such as artificial neural networks, support vector machines, and regression models can predict PV output based on inputs like temperature, irradiance, humidity, and wind speed. The authors highlight that incorporating multiple weather parameters improves the accuracy of energy forecasting compared to

traditional empirical methods, supporting the adoption of data-driven approaches for optimizing PV system performance and grid integration. Additional literature demonstrates a wide range of machine learning, with recent papers emphasizing deep learning approaches superiority (e.g. LSTM and CNN-LSTM in [6]). Although a variety of machine learning algorithms have been applied, there remains a lack of comprehensive comparative analyses as well as insufficient exploration of the potential of generative models.

The quality of ML and DL models is typically assessed using both classical and advanced performance metrics such as RMSE, MAE, MAPE, and R^2 .

2.3 Data description

This study utilizes a comprehensive dataset comprising solar energy consumption data from multiple photovoltaic power plants equipped with different monitoring systems. The primary data sources include power plants from three major manufacturers: SolarEdge (12 power plants), Fusion Solar/Huawei (25 power plants), and iSolarCloud (3 power plants). Each monitoring system presents unique characteristics in terms of data accessibility, measurement intervals, and available metrics, necessitating a systematic approach to data standardization.

The dataset encompasses three main categories of variables: (1) historical energy consumption and production data from solar power plants, (2) actual meteorological conditions, and (3) weather forecast data. As such, this dataset integrates power plant operational data with relevant meteorological and temporal information. All power plant data, along with corresponding auxiliary datasets, are recorded and utilized as standardized 15-minute interval values. Within the database, each time interval is stored in the format YYYY-MM-DD HH:MM:SS (e.g., 2025-02-26 12:15:00). The temporal intervals range from 00:00:00 (the beginning of the day, representing the first interval for a given date) to 23:45:00 (the final interval of that date). Each timestamp is preceded by the corresponding calendar date, ensuring temporal consistency across all records.

The dataset comprises the following groups of features:

- Consumption – Represents the historical energy consumption aggregated from all available power plants.
- Actual weather conditions – Include measured meteorological parameters such as temperature, insolation, and wind speed.
- Weather forecasts – Encompass past, present, and future forecasted meteorological data. The observed parameters correspond to those in the actual weather conditions dataset, namely forecastTemperature, forecastInsolation, and forecastWindSpeed.
- Holiday – A binary indicator (0 or 1) specifying whether a particular day is a public holiday. A value of 0 denotes a regular (non-holiday) day, while 1 indicates a public holiday.
- TimeOfDay – Refers to 15-minute data retrieval intervals (timestamps), represented numerically and appropriately scaled or transformed.

- DayOfWeek – Numerical representation of the day of the week, suitably scaled or transformed.
- DayOfYear – Numerical representation of the day within the year, also scaled or transformed.
- Correction – A binary indicator specifying the validity of the consumption data obtained from power plants. If irregularities occur (e.g., missing or NULL values), such data are considered invalid. For these intervals, the correction flag is set to 1, and the consumption value is recorded as 0 instead of NULL. Conversely, if the consumption data are valid, the actual measured values are used, and the correction flag remains 0 for those intervals.

The process of preparing the dataset for model training was carried out in two main phases: (1) generation of the base dataset and (2) final data transformation and scaling.

In the initial phase, the raw data obtained from power plants and accompanying meteorological sources were preprocessed and merged into a unified base dataset. This phase included temporal partitioning of the data into consistent 15-minute intervals, ensuring complete temporal coverage and alignment of all features. Subsequently, feature engineering was applied to enhance the informativeness of the dataset. Based on the previously described raw variables, the script automatically generated a set of derived features designed to facilitate pattern recognition by the model. These newly created features capture temporal, seasonal, and weather-related dynamics, thereby improving the model's ability to generalize and detect relevant correlations. Before model training, additional transformations were applied to ensure numerical stability and comparability across features. A logarithmic transformation was performed to reduce skewness in variables with large variance and to stabilize the data distribution. Finally, a global scaling procedure was implemented to normalize all input variables to a common range suitable for the selected learning algorithms. After model inference, predicted values were transformed back to their original scale to allow for accurate interpretation and evaluation against real-world measurements.

2.4 Deep learning algorithms

Our investigation covered a broad spectrum of approaches, including: (i) machine learning algorithms, such as Random Forests, to establish baseline performance, (ii) convolutional neural networks (CNN, RCNN) for capturing local temporal patterns and feature dependencies, (iii) recurrent neural networks (RNN), particularly Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), for modeling sequential dependencies and (iv) transformer-based architectures, including the Time Series Transformer (TST), Informer, and specialized variants such as PatchTST with residual connections for enhanced long-range dependency modeling.

Among these, two algorithms (LSTM and TST), showed high level of performance and those algorithms are described below and applied in this research.

Long Short-Term Memory Networks (LSTM) are a class of recurrent neural networks specifically designed to overcome the limitations of traditional RNNs [7], partic-

ularly the vanishing gradient problems during long sequence training. By introducing gated mechanisms - input, forget, and output gates - LSTMs regulate the flow of information through a cell state, enabling them to capture both short- and long-term dependencies in sequential data. This makes LSTMs well-suited for time series forecasting tasks, where patterns may span multiple temporal scales [8]. In this study, LSTMs were employed with varying numbers of layers and hidden units, as well as different regularization schemes, to assess their ability to generalize across complex temporal dynamics.

The Time Series Transformer (TST) adapts the self-attention mechanism of Transformer architectures [9], originally developed for natural language processing, to handle time series data [10]. Unlike recurrent models, which process data sequentially, TSTs operate in parallel, allowing efficient modeling of long-range dependencies without the constraints of recursion. Temporal positional encodings were incorporated to preserve the ordering of observations, while multi-head attention layers learned which past time steps were most relevant for future predictions. Residual connections and normalization techniques were applied to improve stability and convergence during training. The model's flexibility and scalability make it a strong candidate for high-dimensional and multivariate time series forecasting.

3 Research results

This section presents the results of developing and evaluating prediction models for energy consumption. The focus is on assessing predictive performance, analyzing the impact of model architecture and hyperparameter optimization, and validating results through quantitative metrics. The following subsections describe the model design, training process, optimization strategy, and key findings, highlighting the performance of the predictive models.

3.1 Model architecture

The best prediction model is based on the TST, which employs only the encoder component of the original Transformer architecture. This approach proved most effective for capturing complex temporal dependencies in the data. The data flow through the model is structured as follows:

- **Input Projection** – The prepared input vectors (16 input features for prediction models) are first passed through a linear layer. This layer projects the input vectors from their original lower-dimensional space into a higher-dimensional latent space (d_{model}), enabling the model to operate in a richer representational domain.
- **Positional Encoding** – Since the Transformer architecture is inherently permutation-invariant, explicit temporal order must be introduced. Sinusoidal and cosine positional encodings are added to the input vectors, assigning a unique temporal signa-

ture to each point in the 7-day input sequence. This allows the model to understand the sequential structure of the data.

- Transformer Encoder – The core of the model consists of a stack of identical encoder layers (e.g., three layers for consumption forecasting). Each layer includes two main components:
 - Multi-Head Self-Attention – This mechanism enables the model to examine the entire 7-day input sequence simultaneously and determine which time steps are most relevant to each other. Multiple attention heads operate in parallel, with each head capturing different temporal patterns (e.g., daily vs. weekly dependencies).
 - Feed-Forward Network – Following the attention step, each time step is independently processed through a small fully connected neural network to enrich its representation. Residual connections and layer normalization are applied throughout the encoder to ensure stable and efficient training of the deep architecture.
- Information Aggregation and MLP Head – After passing through all encoder layers, only the output corresponding to the last time step of the input sequence is retained. This vector effectively summarizes all relevant information from the preceding 7 days. It is then fed into a multi-layer perceptron (MLP) head consisting of one hidden layer with a ReLU activation. The MLP generates the final forecast for the next 2 days (192 time steps). Compared to a single linear output layer used in earlier versions of the model, the MLP head enables a more expressive nonlinear mapping between the learned temporal representations and the final prediction.

3.2 Hyperparameter optimization

The model training process was guided by optimization tools and hyperparameter configurations to ensure robust convergence and generalization. The AdamW optimizer was employed as a modern and robust variant of Adam, incorporating weight decay directly into the update rule. This improves regularization and often results in better overall performance compared to the standard Adam optimizer. Instead of a fixed learning rate, the OneCycleLR scheduler dynamically adjusts the learning rate throughout training. It begins with a low value, gradually increases to a predefined maximum, and then decreases toward the end of training. This cyclical schedule accelerates convergence and helps the model escape suboptimal local minima. SmoothL1Loss (Huber Loss) was used to balance sensitivity and stability. The loss behaves quadratically for small errors (similar to MSE) and linearly for large errors (similar to MAE), making it less sensitive to sudden spikes and outliers. This encourages the model to handle abrupt changes in consumption without being destabilized by excessive penalty values. Key architectural and regularization parameters were tuned for the consumption models, including the number of encoder layers (NUM_ENCODER_LAYERS), model dimensionality (D_MODEL), number of attention heads (N_HEADS), and regularization factors (DROPOUT_RATE, WEIGHT_DECAY). This hyperparameter optimization strategy ensured that each model variant was appropriately tailored to the specific forecasting challenge, contributing to improved stability and predictive performance.

3.3 Performance evaluation and comparative analysis

Model training was conducted over a predefined number of epochs, with progress carefully monitored to ensure both convergence and generalization. Three key metrics were evaluated during each epoch, separately for the training and validation sets:

- Loss – the value of the loss function, serving as the primary signal for the optimizer.
- MAE (Mean Absolute Error) – the main evaluation metric, representing the average absolute error in kWh and directly indicating how much the model deviates from actual values on average.
- wMAPE (weighted Mean Absolute Percentage Error) – a relative error metric expressed as a percentage of the true value, allowing for performance comparisons across power plants of varying sizes.

Model selection was based on the MAE calculated on the validation set. After each epoch, if the current MAE was lower than all previous values, the model was considered "best so far," and its weights were stored for later use. To prevent overfitting and unnecessary computation, an early stopping mechanism was applied. Training was automatically stopped if no improvement in MAE was observed for a predefined number of consecutive epochs (e.g., 10), using a patience parameter (PATIENCE_EARLY_STOP). The best-performing model for consumption prediction was the Time Series Transformer (TST), which achieved an MAE of 3.80 on the validation set.

Figure 1 illustrates an example of model predictions on the validation and test sets, where the actual values of consumption are compared with the corresponding forecasts. The results show that the model successfully captures the overall temporal patterns and trends, while certain discrepancies occur during rapid fluctuations, where peak values are either underestimated or overestimated. This visual inspection complements the quantitative error metrics and provides an intuitive understanding of the model's predictive performance.

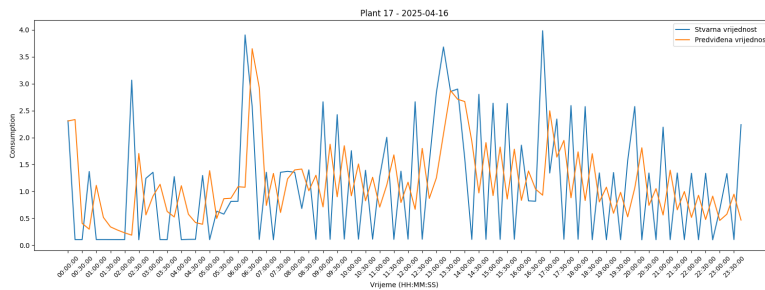


Fig. 1: Actual versus predicted values.

At the final stage of the analysis, a Permutation Feature Importance (PFI) procedure was conducted. This approach quantifies the contribution of each input feature by measuring the deterioration in model performance - expressed as an increase in the Mean

Absolute Error (MAE) - that occurs when the values of a specific feature are randomly permuted. Features causing the largest degradation in performance are interpreted as being the most influential for the model. The results of this analysis are illustrated in figure 2 for a representative example.

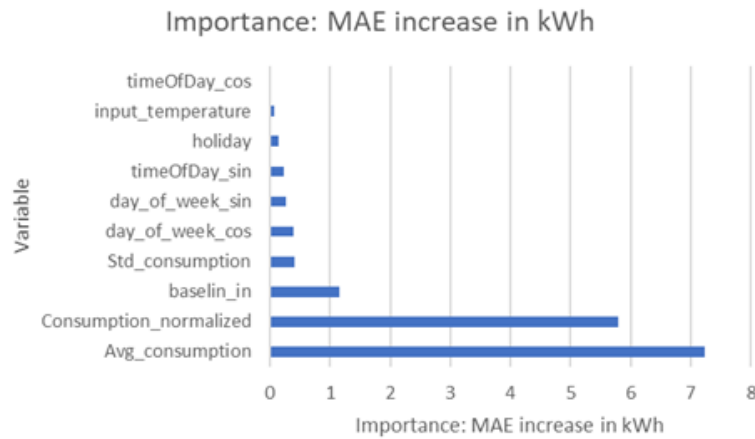


Fig. 2: Feature importance.

Figure 2 shows the permutation feature importance for the consumption prediction model, measured as the increase in MAE (kWh) when each feature is randomly shuffled. The results indicate that `avg_consumption_in` and `consumption_normalized` are by far the most influential variables, contributing 7.24 kWh and 5.79 kWh to prediction accuracy, respectively. Secondary contributors include `baseline_in` and `std_consumption_in`, while temporal features such as day-of-week and time-of-day encodings have moderate importance. Weather-related variables (e.g., temperature, wind speed, irradiation) exhibit minimal impact, suggesting that short-term consumption patterns are driven primarily by historical load profiles rather than external environmental conditions.

4 Discussion

The research has successfully developed and validated deep learning models for solar energy consumption forecasting. Key achievements include:

- **Methodological Innovation:** Implementation of lagged interval training and weather data augmentation techniques.
- **Architectural Advancement:** Development of Transformer architecture optimized for renewable energy forecasting.
- **Performance Excellence:** Achievement of low prediction error rates across comprehensive test datasets.

The results demonstrate that advanced machine learning architectures, when properly adapted for renewable energy applications, can achieve high forecasting accuracy that exceeds traditional approaches. This level of precision has important implications for grid management, energy trading, and sustainable energy system optimization. The high forecasting accuracy achieved by the Transformer model opens significant opportunities for integration into intelligent battery management systems for solar installations. Hereinafter, we discuss economic and environmental sustainability impact of such approaches.

4.1 Economic impact

The low prediction error rate provides the precision necessary for optimal battery charging and discharging decisions, which are critical for maximizing economic returns and extending battery lifecycle, in following:

- **Predictive Charging Strategies:** Accurate consumption forecasts enable proactive battery charging during periods of excess solar generation.
- **Load Balancing Optimization:** Precise demand predictions facilitate optimal energy distribution between immediate consumption, battery storage, and grid export.
- **Battery Lifecycle Management:** Accurate forecasting reduces unnecessary charge-discharge cycles, potentially extending battery operational life.

To fully realize the potential of AI-driven battery optimization, the current consumption forecasting model must be integrated with complementary predictive systems:

- **Electricity Price Forecasting:** Machine learning models predicting short-term electricity market prices with similar accuracy to consumption forecasting.
- **Solar Production Forecasting:** Parallel models predicting energy generation to complement consumption predictions.
- **Battery State Optimization:** Models determining optimal state-of-charge levels based on forecasted consumption, production, and pricing conditions.
- **Grid Interaction Models:** Systems optimizing the timing and magnitude of grid energy purchases and sales.

Preliminary economic analysis suggests that AI-optimized battery management systems could deliver substantial financial benefits. Besides economic benefits, there are environmental sustainability impacts, discussed in the following subsection.

4.2 Environmental sustainability impact

The implementation of high-precision solar energy consumption forecasting delivers significant environmental benefits that align with global sustainability objectives and carbon reduction targets [12]. The model's accuracy directly contributes to reduced

carbon emissions through several interconnected mechanisms that optimize renewable energy utilization while minimizing environmental impact.

Optimized solar energy utilization reduces dependence on carbon-intensive grid electricity, which is typically sourced from fossil fuel power plants. This fossil fuel displacement represents one of the most direct environmental benefits of accurate consumption forecasting. Additionally, precise demand forecasting enables better grid planning and reduces the need for peaking power plants, which often operate with higher emission rates due to their rapid response requirements. The technology ensures maximum utilization of available solar generation through precise consumption predictions, effectively minimizing energy waste and maximizing the environmental value of existing renewable infrastructure.

Based on typical solar installation performance and grid emission factors, the implementation of AI-optimized energy systems delivers quantifiable environmental improvements. These benefits include significant CO₂ emission reductions, substantial grid strain mitigation, and marked improvements in renewable energy efficiency. The cascading effects extend to broader resource conservation through optimized charging patterns that reduce battery degradation, decreasing the frequency of battery replacement and associated manufacturing emissions. This approach reduces grid infrastructure stress through intelligent load management while extending equipment lifecycles, thereby reducing the need for manufacturing replacement components. The AI-driven optimization system supports circular economy principles by minimizing energy waste through precise demand-supply matching, extending the operational lifespan of solar and battery systems, and maximizing the value extracted from existing renewable energy infrastructure. This holistic approach to resource utilization represents a paradigm shift from linear consumption models to circular resource optimization strategies that prioritize longevity and efficiency.

The widespread adoption of accurate consumption forecasting systems contributes meaningfully to broader climate change mitigation efforts by enhancing the economic viability of solar installations, which accelerates renewable energy adoption rates across diverse market segments [11]. Increased distributed renewable generation supports overall grid decarbonization while simultaneously reducing dependence on fossil fuel imports, thereby enhancing energy security and reducing geopolitical risks associated with energy dependency.

The technology promotes environmental sustainability in an equitable manner by improving the economic performance of residential solar systems, making clean energy accessible to broader populations regardless of economic status [10]. Reduced grid strain benefits entire communities through improved electrical system reliability, while localized reductions in fossil fuel consumption contribute to improved air quality in urban and industrial areas where air pollution disproportionately affects vulnerable populations.

Indirect environmental benefits extend beyond immediate energy applications to include reduced mining impact through extended battery lifecycles, which decrease demand for lithium, cobalt, and other battery materials while reducing associated mining environmental impacts [13]. The technology contributes to habitat preservation by reducing the need for new power plant construction, while decreased reliance on thermal

power plants reduces water consumption for cooling systems, supporting broader water conservation objectives.

The environmental benefits scale significantly across multiple implementation levels. Individual residential installations deliver measurable household carbon footprint reductions, while neighborhood-level deployments create substantial local environmental improvements that compound across communities. Large-scale regional adoption contributes meaningfully to regional and national carbon reduction targets, and technology transfer and adoption in developing countries supports global climate objectives through accessible clean energy solutions.

This research directly supports multiple United Nations Sustainable Development Goals, including [14] (Goal 7: Affordable and clean energy), through enhanced solar system economics and performance that advance affordable and clean energy access. The work contributes to [15] (Goal 9: Industry, innovation and infrastructure) by demonstrating advanced AI applications in renewable energy infrastructure, supports [16] (Goal 11: Sustainable cities and communities) through improved urban energy management capabilities, and directly advances [17] (Goal 13: Climate action) through measurable contributions to carbon emission reductions and climate action initiatives.

The development of high-precision consumption forecasting represents an important technological building block for achieving a sustainable energy future. As the technology matures and scales, its environmental impact compounds through widespread adoption that creates cumulative environmental benefits across diverse applications and geographic regions. The success in solar forecasting serves as an innovation catalyst, accelerating the development of similar AI applications across renewable energy sectors while generating demonstrated environmental benefits that support policy frameworks favoring renewable energy adoption. These proven sustainability benefits attract increased investment in clean energy technologies, creating positive feedback loops that accelerate the transition to sustainable energy systems.

The environmental sustainability implications of accurate solar energy consumption forecasting extend far beyond individual installations, contributing to systemic changes in energy production, consumption, and management that collectively advance global environmental objectives while supporting the transition to a sustainable energy future.

5 Conclusions

This research demonstrates that deep learning, particularly advanced architectures such as TST, can enhance the accuracy of solar energy consumption forecasting, achieving precision levels that open new possibilities for intelligent energy management systems. The research reveals that while powerful algorithms are essential, the quality and consistency of underlying data play an equally crucial role in determining model performance. Without careful data preparation and temporal alignment across different monitoring systems, even the most advanced machine learning models cannot achieve their full potential. The evolution from LSTM networks to the final TST implementation illustrates the importance of architectural innovation tailored specifically to the unique characteristics of renewable energy data. The final model achieved high performance

representing advancement over traditional forecasting approaches and positioning the system as ready for live operational testing in real-world environments.

Beyond the technical achievements, this work establishes the pathway for expanding AI-driven energy management ecosystems. The success of accurate consumption forecasting provides the foundation for more sophisticated applications, including the integration of electricity market price forecasting models and the development of comprehensive solar generation prediction systems. The convergence of consumption forecasts, production predictions, and dynamic pricing data creates opportunities for multi-objective optimization frameworks that can simultaneously optimize energy efficiency, cost-effectiveness, and environmental impact.

Several limitations must be acknowledged. The current results, while promising, are based on a relatively constrained dataset from a limited number of power plants, which may affect the generalizability of findings across diverse installation types and geographical regions. Additionally, restricted data access prevented the inclusion of all available monitoring systems. The simulation of historical weather forecasts, while a common practice in the field, may not perfectly capture the noise and uncertainty characteristics of real-world meteorological predictions.

Future research directions focus on addressing these limitations while expanding the scope and impact of the work. Dataset expansion through the inclusion of additional solar power plants and extended data collection periods will enhance model robustness and generalizability across diverse operational contexts. Integration of previously inaccessible systems will provide comprehensive coverage of available infrastructure and improve model validation across different manufacturer platforms. Most critically, the research plans point toward moving beyond prediction into optimization, with particular emphasis on intelligent battery storage management systems that leverage AI to determine optimal energy storage and release strategies.

The ultimate goal of the research includes comprehensive energy management ecosystems where accurate consumption forecasting serves as the foundation for sophisticated optimization algorithms that balance energy efficiency, cost-effectiveness, and environmental sustainability. This includes the development of real-time adaptation systems that continuously learn from changing operational conditions, standardized interfaces for utility grid interaction and compliance, and integrated platforms that simultaneously optimize multiple objectives across consumption, production, storage, and market participation. The success demonstrated in this research validates the potential for AI-driven solutions to address critical challenges in renewable energy management while contributing meaningfully to global sustainability objectives. As the technology matures and scales, its impact will extend beyond individual installations to support systemic changes in energy production, consumption, and management that collectively advance environmental sustainability and economic efficiency in the transition toward a clean energy future.

Generative artificial intelligence usage

During the preparation of this manuscript, the authors used ChatGPT, Claude, Perplexity, and Elicit for editing references, grammar and spelling checks, formatting and edit-

ing the text, and identifying relevant works in the field. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

References

1. Benti, N. E., Chaka, M. D., & Semie, A. G. (2023). Forecasting renewable energy generation with machine learning and deep learning: Current advances and future prospects. *Sustainability*, *15*(9), Article 7087. <https://doi.org/10.3390/su15097087>
2. Suanpang, P., & Jamjuntr, P. (2024). Machine learning models for solar power generation forecasting in microgrid application implications for smart cities. *Sustainability*, *16*(14), Article 6087. <https://doi.org/10.3390/su16146087>
3. Bouquet, P., Jackson, I., Nick, M., & Kaboli, A. (2024). AI-based forecasting for optimised solar energy management and smart-grid efficiency. *International Journal of Production Research*, *62*(13), 4623–4644. <https://doi.org/10.1080/00207543.2023.2293732>
4. Bassey, K. E. (2023). Hybrid renewable energy systems modeling. *Engineering Science and Technology Journal*, *4*(6), 571–588. <https://doi.org/10.51594/estj.v4i6.678>
5. de Campos, B. N., Maionchi, D. d. O., da Silva, J. G., Biudes, M. S., Oliveira, N. N. d., & Palácios, R. d. S. (2025). Photovoltaic energy modeling using machine learning applied to meteorological variables. *Sustainability*, *17*(16), Article 7506. <https://doi.org/10.3390/su17167506>
6. Al-Dahidi, S., Ayadi, O., Alrbai, M., & Adeeb, J. (2019). Ensemble approach of optimized artificial neural networks for solar photovoltaic power prediction. *IEEE Access*, *7*, 81741–81758. <https://doi.org/10.1109/ACCESS.2019.2929991>
7. Abbasimehr, H., & Paki, R. (2022). Improving time-series forecasting using LSTM and attention models. *Journal of Ambient Intelligence and Humanized Computing*, *13*(2), 673–691. <https://doi.org/10.1007/s12652-020-02761-x>
8. Ciechulski, T., & Osowski, S. (2021). High-precision LSTM model for short-time load forecasting in power systems. *Energies*, *14*(11), Article 2983. <https://doi.org/10.3390/en14112983>
9. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long-sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 11106–11115. <https://doi.org/10.1609/aaai.v35i12.17325>
10. Kämäräinen, J.-K. (2025). *Minimal time series transformer* (arXiv:2503.09791). arXiv. <https://doi.org/10.48550/arXiv.2503.09791>
11. Pimenow, S., Pimenowa, O., & Prus, P. (2024). Challenges of artificial intelligence development in the context of energy consumption and impact on climate change. *Energies*, *17*(23), Article 5965. <https://doi.org/10.3390/en17235965>
12. Danish, M. S. S., & Senjyu, T. (2023). AI-enabled energy policy for a sustainable future. *Sustainability*, *15*(9), Article 7643. <https://doi.org/10.3390/su15097643>
13. Würtz, S., Bogenberger, K., Göhner, U., & Rupp, A. (2024). Towards efficient battery electric bus operations: A novel energy forecasting framework. *World Electric Vehicle Journal*, *15*(1), Article 27. <https://doi.org/10.3390/wevj15010027>
14. United Nations Department of Economic and Social Affairs. (2015). *Goal 7: Affordable and clean energy*. United Nations. <https://sdgs.un.org/goals/goal7>
15. United Nations Department of Economic and Social Affairs. (2015). *Goal 9: Industry, innovation and infrastructure*. United Nations. <https://sdgs.un.org/goals/goal9>
16. United Nations Department of Economic and Social Affairs. (2015). *Goal 11: Sustainable cities and communities*. United Nations. <https://sdgs.un.org/goals/goal11>
17. United Nations Department of Economic and Social Affairs. (2015). *Goal 13: Climate action*. United Nations. <https://sdgs.un.org/goals/goal13>

About authors

Dijana Oreški is an Associate Professor of Artificial Intelligence (AI) and Machine Learning (ML) at the University of Zagreb, Faculty of Organization and Informatics. She is the head of the Laboratory for Data Mining and Intelligent Systems (LOUISE). Her research interest lies at the intersection of artificial intelligence and social sciences, focusing on the application of AI and ML to address societal challenges and support sustainable development. She has (co)authored more than 100 scientific papers and has participated in several dozen scientific and professional projects.

Vjeran Strahonja is a Professor Emeritus of Computer Science at the University of Zagreb, Faculty of Organization and Informatics. Throughout his distinguished academic career, he has held a number of leadership and managerial positions, including serving as Vice Dean and Dean of the Faculty of Organization and Informatics. His academic and professional work focuses on information systems, software engineering, and the strategic application of information technology in organizations. His long-standing contribution to academia and institutional development has had a significant impact on the field.

Marija Pokos Lukinec is a research and teaching assistant at the University of Zagreb, Faculty of Organization and Informatics, specializing in the field of artificial intelligence. Her research interests include data analytics and the application of artificial intelligence in education. She has (co)authored several scientific papers and participates in scientific and professional projects related to artificial intelligence.

Financing Green Solutions: Asset Returns and Tail Risks

Eugenijus Gabrielius Ivanauskas,
Liepa Urbonaitė, and
Saulius Jokubaitis

Abstract

The translation of green technologies from theoretical potential to industrial adoption hinges on the stability of the financial mechanisms that fund them. This Chapter investigates the risk dynamics and dependence structures of the Green financial ecosystem, comprising Green Bonds, Clean Energy ETFs, and Carbon Credits. Employing a rolling-window ARMA-GARCH-Vine Copula framework we map the evolving topology of sustainable finance and test the “decoupling hypothesis”. The analysis reveals that while Green Bonds have successfully decoupled and act as effective portfolio diversifiers, Clean Energy equities remain deeply integrated with broad market risks, functioning as a centralised “hub” for volatility transmission. We further identify a distinct structural shift around mid-2024, where with the fade of the post-pandemic volatility the market network stabilises into an R-Vine structure that transmits shocks more efficiently. Finally, we assess the viability of clean cryptocurrencies, finding them structurally incompatible with institutional hedging strategies due to extreme tail-risk dependence.

Keywords: sustainable finance; vine copulas; ARMA-GARCH models; volatility transmission; risk modelling; tail dependence.

Eugenijus Gabrielius Ivanauskas
Vilnius University, Institute of Applied Mathematics, Naugarduko str. 24, LT-03225, Vilnius, Lithuania
e-mail: gabrielius.ivanauskas@mif.stud.vu.lt,
Liepa Urbonaitė
Vilnius University, Institute of Applied Mathematics, Naugarduko str. 24, LT-03225, Vilnius, Lithuania
e-mail: liepa.urbonaite@mif.stud.vu.lt,
Saulius Jokubaitis
Vilnius University, Institute of Applied Mathematics, Naugarduko str. 24, LT-03225, Vilnius, Lithuania
e-mail: saulius.jokubaitis@mif.vu.lt

1 Introduction

Driven by rapid technological innovation, the scientific community continues to develop viable solutions to the climate crisis. Yet, the translation of these scientific breakthroughs from theoretical potential to widespread industrial adoption hinges on a critical enabler: capital. The central question facing the green transition is no longer just *what* technologies we need and can create, but *how* we can finance both their research and the deployment at the necessary scale.

Ideally, this financial process should be a frictionless, seamless flow of global capital into sustainable projects. In reality, however, the financial instruments that fund these solutions are complex, volatile, and deeply interconnected with the very fossil-fuel systems they aim to replace. If investors perceive uncontrolled risk or hidden contagion channels, the cost of capital spikes, as all that risk is priced in as a premium, and the deployment of green technology stalls. Consequently, the success of every innovation proposed in this chapter hinges on a robust understanding of the financial mechanisms that underpin them. If left unmanaged, financial risk forces capital providers to demand prohibitive risk premiums or exit the sector entirely, leaving the market open to purely speculative volatility. This in turn creates a perfect environment for liquidity crises and frozen credit lines, ultimately threatening to bankrupt even the most scientifically promising green ventures before they can reach industrial scale.

The purpose of this chapter is to eliminate this friction. By rigorously mapping the returns, volatility and tail-risk transmission of green financial assets we aim to provide the risk-management blueprint necessary to keep the capital flowing. We move beyond the feasibility of green solutions to assess their financial feasibility, distinguishing how these assets behave in calm markets versus periods of systemic shocks.

1.1 Regulatory drivers and market growth

In recent years, the financial sector has found itself in a strong position to support the shift toward a greener economy. The growing demand for green assets isn't just the result of investors paying more attention to Environmental, Social, and Governance (ESG) principles. A major driver is the rigorous regulatory environment. Policies such as carbon pricing and subsidies for renewable energy have shifted investment trends: they raise the cost of holding high-carbon assets and make sustainable projects more financially attractive [48]. Moreover, the development of market infrastructure has been essential in limiting the risk of greenwashing [19]. Efforts like the Climate Bonds Initiative's Green Bond Dataset Methodology [17] and the EU's Sustainable Finance Disclosure Regulation (SFDR) [7] have introduced clearer, standardized classifications. These frameworks make it easier to verify the credibility of green assets, improve transparency, and strengthen institutional confidence.

Although theory suggests that green assets may offer diversification benefits, empirical results are mixed when it comes to their hedging performance during systemic crises [9]. As a result, it is important to examine tail dependence between sustainable

and traditional assets (e.g. stocks and commodities), together with possible spillover effects and their implications for risk.

1.2 Green asset classes

We distinguish four essential financial flows. First, there is the emissions trading system (ETS), designed to manage carbon emissions and phase out old technologies. ETS sets a declining cap on total greenhouse gas emissions and issues tradable allowances, often called carbon credits. Regulated companies must monitor and report emissions each year and submit allowances equal to those emissions. In case of a breach, they face fines, must buy and submit the missing allowances and the breach is published. Allowances are mostly auctioned, with some free allocation to reduce carbon leakage. Firms can buy and sell allowances, so those that cut emissions can sell surplus units while higher emitters buy more. Because the supply of allowances is policy-controlled and falls over time, companies have an incentive to invest in cleaner technologies and to sell surplus credits as prices rise. ETSs typically cover power generation, heavy industry and parts of transport, however in the EU, for example, coverage is expanding to shipping, with a separate system for buildings and road transport. Trading of such allowances improves liquidity and ideally guides investment and faster transition towards lower-carbon technologies. Unlike a carbon tax, which fixes the price per tonne and leaves total emissions uncertain, an ETS fixes the cap and lets the market set the price, so cuts happen where they cost least [37].

Second, the green bond market is designed to finance sustainable, environmentally focused projects. This type of investment has attracted investor interest, supported by diversification benefits and robust transparency. Typically, the funds are used to finance improvements in transport infrastructure, agricultural production, renewable energy generation, and related climate-oriented investments. An important distinction can be made between a “green bond” and a “climate bond”. While both are similar, the latter is designed specifically to finance initiatives that are related to climate change, such as reducing carbon emissions and preventing pollution. Environmental bonds can be issued by various types of issuers, including supranational organisations, national governments, sub-national authorities (such as regions and cities) and corporations. [10].

Third, there are ETFs, stocks and other instruments that use funds for green companies and projects. These often focus on renewable energy, recycling, waste reduction, pollution control, sustainable transport and other green activities. To identify green companies, investors usually use ESG ratings. MSCI, a global provider of investment research and indices, provides such ratings and evaluates companies from CCC to AAA [43]. Examples of ETFs that use ESG or clean energy criteria include iShares ESG MSCI USA Leaders ETF (SUSL), Vanguard ESG U.S. Stock ETF (ESGV), iShares Global Clean Energy ETF (ICLN). These markets are growing fast, which can also increase price swings and speculative risk.

Finally, we consider an emerging asset class of clean cryptocurrencies. While these digital assets are generally not labelled as green, they are often regarded as more climate-friendly due to (unlike Bitcoin, which relies on a high-energy mining process)

the energy-efficient proof-of-stake protocol that significantly reduces electricity consumption and carbon footprint [26, 42, 56]. One hypothesis is that such cryptocurrencies exhibit different tail-risk characteristics and stronger connectedness with the green economy than traditional cryptocurrencies.

1.3 Systemic risk and the decoupling hypothesis

To attract and commit long-term investments to environmental projects, investors and policymakers need to understand how the risks of these assets change under different market conditions, especially how they behave together during extreme events. As green assets become more widely traded, they are more exposed to spillovers when shocks in speculative markets spread to connected markets. During economic turmoil, market interconnections can cause sell-offs and price drops that destabilise sustainable portfolios. In theory, diversification across green instruments should help counter these effects.

The purpose of this study is to empirically assess these dynamics by modelling the univariate risk and multivariate dependence structures of green financial instruments under both calm and volatile conditions. To capture the drivers of global market stress, the analysis incorporates key external variables, including industrial and precious metal futures (copper, gold), broad equity market trends (S&P 500) and energy commodity shocks (natural gas). The aim is to span a set of specific dimensions of the global economic environment. We consider Gold futures (GCF) to act as a proxy for safe-haven demand and inflationary expectations, allowing us to control for flight-to-safety episodes during market turmoil [5, 38]. Conversely, copper futures (HGF), often referred to as "Doctor Copper", serve as a barometer for global industrial activity and business cycle fluctuations [60]. Crucially, it is also one of the key material inputs needed for green technology development (e.g., electric vehicles, wind turbines), creating a physical link between commodity prices and renewable sector profitability [29]. The S&P 500 represents the baseline for global equity risk sentiment, capturing the broad market movements. Finally, while Natural Gas (TTF) was the central driver of the 2022 crisis, our analysis treats it as a "stress test" variable to evaluate how the market structure adapts to energy shocks. As a marginal price-setter for electricity in many markets [63], the shocks of natural gas price directly impact the cost competitiveness of renewable energy alternatives. This makes the variable crucial for analysing the decoupling hypothesis (see, e.g., [41, 53]).

The decoupling hypothesis posits that the performance of green assets should increasingly diverge from that of the traditional carbon-heavy markets as the transition to a low-carbon economy intensifies [64]. The expected separation is driven by two fundamental drivers. First, while the traditional energy markets are governed by commodity supply shocks and geopolitical events, the green asset returns should be increasingly influenced by technological innovation, regulatory requirements and long-term policy support [13, 27, 39]. If this hypothesis holds, green assets serve as not only attractive instruments for capital allocation, but as effective hedges against carbon transition risk [55]. However, recent evidence suggests that financial contagion and shared

macroeconomic factors may re-couple these sectors during periods of systemic stress [3, 54].

Green bonds typically demonstrate a weak and time-varying correlation with traditional energy markets, indicating partial decoupling from non-climate-friendly assets [45, 49]. However, these relationships are highly dynamic and tend to spike during periods of systemic stress, such as the Global Financial Crisis, the COVID-19 pandemic, or regimes of extreme downside risk. During such stress periods, the interdependencies across assets strengthen, making joint extreme losses more likely and causing diversification benefits to decrease [14, 62]. This dependence differs across energy types: green bonds show an extreme negative tail dependence with oil and coal during crises but a positive dependence with natural gas, likely because both are seen as ways to limit environmental harm [45, 49]. Since green bonds do not closely follow the price swings of traditional energy, they can offer diversification benefits and some hedging potential, particularly for short-term investors seeking protection from energy market volatility [45, 49].

The green bond market is linked to broader financial sectors through spillover effects, with volatility spillovers occurring even when average correlations are low. While broad equity and energy markets tend to have minor direct influence on green bond prices, specific sectors, such as renewable energy equities, play a more important role in transmitting shocks to the green bond market [44, 66]. In the energy sector specifically, crude oil and green bonds are linked through asymmetric spillovers: negative return shocks in oil markets can increase green bond volatility, while green bond shocks may reduce oil market volatility as investors shift away from energy assets [44, 59]. These dynamics are highly time-dependent, as the connection with oil is typically weak in the short-run but strengthens in the long-run due to industrial competition for capital [44]. Furthermore, green bonds receive significant price spillovers from traditional corporate and treasury fixed-income markets, making them poor diversifiers for investors already holding traditional bonds [53, 66]. Carbon markets and cryptocurrencies also play a role in this ecosystem, with cryptocurrencies acting as a notable source of volatility for green bonds in bearish market conditions [59, 66].

To capture these potential non-linear dynamics and complex structures, we use an ARMA–GARCH–Vine–Copula modelling framework [14, 30]. This approach first models the marginal return dynamics and conditional volatility of each asset using ARMA–GARCH type specifications, such as GJR–GARCH, EGARCH, PARCH or others for asymmetric volatility and leverage effects [14, 51, 52, 62]. The dependence structure is then modeled using vine copulas [1, 8, 14, 15, 31], which provide the flexibility needed to capture asymmetric tail dependence and to identify specific causality between assets. The frequent use of these models in previous green finance studies supports their suitability for analyzing volatility, tail risk, and dependence in green financial markets [14, 51, 62].

While alternative methodologies such as DCC-GARCH [21, 57] or wavelet coherence [34, 35, 61] are widely used to assess time-varying connectedness, they often rely on restrictive assumptions regarding the joint distribution of returns. We consider the ARMA-GARCH-Vine-Copula approach more suitable for this analysis as it provides a flexible modelling framework for understanding tail risk without imposing strong symmetry assumptions in the dependence structure. This flexibility allows us to identify

whether green assets maintain their diversification properties when the market is under specific stress from energy commodity shocks.

Our contribution to the existing literature is threefold. First, we model the univariate volatility dynamics of these assets, accounting for the heavy tails and leverage effects characteristic of new financial markets. Second, we employ a vine copula framework to map the multivariate dependence structure. This approach allows us to move beyond simple linear correlations and identify tail dependence: the likelihood of simultaneous extreme losses. Finally, by challenging the decoupling hypothesis we find that the Green Bonds have successfully decoupled and are acting as effective portfolio diversifiers, while the clean energy equities remain deeply integrated with broad market risks. Furthermore, we identify a lagging structural shift, observed in our rolling window experiments around mid 2024. As the memory of the energy crisis fades, the green financial network has hardened, becoming less volatile on the average, but structurally more centralised around specific hubs (Clean Energy ETFs), allowing shocks to transmit faster than before.

2 Methodology

For each asset $j = 1, \dots, d$, a univariate return series is constructed using daily continuously compounded returns:

$$r_{j,t} = \ln(P_{j,t}) - \ln(P_{j,t-1}),$$

where $P_{j,t}$ denotes the closing price of asset j at time t .

To capture time-varying volatility, several GARCH-type models are estimated for each return series. All models follow the general structure:

$$r_{j,t} = \mu_j + \varepsilon_{j,t}, \quad \varepsilon_{j,t} = \sigma_{j,t} z_{j,t}, \quad \mu_j \in \mathbb{R},$$

where $\sigma_{j,t}$ is the conditional standard deviation and $z_{j,t}$ is an i.i.d. standardized innovation.

Since empirical asset returns typically exhibit heavy tails and excess kurtosis that are not adequately captured by the normal distribution [2, 33, 67], we assume that the standardized innovations follow a Student's t -distribution [11] where the degrees of freedom ν are estimated via maximum likelihood.

In this chapter we consider the following volatility models: GARCH, iGARCH, EGARCH, GJR-GARCH, APARCH, csGARCH. The first moments were modelled by using ARMA models.

The adequacy of the model is assessed in two stages. At first, standardized residuals are evaluated using the Ljung–Box and ARCH–LM tests to examine the remaining autocorrelation and conditional heteroskedasticity. Secondly, implied Value-at-Risk forecasts are validated using the Kupiec unconditional coverage test, Christoffersen conditional coverage test, the independence test, and the Basel Traffic Light test.

The baseline GARCH(1,1) model [12], which captures short-term memory and volatility clustering, is defined by eq. (1):

$$\sigma_{j,t}^2 = \omega_j + \alpha_j \varepsilon_{j,t-1}^2 + \beta_j \sigma_{j,t-1}^2. \quad (1)$$

While this baseline provides a foundational framework, our preliminary testing revealed that models incorporating leverage effects (GJR-GARCH) or long-run volatility components (csGARCH) provided a superior fit for our data. Consequently, for the sake of brevity, the detailed mathematical definitions and empirical results presented in the following sections focus strictly on these best-performing frameworks.

Value-at-Risk (VaR) forecasts are computed directly from the conditional mean and volatility:

$$\text{VaR}_{j,t+1}(\alpha) = \widehat{\mu}_{j,t+1} + \widehat{\sigma}_{j,t+1} q_\alpha,$$

where q_α is the α -quantile of the Student's t -distribution. We consider two confidence levels, namely the 95% and the 99% VaR.

2.1 GJR-GARCH

The GJR-GARCH model of Glosten, Jagannathan, and Runkle [24] extends the standard GARCH framework by allowing asymmetric volatility responses to positive and negative shocks. This asymmetry, known as the leverage effect, captures the empirically observed phenomenon that negative return innovations tend to increase volatility more than positive ones.

The conditional variance equation is given by:

$$\sigma_{j,t}^2 = \omega_j + \sum_{k=1}^q \left(\alpha_{j,k} \varepsilon_{j,t-k}^2 + \gamma_{j,k} I_{j,t-k} \varepsilon_{j,t-k}^2 \right) + \sum_{k=1}^p \beta_{j,k} \sigma_{j,t-k}^2, \quad (2)$$

where the indicator function $I_{j,t-k}$ is defined as:

$$I_{j,t-k} = \begin{cases} 1, & \varepsilon_{j,t-k} \leq 0, \\ 0, & \varepsilon_{j,t-k} > 0. \end{cases} \quad (3)$$

The parameter γ_k measures the leverage effect; $\gamma_k > 0$ implies stronger volatility responses to negative shocks.

2.2 csGARCH model

The model proposed by Lee and Engle decomposes the conditional variance into two distinct components: a permanent component and a transitory component [22]. This

decomposition allows for the analysis of both long- and short-term volatility dynamics in financial assets. The permanent component, denoted by $m_{j,t}$, evolves over time and replaces the constant intercept of the standard GARCH model with a time-varying term that follows first-order autoregressive dynamics. The transitory component is defined as the deviation of the conditional variance from its long-term trend, given by

$$\sigma_{j,t}^2 - m_{j,t}.$$

The model can be written as:

$$\sigma_{j,t}^2 = m_{j,t} + \sum_{k=1}^q \alpha_{j,k} (\varepsilon_{j,t-k}^2 - m_{j,t-k}) + \sum_{k=1}^p \beta_{j,k} (\sigma_{j,t-k}^2 - m_{j,t-k}), \quad (4)$$

$$m_{j,t} = \omega_j + \rho_j m_{j,t-1} + \phi_j (\varepsilon_{j,t-1}^2 - \sigma_{j,t-1}^2). \quad (5)$$

To ensure that the conditional variance remains non-negative, the model imposes parameter restrictions as described in [22]. In particular, the sum of the GARCH parameters α and β must be less than one, and the persistence parameter ρ governing the permanent component must satisfy $\rho < 1$. Together, these conditions guarantee the stationarity and stability of both the transitory and permanent volatility components.

2.3 eGARCH model

The exponential GARCH (eGARCH) model introduced by Nelson [47] can be written as:

$$\begin{aligned} \ln(\sigma_{j,t}^2) = & \left(\omega_j + \sum_{k=1}^m \zeta_{j,k} v_{j,k,t} \right) + \sum_{k=1}^q (\alpha_{j,k} z_{j,t-k} + \gamma_{j,k} (|z_{j,t-k}| - \mathbb{E}|z_{j,t-k}|)) \\ & + \sum_{k=1}^p \beta_{j,k} \ln(\sigma_{j,t-k}^2), \end{aligned} \quad (6)$$

where $z_{j,t} = \varepsilon_{j,t}/\sigma_{j,t}$ denotes the standardized innovations. The constant term ω_j captures the long-term level of logarithmic conditional variance, while the term $\zeta_{j,k} v_{j,k,t}$ allows the inclusion of additional variables that may influence volatility dynamics, such as macroeconomic or market-specific factors. When no additional variables are considered, this term is typically omitted.

A key advantage of the eGARCH model is that it does not require parameter restrictions, since the conditional variance is modelled in logarithmic form. This guarantees the positivity of the variance by construction. Using the logarithm of volatility also means that shocks affect volatility in relative terms rather than absolute levels. As a result, very large shocks do not cause excessively large increases in the conditional variance, leading to a more stable reaction to extreme market movements.

The parameters $\alpha_{j,k}$ and $\gamma_{j,k}$ describe how shocks influence volatility. The coefficient $\alpha_{j,k}$ captures asymmetric effects, allowing negative and positive shocks to have different impacts on volatility. The parameter $\gamma_{j,k}$ measures the effect of the size of shocks and determines how strongly volatility responds to large unexpected returns, regardless of their sign.

The persistence of volatility in the eGARCH model is mainly determined by coefficients $\beta_{j,k}$, which control how long the impact of shocks remains in the variance process. Due to the logarithmic specification, the persistence of volatility is more flexible than in standard GARCH models, as the speed at which shocks decay depends on the estimated parameters rather than fixed constraints.

2.4 Copulas

2.4.1 Sklar's theorem

The copula approach is based on Sklar's Theorem [58]. Let F be a d -dimensional joint distribution function with marginals F_1, \dots, F_d . Then there exists a copula $C : [0, 1]^d \rightarrow [0, 1]$ such that:

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)), \quad (x_1, \dots, x_d) \in \mathbb{R}^d. \quad (7)$$

Archimedean copulas

Archimedean copulas are constructed using a generator function ϕ :

$$C(u_1, \dots, u_d) = \phi^{-1}(\phi(u_1) + \dots + \phi(u_d)), \quad (8)$$

where $u_i = F_i(x_i)$ are the marginal probabilities in $[0, 1]$, $\phi : [0, 1] \rightarrow [0, \infty)$ is a strict Archimedean generator function (continuous, strictly decreasing, convex), ϕ^{-1} is the pseudo-inverse of the generator.

Elliptical copulas

Elliptical copulas are derived by inverting Sklar's theorem for elliptical distributions.

$$C(u_1, \dots, u_d) = F[F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)], \quad (9)$$

where F is a joint elliptical distribution function (e.g., Multivariate Normal or Student's t), F_j^{-1} is the inverse CDF of the univariate marginals of F .

2.5 Vine copula structures

To model dependence of the residuals of univariate volatility models we use the vine copula approach. Unlike Archimedean copulas, which typically rely on a single parame-

ter and can be restrictive in higher dimensions, vines allow each variable pair to be fitted with an appropriate copula family, providing flexibility, tail-dependence modelling and asymmetry. Vine copulas let different variable pairs be modelled with different copula families, providing substantial modelling flexibility. They build pair-copula constructions that decompose a multivariate distribution into bivariate components, with variables connected across a sequence of trees to represent dependence. The key property is that the joint density can be factorised into products of conditional densities [1, 8, 14, 15, 31]. Below we define all relevant notation.

2.5.1 C-Vine (Canonical vine)

The C-Vine structure is a "star-like" dependency where one pivot variable drives the dependence of the others. Joint density function is denoted as follows:

$$f(x_1, \dots, x_d) = \prod_{k=1}^d f_k(x_k) \times \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} c_{j, j+i|1:\dots:j-1} (F(x_j|\mathbf{x}_{1:j-1}), F(x_{j+i}|\mathbf{x}_{1:j-1})), \quad (10)$$

where $f_k(x_k)$ are the marginal densities, $c_{j, j+i|1:\dots:j-1}$ represents the pair-copula density between variable j and $j+i$, conditional on previous variables, $1 : \dots : j-1$ denotes the set of conditioning variables $\{x_1, \dots, x_{j-1}\}$. In a C-Vine, variable 1 is the root node for the first tree, variable 2 for the second, etc.

2.5.2 D-Vine (Drawable vine)

The D-Vine structure assumes a "line-like" or sequential dependency. A D-vine emerges when the dependence structure is sequential, with no dominant "hub" variable, meaning that each variable mainly depends on its neighbours rather than on a central node. Joint density function denoted as:

$$f(x_1, \dots, x_d) = \prod_{k=1}^d f_k(x_k) \cdot \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} c_{i, i+j|i+1:\dots:i+j-1} (F(x_i|\mathbf{x}_{i+1:i+j-1}), F(x_{i+j}|\mathbf{x}_{i+1:i+j-1})) \quad (11)$$

where the conditioning set $i+1 : \dots : i+j-1$ represents the variables strictly "between" i and $i+j$ in the sequence. For the first tree ($j = 1$), the conditioning set is empty, giving unconditional pair copulas $c_{i, i+1}$. D-vine is a special case of R-vine structure.

2.5.3 R-Vine (Regular vine)

The R-Vine is the most general structure, defined by a sequence of trees. Joint density is denoted as:

$$f(x_1, \dots, x_d) = \prod_{k=1}^d f_k(x_k) \times \prod_{i=1}^{d-1} \prod_{e \in E_i} c_{j(e), k(e) | D(e)} (F(x_{j(e)} | \mathbf{x}_{D(e)}), F(x_{k(e)} | \mathbf{x}_{D(e)})) \quad (12)$$

where E_i is the set of edges in tree T_i , $e = \{j(e), k(e)\}$ describes the two variables connected by the edge, $D(e)$ is the conditioning set for edge e (the variables needed to make the pair conditional).

2.5.4 Copula families

In this study, the candidates for the pair-copulas are restricted to a specific selection of Elliptical (Gaussian, Student's t) and Archimedean (Gumbel, Clayton) families (listed in Table 1). This selection is designed to provide a full coverage of potential dependence structures, specifically regarding tail behaviour, symmetry, and direction. The Gaussian copula is included as a benchmark for linear, symmetric dependence, suitable for capturing correlation in the centre of the distribution. In contrast, the Student's t copula is employed to capture symmetric fat tails, allowing for the modelling of simultaneous extreme events in both directions.

To account for asymmetric dependence, the Clayton and Gumbel families are used. The standard Clayton copula models lower tail dependence, while the Gumbel copula captures upper tail dependence. Furthermore, we extend the copulas set to include the rotated versions (90° , 180° , and 270°) of these Archimedean families. This extension is critical because standard Archimedean copulas are restricted to positive dependence, the 90° and 270° rotations allow the vine model to capture negative dependence structures. Additionally, the 180° rotations (or survival copulas) offer flexible forms for tail dependence in the opposite direction of the standard form, such as using a Survival Gumbel to model lower tail dependence with a different shape than the Clayton. Collectively, this set ensures the model selection algorithm can flexibly adapt to data exhibiting no tail dependence, symmetric tail dependence, or asymmetric dependence in any direction.

In the general case for a bivariate copula $C(u, v)$, the rotated versions are defined as: $C_{90^\circ}(u, v) = v - C(1 - u, v)$, $C_{180^\circ}(u, v) = u + v - 1 + C(1 - u, 1 - v)$, $C_{270^\circ}(u, v) = u - C(u, 1 - v)$. See for example [32, 46] as reference. Here $u, v \in (0, 1)$ denote uniform marginal variables obtained via the probability integral transform. In Table 1 $\Phi(\cdot)$ and $\Phi^{-1}(\cdot)$ represent the standard normal cumulative distribution function and its inverse, while $\Phi_\rho(\cdot, \cdot)$ denotes the bivariate standard normal distribution with correlation parameter ρ . Similarly, $t_\nu(\cdot)$ and $t_\nu^{-1}(\cdot)$ denote the cumulative distribution function and quantile function of the univariate Student's t -distribution with ν degrees of freedom, and $t_{\nu, \rho}(\cdot, \cdot)$ denotes the corresponding bivariate Student's t -distribution with correlation ρ . The parameter θ controls the strength of dependence in the Archimedean copulas, with $\theta > 0$ for the Clayton family and $\theta \geq 1$ for the Gumbel family.

Table 1: Copula families and their rotated versions used in the vine model.

Family	Copula formula	Parameter(s)	Tail Dependence
Gaussian	$\Phi_\rho(\Phi^{-1}(u), \Phi^{-1}(v))$	$\rho \in (-1, 1)$	None
Student- t	$t_{\nu, \rho}(t_\nu^{-1}(u), t_\nu^{-1}(v))$	$\rho \in (-1, 1)$, $\nu > 0$	Upper & Lower
Clayton	$(u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}$	$\theta > 0$	Lower
Clayton 90°	$v - ((1 - u)^{-\theta} + v^{-\theta} - 1)^{-1/\theta}$	$\theta > 0$	Negative
Clayton 180°	$u + v - 1 + ((1 - u)^{-\theta} + (1 - v)^{-\theta} - 1)^{-1/\theta}$	$\theta > 0$	Upper
Clayton 270°	$u - (u^{-\theta} + (1 - v)^{-\theta} - 1)^{-1/\theta}$	$\theta > 0$	Negative
Gumbel	$\exp\left\{-\left[(-\ln u)^\theta + (-\ln v)^\theta\right]^{1/\theta}\right\}$	$\theta \geq 1$	Upper
Gumbel 90°	$v - \exp\left\{-\left[(-\ln(1 - u))^\theta + (-\ln v)^\theta\right]^{1/\theta}\right\}$	$\theta \geq 1$	Negative
Gumbel 180°	$u + v - 1 + \exp\left\{-\left[(-\ln(1 - u))^\theta + (-\ln(1 - v))^\theta\right]^{1/\theta}\right\}$	$\theta \geq 1$	Lower
Gumbel 270°	$u - \exp\left\{-\left[(-\ln u)^\theta + (-\ln(1 - v))^\theta\right]^{1/\theta}\right\}$	$\theta \geq 1$	Negative

2.6 Value at risk and expected shortfall calculation

2.6.1 VaR for copula models

Let d denote the number of assets in the portfolio and let $\mathbf{w} = (w_1, \dots, w_d)^\top$ be the vector of portfolio weights, where $w_j = 1/d$ for all $j = 1, \dots, d$. Let $t + 1$ denote the 1-day-ahead forecast horizon and let $\hat{\sigma}_{j,t+1}$ be the conditional volatility of asset j at horizon $t + 1$.

Let $\mathbf{Z} = (\widehat{Z}_1, \dots, \widehat{Z}_d)^\top$, denote standardized residuals of length 10000, simulated from the fitted vine copula model with skewed Student's t -marginals. The simulated return of asset j at horizon $t + 1$ is given by

$$\widehat{R}_{j,t+1} = \hat{\sigma}_{j,t+1} \widehat{Z}_j, \quad j = 1, \dots, d. \quad (13)$$

The corresponding simulated portfolio return is

$$\widehat{\mathbf{R}}_{t+1} = \sum_{j=1}^d w_j \widehat{R}_{j,t+1} = \sum_{j=1}^d w_j \hat{\sigma}_{j,t+1} \widehat{Z}_j, \quad (14)$$

For a confidence level $\alpha \in (0, 1)$, the VaR of the portfolio 1 day-ahead is defined as the lower α -quantile of the portfolio return distribution:

$$\text{VaR}_{\alpha,t+1} = \inf \left\{ x \in \mathbb{R} : \mathbb{P}(\widehat{\mathbf{R}}_{t+1} \leq x) \geq \alpha \right\}. \quad (15)$$

For backtesting purposes, a VaR exceedance (hit) indicator is defined as

$$\mathbb{I}_{t+1}(\alpha) = \mathbf{1} \{ \mathbf{R}_{t+1}^{\text{real}} < \text{VaR}_{\alpha,t+1} \}, \quad (16)$$

where $\mathbf{R}_{t+1}^{\text{real}}$ denotes the realized equally weighted portfolio return at time $t + 1$.

2.6.2 Expected shortfall

The Expected Shortfall (ES) at level α and horizon h is defined as the expected portfolio simulated return conditional on simulated returns falling below the VaR threshold:

$$\text{ES}_{\alpha,t+1} = \mathbb{E} \left[\widehat{\mathbf{R}}_{t+1} \mid \widehat{\mathbf{R}}_{t+1} \leq \text{VaR}_{\alpha,t+1} \right]. \quad (17)$$

2.7 Moving window approach

For each asset, a fixed-size rolling window of 1000 historical observations is used to estimate the parameters of an ARMA(1,1)–GARCH–type model, where the ARMA component captures conditional mean dynamics [14, 30] with different volatility models specified in sections before. The window rolls forward through time, refitting the model every 10 observations to balance computational efficiency with parameter adaptability. The standardized residuals of each window i and asset j (denoted as $z_{j,t}^{(i)}$) are computed by dividing the mean-adjusted returns by the conditional volatility estimates. The rolling procedure generates one-step-ahead conditional volatility forecasts ($\widehat{\sigma}_{j,t+1}$) for each time point.

The extracted residuals undergo diagnostic testing using Ljung-Box tests for serial correlation, Ljung-Box tests on squared residuals for remaining ARCH effects, and ARCH-LM tests. Only models that pass these diagnostic criteria with 5% significance are retained for further analysis, ensuring that the standardized residuals exhibit the white noise properties required for subsequent copula modelling.

This extraction process provides the essential inputs for the vine copula framework: properly standardized residuals that capture the pure innovations from each marginal model, along with corresponding volatility forecasts needed for risk aggregation and backtesting.

Let

$$\mathbf{z}_t^{(i)} = \left(z_{1,t}^{(i)}, z_{2,t}^{(i)}, \dots, z_{d,t}^{(i)} \right)$$

denote the vector of standardized residuals of volatility model at time t corresponding to a given window.

In the next step, for each rolling window, with window size W , $i = 1, \dots, T - W$, the joint dependence structure of the residual vector

$$\{\mathbf{z}_t\}_{t=i}^{i+W-1}$$

is modelled using a vine copula. Prior to dependence modelling, the marginal distributions of the standardized residuals are estimated parametrically within each rolling window using a skewed Student's t -distribution. The standardized residuals are then transformed to the unit interval via the probability integral transform (PIT):

$$U_{j,t} = F_{\text{skewed-}t}(z_{j,t}), \quad j = 1, \dots, d.$$

Based on the resulting multivariate copula model, joint simulations of standardized residuals are generated for each window. These simulated residuals are then rescaled using the corresponding volatility forecasts obtained from the rolling volatility models.

Finally, for each window, the simulated portfolio return distribution is constructed and used to compute one-day-ahead VaR forecasts at the desired confidence levels. The realized portfolio returns are then compared with the corresponding VaR forecasts, allowing for out-of-sample backtesting of the risk model. Additionally, Expected Shortfall is estimated using given VaR.

This moving window approach ensures that both marginal dynamics and dependence structures are allowed to evolve over time, providing a flexible framework for capturing time-varying risk characteristics in multivariate financial portfolios.

2.8 Backtesting value at risk

VaR backtesting evaluates whether the risk model accurately captures extreme losses by comparing predicted VaR levels with realized returns.

2.8.1 Basel traffic light test

The Basel Traffic Light Test evaluates whether the risk model accurately captures extreme losses by comparing predicted VaR levels with realized returns, where model performance is assessed by the number of VaR exceptions and their cumulative binomial probability as formalized in the Basel Traffic Light Test [4, 18].

Table 2: Model evaluation zones based on exceedance probability.

Zone	Condition
Green Zone	$\mathbb{P}(X \leq x) < 0.95$
Yellow Zone	$0.95 \leq \mathbb{P}(X \leq x) < 0.9999$
Red Zone	$\mathbb{P}(X \leq x) \geq 0.9999$

One of the criteria of rejecting the model is when the red colour assigned. Yellow light may indicate potential model problems that warrant further analysis.

2.8.2 Unconditional coverage (Kupiec) test

Let $N = T - W$ be the total number of observations and

$$x = \sum_{t=1}^N \mathbb{I}_{t+1}(\alpha)$$

the total number of VaR violations.

The unconditional coverage test examines whether the observed frequency of VaR violations is consistent with the nominal confidence level. A rejection of the null hypothesis indicates systematic underestimation or overestimation of risk [16, 36].

The likelihood ratio test statistic is

$$LR_{\text{POF}} = -2 \ln \left(\frac{(1 - \alpha)^{N-x} \alpha^x}{\left(1 - \frac{x}{N}\right)^{N-x} \left(\frac{x}{N}\right)^x} \right),$$

which asymptotically follows a chi-squared distribution with one degree of freedom:

$$LR_{\text{POF}} \sim \chi^2(1).$$

2.8.3 Independence test

Following the likelihood-based framework for testing interval forecasts introduced by Christoffersen [16], the independence of VaR violations can be examined using a transition matrix of exceedance indicators.

Let n_{ij} denote the number of transitions from state i at time $t - 1$ to state j at time t , where $i, j \in \{0, 1\}$. The transition probabilities are defined as

$$\pi_{01} = \frac{n_{01}}{n_{00} + n_{01}}, \quad \pi_{11} = \frac{n_{11}}{n_{10} + n_{11}}.$$

Under the null hypothesis of independence, the likelihood function is

$$L_0 = (1 - \pi)^{n_{00} + n_{10}} \pi^{n_{01} + n_{11}}, \quad \pi = \frac{n_{01} + n_{11}}{T}.$$

The unrestricted likelihood is

$$L_1 = (1 - \pi_{01})^{n_{00}} \pi_{01}^{n_{01}} (1 - \pi_{11})^{n_{10}} \pi_{11}^{n_{11}}.$$

The independence test statistic is

$$LR_{\text{IND}} = -2 \ln \left(\frac{L_0}{L_1} \right) \sim \chi^2(1).$$

2.8.4 Conditional coverage (CC) test

The conditional coverage test combines the unconditional coverage and independence tests:

$$LR_{CC} = LR_{POF} + LR_{IND} \sim \chi^2(2).$$

The conditional coverage test checks whether the number of violations is correct and whether violations occur independently. This joint likelihood ratio test of coverage and independence was developed by Christoffersen [16]. Passing this test indicates that the VaR model performs well overall.

2.9 Backtesting expected shortfall

For ES backtesting we consider the Exceedance Residual (ER), Conditional Calibration (CCL) and ES Regression (ESR), briefly described in Sections 2.9.1–2.9.3 Let $\{\mathbf{R}_t^{real}\}_{t=1}^T$ denote a sequence of portfolio returns and let $q_{t+1}(\alpha)$ and $e_{t+1}(\alpha)$ be the corresponding one-step-ahead VaR and ES forecasts at level $\alpha \in (0, 1)$.

2.9.1 Exceedance residual (ER) backtest

Following McNeil and Frey [40], we define the exceedance residual as the difference between the realized return and the Expected Shortfall forecast, standardized by the conditional standard deviation $\hat{\sigma}_{t+1}$:

$$ER_{t+1} = \frac{e_{t+1}(\alpha) - \mathbf{R}_{t+1}}{\hat{\sigma}_{t+1}} \mathbb{I}_{t+1}(\alpha), \quad (18)$$

where $\mathbb{I}_{t+1}(\alpha) = I\{\mathbf{R}_{t+1} \leq q_{t+1}(\alpha)\}$ is the indicator function for VaR violations.

The null hypothesis of a correct Expected Shortfall specification is that these residuals have a conditional mean of zero:

$$H_0 : \mathbb{E}[ER_{t+1}] = 0. \quad (19)$$

2.9.2 Conditional calibration (CCL) backtest

The CCL backtest of Nolde and Ziegel [50] is based on the joint identification function for VaR and ES,

$$V_{t+1}(q_{t+1}, e_{t+1}, \widehat{\mathbf{R}}_{t+1}) = \begin{pmatrix} \mathbb{I}\{\widehat{\mathbf{R}}_{t+1} \leq q_{t+1}\} - \alpha \\ e_{t+1} - q_{t+1} + \frac{1}{\alpha}(q_{t+1} - \widehat{\mathbf{R}}_{t+1})\mathbb{I}\{\widehat{\mathbf{R}}_{t+1} \leq q_{t+1}\} \end{pmatrix}. \quad (20)$$

Let Z_t denote a vector of instruments measurable with respect to the information set \mathcal{F}_t . The null hypothesis of conditional calibration is

$$H_0 : \mathbb{E}[Z_t \otimes V_{t+1}(q_{t+1}, e_{t+1}, \mathbf{R}_{t+1})] = 0, \quad (21)$$

where \otimes denotes the Kronecker product.

2.9.3 ES regression (ESR) backtest

The ESR backtest of Bayer and Dimitriadis [6] utilizes a joint Mincer-Zarnowitz regression framework. Since ES is not evaluated independently stand-alone, the test is based on a joint regression model for the VaR and the ES.

In the Strict ESR specification, which only requires ES forecasts as input variables, we estimate the following joint system for portfolio returns $\widehat{\mathbf{R}}_{t+1}$ and ES forecasts $e_{t+1}(\alpha)$:

$$\begin{aligned} \widehat{\mathbf{R}}_{t+1} &= \alpha_1 + \alpha_2 e_{t+1}(\alpha) + u_{t+1}^q, \\ \widehat{\mathbf{R}}_{t+1} &= \beta_1 + \beta_2 e_{t+1}(\alpha) + u_{t+1}^e, \end{aligned} \quad (22)$$

where (α_1, α_2) are the auxiliary VaR parameters, (β_1, β_2) are the ES-specific parameters and u is error term. The null hypothesis of a correctly specified Expected Shortfall is defined by the ES-specific regression coefficients:

$$H_0 : \beta_1 = 0 \quad \text{and} \quad \beta_2 = 1. \quad (23)$$

Under H_0 , the ES forecasts are correctly specified such that

$$e_{t+1}(\alpha) = \mathbb{E}[\widehat{\mathbf{R}}_{t+1} | \widehat{\mathbf{R}}_{t+1} \leq q_{t+1}].$$

3 Data

In this section we present daily asset returns used for the analysis, obtained from Yahoo Finance and EU ETS. The sample covers the period from 2018 November 27 to 2025 November 21, all assets in portfolio are chosen to be equally weighted.

The iShares USD Green Bond ETF (BGRN) is included to represent green bonds, whose funds are used exclusively to finance climate-related and other environmental projects. Clean energy equities are proxied by the iShares Global Clean Energy ETF (ICLN), which reflects the performance of firms engaged in renewable energy production and related technologies, while the Invesco Solar ETF (TAN) provides more targeted exposure to the solar energy sector. In addition, EU Emissions Allowances (EUA) are incorporated to represent ETS.

To capture the interaction between green assets and the broader economic and financial environment, several conventional market indicators are included. The S&P 500 index (GSPC) represents global equity market conditions and overall investor sentiment.

Gold futures (GCF) are employed as a traditional safe-haven asset, often associated with periods of heightened uncertainty and inflationary pressure, while copper futures (HGF) serve as a proxy for global industrial activity and economic growth, reflecting demand conditions in manufacturing and energy infrastructure. Natural gas futures (TTFF) are included to represent energy market dynamics and supply-side shocks, particularly relevant during periods of energy market stress.

Finally, we choose Cardano (ADA) as a proxy for clean cryptocurrencies, for being one of the largest in its class. Also, we consider Bitcoin (BTC) to oppose ADA for being the largest by market capitalisation.

4 Main results

For each asset an adequate univariate return model has been fitted. Based on the outcomes of the Basel Traffic Light Test, the unconditional coverage, independence and conditional coverage tests. GJR-GARCH model was best suited for BGRN, GCF, HGF, GSPC, ICLN and TAN assets, while csGARCH was more suitable for EUA, TTFF and ADA assets. For BTC asset the best fit was eGARCH model.

Table 3 shows the number of actual violations for different assets and their corresponding Basel zones. Assets BGRN, HGF, ICLN, TAN, EUA, TTFF, ADA and BTC have relatively low violations and fall within the Green Basel zone, suggesting that the predicted VaR aligns well with actual losses. In contrast, GCF and GSPC have higher violations and are classified in the Yellow Basel zone, signaling moderate risk and the need for closer monitoring.

Table 3: Basel 99% VaR Backtesting Results.

Asset	Backtesting days	Violations	Basel zone	Volatility model
BGRN	757	15	Green	GJR-GARCH
GCF	1237	28	Yellow	GJR-GARCH
HGF	1238	24	Green	GJR-GARCH
GSPC	1236	27	Yellow	GJR-GARCH
ICLN	1236	15	Green	GJR-GARCH
TAN	1236	21	Green	GJR-GARCH
EUA	1287	24	Green	csGARCH
TTFF	1035	12	Green	csGARCH
ADA-USD	1938	28	Green	csGARCH
BTC-USD	2249	21	Green	eGARCH

Table 4 presents the results of backtesting for Value-at-Risk (VaR) models across the assets. We present the results of Unconditional Coverage, Independence, and Conditional Coverage tests. Most assets fail to reject the null hypothesis with 5% significance. For these assets, the models predict VaR adequately, the number of exceedances matches the theoretical expectation, and they are spread out independently over time. However, for EUA, GCF and GSPC the results suggest potential inadequacy in captur-

Table 4: Conditional Coverage, Independence, and Unconditional Coverage Test Results.

Asset	Conditional Coverage Test		Independence Test		Unconditional Coverage Test		Volatility model
	VaR 95%	VaR 99%	VaR 95%	VaR 99%	VaR 95%	VaR 99%	
ADA	1.00000	0.71600	0.79600	0.85600	0.96700	0.92100	csGARCH
BGRN	0.72229	0.01668	0.06958	-	0.18094	-	GJR-GARCH
BTC	0.84100	0.87400	0.19600	0.06100	0.42600	0.17000	eGARCH
EUA	0.07691	0.00537	0.06238	0.00869	0.03685	0.00066	csGARCH
ICLN	0.61645	0.46525	0.86267	-	0.86901	-	GJR-GARCH
TAN	0.23960	0.02472	0.27600	-	0.27670	-	GJR-GARCH
GCF	0.78027	0.00013	0.69936	0.15706	0.89272	0.00024	GJR-GARCH
HGF	0.88625	0.00328	0.65271	0.48632	0.89451	0.01041	GJR-GARCH
GSPC	0.02281	0.00030	0.93145	0.61863	0.07462	0.00127	GJR-GARCH
TTFE	0.32526	0.61515	1.00000	0.12528	1.00000	0.27215	csGARCH

ing extreme tail risk. In some cases, test statistics are not reported (indicated by “-”). This occurs when the number of VaR exceedances is either zero or equal to the expected number under the nominal coverage level, which prevents the computation of likelihood-based backtesting statistics. In particular, the Independence and Conditional Coverage tests require a minimum number of violations to estimate transition probabilities between exceedance states. When this condition is not satisfied, the corresponding test is not well-defined and therefore omitted.

After selecting optimal univariate models, we proceed with constructing a portfolio and fitting appropriate Vine Copula models for the univariate residuals. By comparing several alternative portfolio structures and performing comprehensive backtesting for each, we conclude with a final model that excludes both cryptocurrencies. Backtesting indicated that Cardano (ADA) had a significant negative impact on model accuracy, its inclusion caused the violation rate to increase drastically and introduced significant dependence between violations. Furthermore, after excluding ADA and remaining with BTC, the results continued to show dependent violations, suggesting that the volatility of the crypto market is incompatible with the chosen portfolio. The 8-asset R-vine and C-vine models, consisting of traditional commodities, equities, and green energy indices, showed robust results for the 1-day-ahead horizon. As shown in Table 5, at both 1% and 5% confidence levels, the Kupiec test p-values exceed 0.05, meaning statistically significant actual violations rate alignment with expected rate. Independence and Conditional Coverage tests confirm that violations are not clustered, meaning models pass all three fundamental VaR backtesting criteria. Additionally, both vine models fall within the “Green” zone of the Basel traffic light system. Comparing the two vine structures in Table 6, it is seen that the R-vine slightly outperforms the C-vine, as demonstrated by its lower Average AIC (-2890.044) and BIC (-2801.852). Results in Table 7 illustrate that both models remain valid at the 99% confidence level (p-values of all tests are greater than 0.05). However, at the 95% level, the models show marginal performance (some p-values being less than 0.05), suggesting that while the models accurately predict VaR, they may struggle to estimate reliable losses having a breach in

VaR. Since the average AIC and BIC are lower for the R-vine, and this structure exhibits more flexibility, in the following analysis we focus on describing and interpreting results only for the R-vine model. The corresponding 1% and 5% VaR and ES estimates are illustrated in Figures 5 and 6.

Table 5: Backtesting results comparison. **Notation:** ER – expected rate of violations, AR – actual rate of violations, POF – p-value of proportion of failures (Kupiec) test, IND – p-value of independence test, CC – p-value of conditional coverage test.

Model	Horizon	Confidence	ER	AR	POF	IND	CC	Basel traffic light
R-vine	1d	99%	0.010	0.015	0.346	0.121	0.192	Green
	1d	95%	0.050	0.056	0.464	0.096	0.192	Green
C-vine	1d	99%	0.010	0.015	0.203	0.151	0.159	Green
	1d	95%	0.050	0.055	0.569	0.079	0.182	Green

Table 6: Average information criteria of both models.

Model	Average AIC	Average BIC
R-vine	-2890.044	-2801.852
C-vine	-2886.251	-2795.934

Table 7: Expected Shortfall backtesting results. **Notation:** ER – Exceedance Residual test, CCL – Conditional Calibration test, ESR – Expected Shortfall Regression test.

Model	Confidence	ER p-value	CCL p-value	ESR p-value
R-vine	99%	0.100	0.170	0.267
	95%	0.069	0.171	0.048
C-vine	99%	0.123	0.196	0.247
	95%	0.048	0.129	0.037

Table 8 describes the learned unconditional dependence structure throughout the rolling window exercise, which reveals a distinct hierarchy characterized by clear thematic clustering and asymmetric tail risks. The strongest and the most robust dependence is observed within the clean energy sector itself, with the ICLN-TAN pair exhibiting exceptionally high mean Kendall's τ of 0.76, and symmetric tail dependence. This "lockstep" behaviour indicates that despite sub-sector differences, these assets respond almost identically to systemic shocks, offering little diversification benefit against one another.

Furthermore, the observed results challenge the decoupling hypothesis regarding the broader equity market. Clean energy assets maintain a moderate-to-strong integration

Table 8: This table presents descriptive statistics for the unconditional dependence structure between asset pairs, averaged across the full rolling window estimation period. **Notation:** $\bar{\tau}$ denotes the mean unconditional Kendall’s rank correlation coefficient; $\text{sd}(\tau)$, \min , and \max represent the standard deviation, minimum, and maximum values of τ , respectively; Q_2 and Q_3 correspond to the median and 75th percentile of the correlation distribution. The columns $\bar{\lambda}_L$ and $\bar{\lambda}_U$ report the mean empirical lower and upper tail dependence coefficients, respectively, quantifying the probability of simultaneous extreme negative or positive returns.

Assets		$\bar{\tau}$	$\text{sd}(\tau)$	$\min \bar{\tau}$	$\max \bar{\tau}$	$\bar{\tau}(Q_2)$	$\bar{\tau}(Q_3)$	$\bar{\lambda}_L$	$\bar{\lambda}_U$
ICLN	TAN	0.76	0.02	0.70	0.80	0.75	0.78	0.74	0.74
GSPC	ICLN	0.42	0.02	0.35	0.47	0.40	0.43	0.38	0.37
GSPC	TAN	0.36	0.02	0.31	0.41	0.35	0.37	0.32	0.33
BGRN	GCF	0.20	0.02	0.13	0.25	0.19	0.21	0.19	0.19
GCF	HGF	0.16	0.02	0.09	0.22	0.14	0.17	0.23	0.11
HGF	GSPC	0.15	0.05	0.03	0.26	0.11	0.19	0.13	0.12
BGRN	ICLN	0.14	0.06	0.03	0.24	0.08	0.19	0.15	0.15
HGF	ICLN	0.13	0.04	0.02	0.20	0.09	0.16	0.13	0.10
BGRN	TAN	0.12	0.05	0.02	0.22	0.07	0.16	0.14	0.13
BGRN	GSPC	0.11	0.05	0.00	0.20	0.07	0.15	0.12	0.13
HGF	TAN	0.11	0.04	0.01	0.19	0.08	0.14	0.12	0.09
GCF	ICLN	0.11	0.03	0.04	0.18	0.09	0.13	0.11	0.13
GCF	TAN	0.08	0.02	0.01	0.14	0.06	0.10	0.10	0.12

with the S&P 500, with mean τ ranging from 0.36 to 0.42. Notably, the estimated Vine Tree structures position ICLN as the central bridge connecting the broader equity market to the specialized green sector (TAN). This suggests that clean energy assets currently are still sensitive to general market sentiment, rather than act as isolated alternative asset class. Given the reliance of green innovation on technological breakthroughs, it is not surprising that these assets exhibit the volatility profiles typical of the technology sector. However, partial decoupling may emerge as the industry matures from a speculative growth phase into established critical infrastructure, governed by regulatory and demand drivers.

Furthermore, we observe a pronounced asymmetry in the commodity-green nexus. While Gold (GCF) and Copper (HGF) display a weak average correlation ($\bar{\tau} = 0.16$), their tail behaviour reveals a hidden systemic risk. The probability of joint crashes ($\bar{\lambda}_L = 0.23$) is more than double the probability of joint growth ($\bar{\lambda}_U = 0.11$). This “panic coupling” implies that industrial metals and safe-haven assets, which typically follow different cycles [23], tend to re-couple dangerously during severe market downturns, likely amplifying liquidity stress for green technologies reliant on these raw materials. Thus Gold cannot be confidently relied upon as a perfect hedge for green technology during systemic crises.

Note, that in contrast to the equity-heavy clean energy sector, Green Bonds (BGRN) emerge as the most effective diversifier. The Table 8 highlights consistently low correlations with both the S&P 500 ($\bar{\tau} = 0.11$) and clean energy equities ($\bar{\tau} = 0.14$). Interestingly, BGRN shows a stronger relative linkage to Gold ($\bar{\tau} = 0.2$), suggesting its

performance is driven more by interest rate and safe-haven flows than by the "risk-on" technology cluster. This validates the role of Green Bonds as a portfolio stabilizer.

Moving beyond the static averages, the dynamic Vine Copula analysis demonstrates a profound structural shift in how green markets transmit risk. As illustrated by Figure 2, ICLN acts as the dominant hub for the majority of the sample, reinforcing its role as the central transmitter of shocks in the green economy. In the beginning of the sample, potentially strongly affected by the 2020–2022 volatility, the market dynamics shift dramatically over time, but in the latter half of the sample the network topology locks into a stable R-Vine structure. When ICLN moves, it pulls the entire network – S&P 500, Solar stocks, and Green Bonds. This can be observed in Figure 2, that shows persistently high and stable dependency between ICLN and its satellites.

However, a secondary, unstable hub emerges around Gold. Our analysis indicates that Gold's ability to act as a central node is fragile and directly linked to the Carbon market (EUA). The presence and signal strength of Carbon allowances acts as a switch, i.e., only when EUA signals are active in the structure does Gold successfully consolidate the commodity cluster. However, as seen by Figure 2, this relationship is volatile. The connection between the Gold and EUA "flickers", leading to periods where the commodity hub disintegrates. Notably, during these low-magnitude periods the conservative independence tests often fail to reject the null, rendering some dependency estimates statistically indistinguishable from zero. We deem that this statistical invisibility does not imply a lack of economical linkage, but rather the existence of a mixture of opposing forces, pulling the market in different directions. On a pro-cyclical movement, the industrial demand pulls carbon prices up when the economy (along with the commodities) are growing. On a counter-cyclical aspect, the regulatory intervention and "green hedging" demand can push prices up even when the broad market falls. The cancelling out of these vectors may result in a near-zero net correlation in certain periods. This erratic behaviour highlights the lack of maturity of the investor base and sentiment surrounding regulatory assets, as the market has not yet settled on a consistent pricing model for carbon risk.

Importantly, we observe a distinct regime change around the rolling window corresponding to June 2024. Note, that the observed date is lagging due to the large training window selected for the rolling window experiment. However, this pivot point marks the interesting transition of the green market from a previous "crisis response" regime to a "structural integration" regime. Judging from the stress index, presented by Figure 4, around that point a peak was reached, and then sharply declined. In the context of post-pandemic volatility and the 2022 energy crisis, the observed decline suggests that the recent market stability in 2023–2024 has begun to dilute the impact of the earlier shocks. As the immediate correlation between fossil fuels and green assets weakens in the recent data, the network decouples from the geopolitical conflict signals. Paradoxically, while the average correlation dropped, the structural risk increased, at which the network confidently shifts to a centralised hub structure (R-Vines). The structural hardening is confirmed by the average path length metric, see Figure 1, which drops significantly in the post-June 2024 period. In Network theory, a shorter path length implies a faster transmission mechanism of the shocks in the network. Thus, while the market is less stressed on the average, it has become more efficient at transmitting shocks. A fail-

ure in the central green hub (through ICLN) now propagates to the periphery (through Bonds and Commodities) more rapidly than it did during the early crisis years.

Fig. 1 Average path length in the estimated vine structure at ever point of the rolling window experiment.

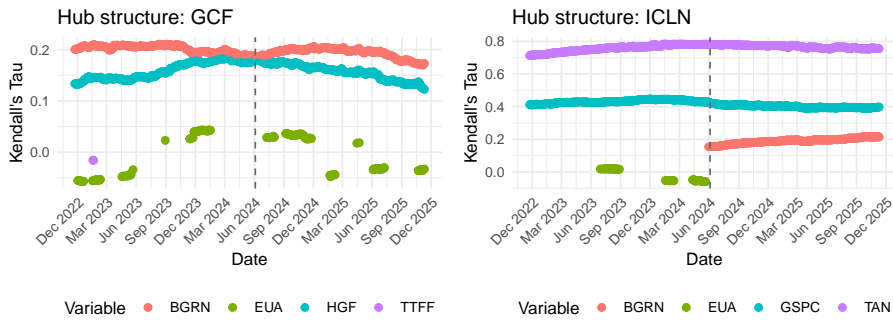
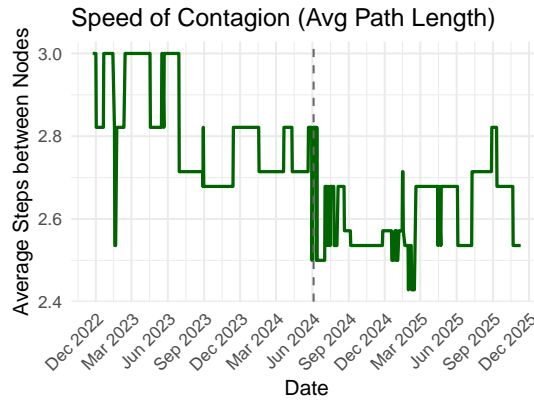


Fig. 2: The figure demonstrates the Kendall's Tau values around two hubs throughout the rolling window experiment. Namely, the GCF (**left**) and ICLN (**right**). The colors denote different variables, while the black vertical line denotes the observed structural shift around June 2024.

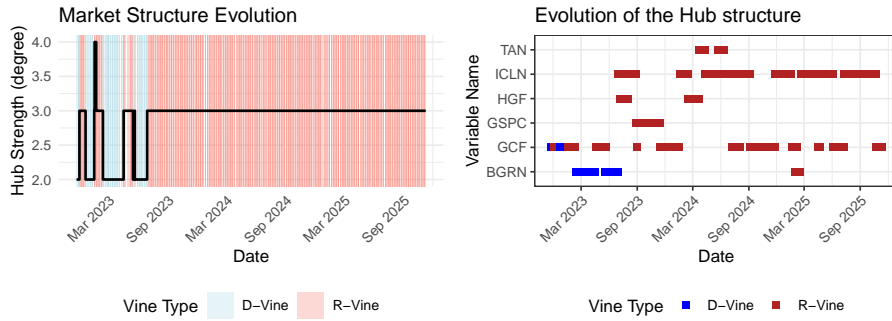


Fig. 3: The topological evolution of the market network, showing the unstable switch from a D-Vine structure into R-Vine structures (**left**), when the hubs start to consistently appear. In order to better understand the dynamic changes, on the **right** graph we present the evolving "hubs".

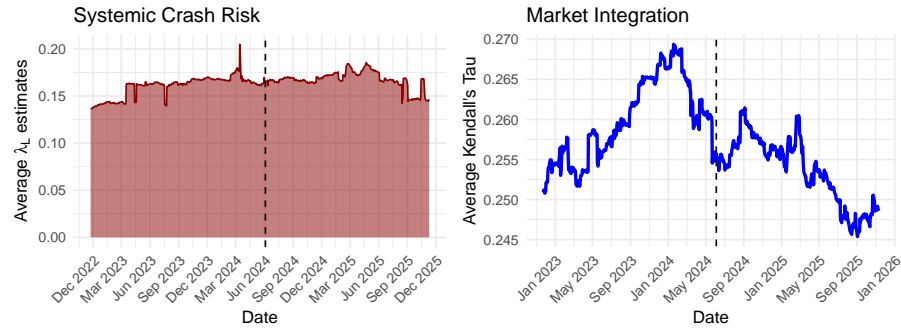


Fig. 4: On the **left** graph we demonstrate the average λ_L values of the whole vine copula structure and its dynamics throughout the rolling window experiments. Similarly, on the **right** graph we present the average Kendall's Tau values. In both cases the black vertical line denotes the observed structural shift around June 2024.

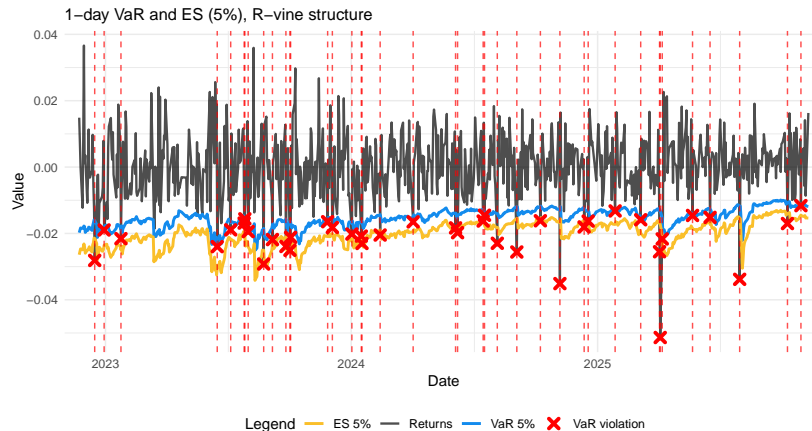


Fig. 6: 1-day VaR and ES (5%) of R-vine structure.

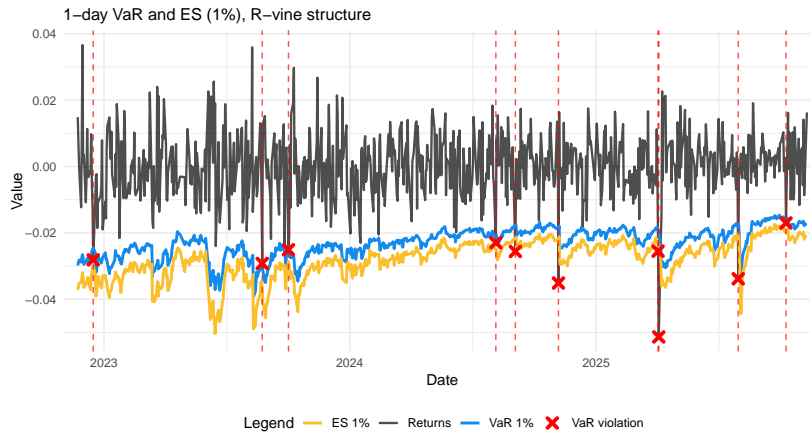


Fig. 5: 1-day VaR and ES (1%) of R-vine structure

Conclusions

In this Chapter, we model the risk dynamics and dependence structure of a portfolio composed of green financial assets, traditional commodities, and broad market indices. By employing a rolling-window ARMA-GARCH-Vine Copula framework, we identified the specific transmission channels of financial stress and evaluated the diversification potential of the green economy.

The univariate analysis confirms that green finance is not a monolithic asset class. We found that sustainable instruments exhibit highly distinct risk behaviours, ranging from the stable, mature profiles of Green Bonds, to the complex, long-memory volatility of Carbon allowances. Our risk modelling demanded the exclusion of clean cryptocurrencies from the final portfolio. Both Cardano and Bitcoin exhibited extreme volatility and dependent VaR violations.

The multivariate dependence analysis shows that the green assets are moving from their earlier speculative phase and is becoming a more accepted part of the global financial system. Our results reject the "decoupling hypothesis" as a universal characteristic of green finance. Instead, we observe a clear division. Clean Energy equities (ICLN) and Solar stocks (TAN) move in lockstep and maintain moderate integration with the S&P 500. With the Vine methodology the Clean Energy ETF is identified as a central "risk hub", effectively transmitting broad market shocks into the specialized green sector. In contrast, Green Bonds have successfully decoupled, behaving more like stabilizing fixed-income instruments than speculative technology assets, validating their role as effective portfolio diversifiers.

The analysis of traditional commodities reveals hidden tail risks. While Gold and Copper display a weak average correlation, we observe strong asymmetric tail dependence. The probability of simultaneous extreme losses is more than double the probability of simultaneous gains. Such "panic coupling" suggests that during the severe market

downturns, industrial and safe-haven commodities synchronize, significantly reducing the hedging efficiency of Gold when it is most needed. Furthermore, the relationship between Gold and Carbon allowances was found to be unstable, too weak to form a consistent commodity hub.

The dynamic rolling-window analysis identified a structural shift in the market, observable in our results since around June 2024. The Vine topology transitioned from an unstable sequential D-Vine structure to a centralised R-Vine structure dominated by Clean Energy ETF. Coinciding with this shift, the average path length of the network dropped significantly, indicating that the transmission of shocks has become more efficient. While the overall market volatility has declined since the 2022 energy crisis, the market structure has become observably more centralised. The results suggest that shocks to the clean energy hub are likely to spread to other assets more rapidly than in previous regimes.

These findings have several implications for investors. Green Bonds stand out as an effective hedge, maintaining low and stable correlations with both the broader equity market and clean energy assets. Gold, despite its safe-haven reputation, does not consistently provide protection during systemic downturns, as it tends to move together with industrial metals (proxied by copper) during market stress. Also, the expanding importance of ICLN raises caution, as shocks to this single ETF may now spread more quickly across the green asset market. Meanwhile, carbon credits should be treated carefully, given their unstable behaviour and sensitivity to regulatory changes.

Finally, the results highlight several directions for future research. First, the growing centralization of the green financial network around Clean Energy ETFs suggests the need to study the velocity of shock transmission. Future studies should consider additional connectedness measures, such as Diebold-Yilmaz spillover indices or Causal Vine copulas, potentially across a wider variety of assets, including upstream supply chain equities and sovereign green bonds to more accurately map the evolving topology of sustainable finance.

Furthermore, special attention should be paid to validating the drivers of the observed structural shift in dependence. While this study posits that the shift post-2024 stems from the "washing-out" of post-pandemic-era volatility, future work should employ formal structural break tests to pinpoint the exact timing of the decoupling without the inherent lag effects due to the rolling window estimation.

Finally, the exclusion of cryptocurrencies and the backtesting challenges associated with certain commodities suggest that standard volatility models are insufficient for the most volatile assets. Future research could employ Markov-Switching GARCH or Extreme Value Theory to explicitly capture regime-dependent tail risks. Such methodological advancements would potentially allow for the safe inclusion of high-volatility instruments, enabling the construction of larger, more resilient portfolios than the one considered in this work.

References

1. Aas, K., Czado, C., Frigessi, A., & Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2), 182–198. <https://doi.org/10.1016/j.insmatheco.2007.11.001>
2. Aparicio, F. M., & Estrada, J. (2001). Empirical distributions of stock returns: European securities markets, 1990–95. *The European Journal of Finance*, 7(1), 1–21. <https://doi.org/10.1080/13518470122131>
3. Arif, M., Naeem, M. A., Farid, S., Nepal, R., & Jamasb, T. (2022). Diversifier or more? Hedge and safe haven properties of green bonds during COVID-19. *Energy Policy*, 168, Article 113102. <https://doi.org/10.1016/j.enpol.2022.113102>
4. Basel Committee on Banking Supervision. (1996). *Supervisory framework for the use of backtesting in conjunction with the internal models approach to market risk capital requirements* (Tech. Rep.). Bank for International Settlements.
5. Baur, D. G., & McDermott, T. K. (2010). Is gold a safe haven? International evidence. *Journal of Banking & Finance*, 34(8), 1886–1898. <https://doi.org/10.1016/j.jbankfin.2009.12.008>
6. Bayer, S., & Dimitriadis, T. (2020). Regression-based expected shortfall backtesting. *Journal of Financial Econometrics*, 20(3), 437–471. <https://doi.org/10.1093/jfinec/nbaa013>
7. Becker, M. G., Martin, F., & Walter, A. (2022). The power of ESG transparency: The effect of the new SFDR sustainability labels on mutual funds and individual investors. *Finance Research Letters*, 47, Article 102708. <https://doi.org/10.1016/j.frl.2022.102708>
8. Bedford, T., & Cooke, R. M. (2002). Vines—A new graphical model for dependent random variables. *The Annals of Statistics*, 30(4), 1031–1068. <https://doi.org/10.1214/aos/1031689016>
9. Ben Ameer, H., Ftiti, Z., Louhichi, W., & Yousfi, M. (2024). Do green investments improve portfolio diversification? Evidence from mean conditional value-at-risk optimization. *International Review of Financial Analysis*, 94, Article 103255. <https://doi.org/10.1016/j.irfa.2024.103255>
10. Bhutta, U. S., Tariq, A., Farrukh, M., Raza, A., & Iqbal, M. K. (2022). Green bonds for sustainable development: Review of literature on development and impact of green bonds. *Technological Forecasting and Social Change*, 175, Article 121378. <https://doi.org/10.1016/j.techfore.2021.121378>
11. Bollerslev, T. (1987). A conditionally heteroskedastic time series model for speculative prices and rates of return. *The Review of Economics and Statistics*, 69(3), 542–547. <https://doi.org/10.2307/1925546>
12. Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307–327. [https://doi.org/10.1016/0304-4076\(86\)90063-1](https://doi.org/10.1016/0304-4076(86)90063-1)
13. Cepni, O., Demirer, R., & Rognone, L. (2022). Hedging climate risks with green assets. *Economics Letters*, 212, Article 110312. <https://doi.org/10.1016/j.econlet.2022.110312>
14. Chaiyawat, T., & Guayjarernpanishk, P. (2025). Enhancing insurer portfolio resilience and capital efficiency with green bonds: A framework combining dynamic R-vine copulas and tail-risk modeling. *Risks*, 13(9), Article 163. <https://doi.org/10.3390/risks13090163>
15. Cherubini, U., Mulinacci, S., Gobbi, F., & Romagnoli, S. (2011). *Dynamic copula methods in finance*. John Wiley & Sons.
16. Christoffersen, P. F. (1998). Evaluating interval forecasts. *International Economic Review*, 39(4), 841–862. <https://doi.org/10.2307/2527341>
17. Climate Bonds Initiative. (2024). *Green bond dataset methodology* (Tech. Rep.). Climate Bonds Initiative.
18. Costanzino, N., & Curran, M. (2018). A simple traffic light approach to backtesting expected shortfall. *Risks*, 6(1), Article 2. <https://doi.org/10.3390/risks6010002>
19. de Freitas Netto, S. V., Sobral, M. F. F., Ribeiro, A. R. B., & Soares, G. R. L. (2020). Concepts and forms of greenwashing: A systematic review. *Environmental Sciences Europe*, 32, Article 19. <https://doi.org/10.1186/s12302-020-0300-3>
20. Demiralay, S., Gencer, H. G., & Bayraci, S. (2024). Time-scale behaviour of co-movements between renewable energy stocks and other financial assets. In *Transition to the circular economy model: The case of Turkey* (pp. 105–132). Springer. https://doi.org/10.1007/978-3-031-43283-5_6

21. Engle, R. (2002). Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business & Economic Statistics*, 20(3), 339–350. <https://doi.org/10.1198/073500102288618487>
22. Engle, R. F., & Lee, G. G. J. (1999). A long-run and short-run component. In *Cointegration, causality, and forecasting* (p. 475). Oxford University Press.
23. Erb, C. B., & Harvey, C. R. (2006). The strategic and tactical value of commodity futures. *Financial Analysts Journal*, 62(2), 69–97. <https://doi.org/10.2469/faj.v62.n2.4083>
24. Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *The Journal of Finance*, 48(5), 1779–1801. <https://doi.org/10.1111/j.1540-6261.1993.tb05128.x>
25. Guo, J. (2018). Co-movement of international copper prices, China's economic activity, and stock returns: Structural breaks and volatility dynamics. *Global Finance Journal*, 36, 62–77. <https://doi.org/10.1016/j.gfj.2017.10.002>
26. Gursoy, S., Kilic, E., Eksi, I. H., Yudaruddin, R., & Tabash, M. I. (2024). Dynamic volatility interactions between sustainable crypto currencies and green economy indicators. In *2024 International Conference on Sustainable Islamic Business and Finance (SIBF)* (pp. 244–249). IEEE. <https://doi.org/10.1109/SIBF61715.2024.10543632>
27. Hamilton, J.D. (2009). *Causes and consequences of the oil shock of 2007-08* (Tech. Rep. No. w15002). National Bureau of Economic Research. <https://doi.org/10.3386/w15002>
28. Hong, Z. (2025). Risks and prevention methods in green finance. *SHS Web of Conferences*, 218, Article 03018. <https://doi.org/10.1051/shsconf/202521803018>
29. Hu, Q., & Gu, Y. (2024). Copper economic dynamics: Navigating resource scarcity, price volatility, and green growth. *Resources Policy*, 89, Article 104462. <https://doi.org/10.1016/j.resourpol.2023.104462>
30. Ji, H., Wang, H., & Liseo, B. (2018). Portfolio diversification strategy via tail-dependence clustering and ARMA-GARCH vine copula approach. *Australian Economic Papers*, 57(3), 265–283. <https://doi.org/10.1111/1467-8454.12119>
31. Joe, H. (1997). *Multivariate models and multivariate dependence concepts*. CRC Press.
32. Joe, H. (2014). *Dependence modeling with copulas*. CRC Press.
33. Karoglou, M. (2010). Breaking down the non-normality of stock returns. *The European Journal of Finance*, 16(1), 79–95. <https://doi.org/10.1080/13518470903036230>
34. Kirikkaleli, D. (2021). Analyses of wavelet coherence: Financial risk and economic risk in China. *Journal of Financial Economic Policy*, 13(5), 587–599. <https://doi.org/10.1108/JFEP-03-2020-0061>
35. Kristoufek, L. (2015). What are the main drivers of the Bitcoin price? Evidence from wavelet coherence analysis. *PLOS ONE*, 10(4), Article e0123923. <https://doi.org/10.1371/journal.pone.0123923>
36. Kupiec, P. H. (1995). Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives*, 3(2), 73–84. <https://doi.org/10.3905/jod.1995.407942>
37. Lyu, C., & Scholtens, B. (2024). Integration of the international carbon market: A time-varying analysis. *Renewable and Sustainable Energy Reviews*, 191, Article 114102. <https://doi.org/10.1016/j.rser.2023.114102>
38. Madani, M. A., & Ftiti, Z. (2022). Is gold a hedge or safe haven against oil and currency market movements? A revisit using multifractal approach. *Annals of Operations Research*, 313(1), 367–400. <https://doi.org/10.1007/s10479-021-04212-3>
39. McNerney, C., & Bunn, D. W. (2019). Expansion of the investor base for the energy transition. *Energy Policy*, 129, 1240–1244. <https://doi.org/10.1016/j.enpol.2019.03.037>
40. McNeil, A. J., & Frey, R. (2000). Estimation of tail-related risk measures for heteroscedastic financial time series: An extreme value approach. *Journal of Empirical Finance*, 7(3), 271–300. [https://doi.org/10.1016/S0927-5398\(00\)00012-8](https://doi.org/10.1016/S0927-5398(00)00012-8)
41. Mejdoub, H., & Ghorbel, A. (2018). The dynamic relationship between oil prices and returns on renewable energy companies. *American Journal of Finance and Accounting*, 5(2), 173–192. <https://doi.org/10.1504/AJFA.2018.091321>
42. Mensi, W., Belghouthi, H. E., Al-Kharusi, S., & Kang, S. H. (2025). Tail risk contagion and connectedness between clean cryptocurrency, green assets and commodity markets. *International Review of Financial Analysis*, 105, Article 103823. <https://doi.org/10.1016/j.irfa.2025.103823>

43. MSCI ESG Research LLC. (2023). *ESG ratings methodology*. MSCI.
44. Naeem, M. A., Adekoya, O. B., & Oliyide, J. A. (2021). Asymmetric spillovers between green bonds and commodities. *Journal of Cleaner Production*, 314, Article 128100. <https://doi.org/10.1016/j.jclepro.2021.128100>
45. Naeem, M. A., Bouri, E., Costa, M. D., Naifar, N., & Shahzad, S. J. H. (2021). Energy markets and green bonds: A tail dependence analysis with time-varying optimal copulas and portfolio implications. *Resources Policy*, 74, Article 102434. <https://doi.org/10.1016/j.resourpol.2021.102434>
46. Nelsen, R. B. (2006). *An introduction to copulas* (2nd ed.). Springer. <https://doi.org/10.1007/0-387-28678-0>
47. Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*, 59(2), 347–370. <https://doi.org/10.2307/2938260>
48. Network for Greening the Financial System. (2024). *The green transition and the macroeconomy: A monetary policy perspective* (Tech. Rep.). NGFS.
49. Nguyen, T. T. H., Naeem, M. A., Balli, F., Balli, H. O., & Vo, X. V. (2021). Time-frequency comovement among green bonds, stocks, commodities, clean energy, and conventional bonds. *Finance Research Letters*, 40, Article 101739. <https://doi.org/10.1016/j.frl.2020.101739>
50. Nolde, N., & Ziegel, J. F. (2017). Elicitability and backtesting: Perspectives for banking regulation. *The Annals of Applied Statistics*, 11(4), 1833–1874. <https://doi.org/10.1214/17-AOAS1041>
51. Omeir, A. K., Štreimikienė, D., & Vasiliauskaitė, D. (2025). Sustainable investments: Assessment of risks. *Journal of Business Economics and Management*, 26(3), 576–598. <https://doi.org/10.3846/jbem.2025.23412>
52. Reboredo, J. C. (2015). Is there dependence and systemic risk between oil and renewable energy stock prices? *Energy Economics*, 48, 32–45. <https://doi.org/10.1016/j.eneco.2014.12.009>
53. Reboredo, J. C. (2018). Green bond and financial markets: Co-movement, diversification and price spillover effects. *Energy Economics*, 74, 38–50. <https://doi.org/10.1016/j.eneco.2018.05.030>
54. Reboredo, J. C., & Ugolini, A. (2020). Price connectedness between green bond and financial markets. *Economic Modelling*, 88, 25–38. <https://doi.org/10.1016/j.econmod.2019.09.004>
55. Saeed, T., Bouri, E., & Tran, D. K. (2020). Hedging strategies of green assets against dirty energy assets. *Energies*, 13(12), Article 3141. <https://doi.org/10.3390/en13123141>
56. Sharif, A., Brahim, M., Dogan, E., & Tzeremes, P. (2023). Analysis of the spillover effects between green economy, clean and dirty cryptocurrencies. *Energy Economics*, 120, Article 106606. <https://doi.org/10.1016/j.eneco.2023.106606>
57. Silvennoinen, A., & Teräsvirta, T. (2009). Multivariate GARCH models. In *Handbook of financial time series* (pp. 201–229). Springer. https://doi.org/10.1007/978-3-540-71297-8_9
58. Sklar, M. (1959). Fonctions de répartition à n dimensions et leurs marges. *Annales de l'ISUP*, 8(3), 229–231.
59. Su, Y. H., Rizvi, S. K. A., Umar, M., & Chang, H. (2023). Unveiling the relationship between oil and green bonds: Spillover dynamics and implications. *Energy Economics*, 127, Article 107055. <https://doi.org/10.1016/j.eneco.2023.107055>
60. Suimon, Y. (2024). Copper as a central commodity in network analysis of price dynamics and its connection to macroeconomic indicators. In *17th International Congress on Advanced Applied Informatics (IIAI-AAI-Winter)* (pp. 136–141). IEEE. <https://doi.org/10.1109/IIAI-AAI-Winter64350.2024.00034>
61. Vacha, L., & Barunik, J. (2012). Co-movement of energy commodities revisited: Evidence from wavelet coherence analysis. *Energy Economics*, 34(1), 241–247. <https://doi.org/10.1016/j.eneco.2011.10.007>
62. Villar-Rubio, E., Huete-Morales, M. D., & Galán-Valdivieso, F. (2023). Using EGARCH models to predict volatility in unconsolidated financial markets: The case of European carbon allowances. *Journal of Environmental Studies and Sciences*, 13(3), 500–509. <https://doi.org/10.1007/s13412-023-00839-4>
63. Zakeri, B., Staffell, I., Dodds, P. E., Grubb, M., Ekins, P., Jääskeläinen, J., Cross, S., & Rinne, E. (2023). The role of natural gas in electricity prices in Europe. *Energy Reports*, 10, 2778–2792. <https://doi.org/10.1016/j.egy.2023.09.055>
64. Zerbib, O. D. (2025). The effect of pro-environmental preferences on bond prices: Evidence from green bonds. In *Handbook of quantitative sustainable finance* (pp. 410–448). Chapman and Hall/CRC.

65. Zhang, R. (2025). Green finance and ESG investment strategies under climate risk management. *Economics and Management Innovation*, 2(1), 140–148. <https://doi.org/10.62051/emi.v2i1.42>
66. Zhang, Y., & Umair, M. (2023). Examining the interconnectedness of green finance: An analysis of dynamic spillover effects among green bonds, renewable energy, and carbon markets. *Environmental Science and Pollution Research*, 30(31), 77093–77109. <https://doi.org/10.1007/s11356-023-27715-x>
67. Zhou, G. (1993). Asset-pricing tests under alternative distributions. *The Journal of Finance*, 48(5), 1927–1942. <https://doi.org/10.1111/j.1540-6261.1993.tb05135.x>

About authors

Eugenijus Gabrielius Ivanauskas is a first-year Master’s student in Mathematics at the same faculty. In 2025, he completed his Bachelor’s degree in Econometrics at Vilnius University, graduating cum laude. His main interests include mathematical modelling, statistics, and financial risk, which he plans to further develop through PhD studies after 2027. Alongside the studies, he works at Danske Bank in Model Risk Management, where he validates quantitative models related to financial risk, including ISDA SIMM, Value-at-Risk models for asset management, and CLO pricing frameworks. His experience combines a solid theoretical background with practical applications in finance.

Liepa Urbonaitė is a second-year Bachelor’s student in Data Science at Vilnius University. Her primary academic interests lie in mathematics and statistics, which form the core of her studies and future aspirations. She plans to further deepen her knowledge in these areas through Master’s studies. Alongside her academic work, she is employed as a tutor, teaching chemistry, mathematics, and physics. This role allows her to strengthen her analytical thinking while helping others build a solid foundation in scientific subjects. Her experience reflects a strong combination of theoretical understanding and practical application in problem-solving. She is also a recipient of multiple national awards in chemistry and has represented Lithuania in international competitions, earning bronze and silver medals at the International Junior Science Olympiad and the European Olympiad of Experimental Sciences.

Saulius Jokubaitis is an assistant professor at Vilnius University, Institute of Applied Mathematics, Faculty of Mathematics and Informatics. He holds a PhD in Mathematics from Vilnius University. His research focus is machine learning, econometrics, time-series analysis and risk modelling. In addition to his academic work, he serves as a consultant in the financial sector, applying advanced risk modelling techniques to real-world business challenges.

Fault Detection in Solar Power Plants Based on Energy Production Data

Dominykas Vilčinskas,
Lukas Voveris, and
Jolita Bernatavičienė

Abstract

This research addresses the critical need for the timely identification of faults in solar power plants to minimize electricity loss. The study analyses energy production data from a Lithuanian solar power plant, comprising 143 strings distributed across 12 inverters, over a 19-month period. During data preprocessing, 16 key features were extracted from each string's time series data to represent the global structure of the data. The extraction process resulted in a transformed dataset, where each time series is represented as an object with 16 features, enabling more effective analysis. Statistical and machine learning techniques - including PCA + α -HULL, Isolation Forest (iForest), and Local Outlier Factor (LOF) - were employed to identify systems exhibiting abnormal behavior. The results demonstrate that a combination of these methods can help effectively identify outliers, with a combined anomaly score providing a comprehensive assessment of string performance. Additionally, RANSAC and DBSCAN methods were used to construct fault profiles, which enabled a more in-depth analysis of each system's performance and provided further confirmation of previously identified systems exhibiting abnormal behavior.

Keywords: solar power plants; fault detection; anomaly detection; energy production; time series features; PCA; α -HULL; Isolation Forest; Local Outlier Factor; DBSCAN

Dominykas Vilčinskas and Lukas Voveris
Vilnius University, Institute of Applied Mathematics, Naugarduko str. 24, 03225 Vilnius, Lithuania
e-mail: dominykas.vilcinskas, lukas.voveris@mif.stud.vu.lt
Jolita Bernatavičienė
Vilnius University, Institute of Data Science and Digital Technologies, Akademijos str. 4, 03225 Vilnius, Lithuania
e-mail: jolita.bernatavicienne@mif.vu.lt

1 Introduction

Solar power is increasingly prevalent, largely driven by growing concerns about climate change and environmental sustainability. Its widespread acceptance stems from its reputation as a practical solution to rising energy demands without worsening environmental degradation. Governments, corporations, and individuals are turning to solar energy for its clean, renewable characteristics and its potential for long-term cost savings. However, solar power plants face various challenges: according to [1], without proper maintenance, PV power plants have a relatively high likelihood of operating unsatisfactorily, resulting in energy losses of up to 10%. Therefore, numerous studies have been conducted to identify anomalies and issues in solar power plants, as early detection and mitigation increase overall plant efficiency. Solar power plants are susceptible to various malfunctions. Vishwakarma et al. [2] categorized faults into two main groups: acute and chronic. One subset of chronic faults is shadowing. According to [3], shadows reduce both the anticipated power and the actual power output of the shaded system. Shadowing can be a persistent problem when solar irradiance is sufficiently high.

The study [4] evaluates the performance of a solar energy generation system by comparing its energy output against a reference dataset. For any solar panel system of interest, they used either global irradiance data or the energy output of a nearby system with the same capacity and specifications as a reference. Under ideal conditions, the observed solar panel system is expected to exhibit a nearly linear relationship with the reference source, except for noise arising from data input errors. To compare any system with a reference source, the Random Sample Consensus (RANSAC) method is used. Tsafarakis and van Sark conducted further studies on the same methodology [5]. The authors have developed a new algorithm that creates a reference dataset by using the data of other PV systems in the surrounding area. The authors then used RANSAC to identify outlying energy production data points. The distribution of these points was further analysed by plotting them on a scatter plot, categorized by the hour and time of day, across every day of the year. Since there were anomalous points that were not consistent in a scatter plot based on hourly and daily time (marking them as noise), "Density-Based Spatial Clustering of Applications with Noise" (DBSCAN) [6] algorithm was used to eliminate the unwanted noise. Detected clusters are then grouped to form a profile of shadow affecting the solar panel. This profile allows for the identification of the extent and intensity of shadowing. Since solar energy production typically comes as a type of time series, other research [7] has been conducted on anomaly detection in time series, which could be useful in this context. In the paper, the authors sought to identify servers exhibiting unusual behavior by detecting anomalous patterns. The performance of each server is characterized using univariate time-series data. The article presents an idea that extracts 18 scalar features from distinct time series and then applies dimensionality reduction methods, such as Principal Component Analysis (PCA) [8], to identify patterns across these collections. This simplification enables the use of various high-dimensional outlier detection algorithms to identify anomalous time series.

This paper examines fault detection in solar power plants using energy production data. The proposed approach uses features extracted from time-series data of solar panel

strings to identify systems exhibiting abnormal behavior. The study compares the performance of three anomaly detection methods: PCA with α -HULL, Isolation Forest, and Local Outlier Factor. In addition, fault profiles are constructed for the most anomalous systems using RANSAC and DBSCAN, allowing a more detailed analysis of their behavior. The remainder of the paper is organized as follows. The second section describes the dataset and preprocessing steps. The third section explains the methodology. The fourth section presents and compares the results. Finally, the last section discusses the main conclusions and limitations of the study.

2 Data

Solar power plant data has been received from a Lithuanian company. This dataset includes detailed information on the power plant's solar energy production. There are twelve different inverters. Each inverter converts the direct current generated by a set of solar panel strings—the number of strings connected to each inverter may vary, ranging from 10 to 13. Inverters 1, 2, 3, 9, 10 have 13 strings. Inverters 4, 5, 8, 11 have 12 strings. Inverters 6, 7, 12 have 10 strings. In total, five inverters have 13 strings, four have 12, and three have 10.

Each string has exactly 34 solar panels, all of the same specifications. Energy production data for each string in the power plant are provided. Each record consists of the following features: timestamp, electric current (A), and voltage (Vdc). There are precisely 57788 records collected between March 1, 2022, and October 24, 2023, each 15 minutes apart and describing the energy generation of 143 different strings in the solar power plant.

Having electrical current (A) and voltage (Vdc) of some solar panel string at a given timestamp, the total energy generated by a string for that specific timestamp could be described as electrical power (W), which can be calculated using the given formula:

$$P = V \cdot I, \quad (1)$$

here

- P - electrical power (W),
- V - voltage (V),
- I - electrical current (A).

Figure 1 presents a comparative analysis of daily energy generation for each string across selected months: December 2022, January, February, June, July, and August 2023. These months were chosen to capture the contrasting patterns in energy production between the colder, low-solar-irradiance months and the peak solar activity during summer. For each month, the days illustrated correspond to the peak daily energy generation observed across the entire power plant. The data reveal a seasonal pattern, with energy production significantly lower in winter than in summer. Moreover, during the summer period, discrepancies between individual strings become more evident, as underperforming strings, those producing less electricity than the majority, stand out

more. In contrast, during the winter months, such deviations are minimal or nearly absent. This comparative analysis emphasizes both the seasonal impact on overall energy output and the variability in relative performance among the strings under different solar irradiance conditions.

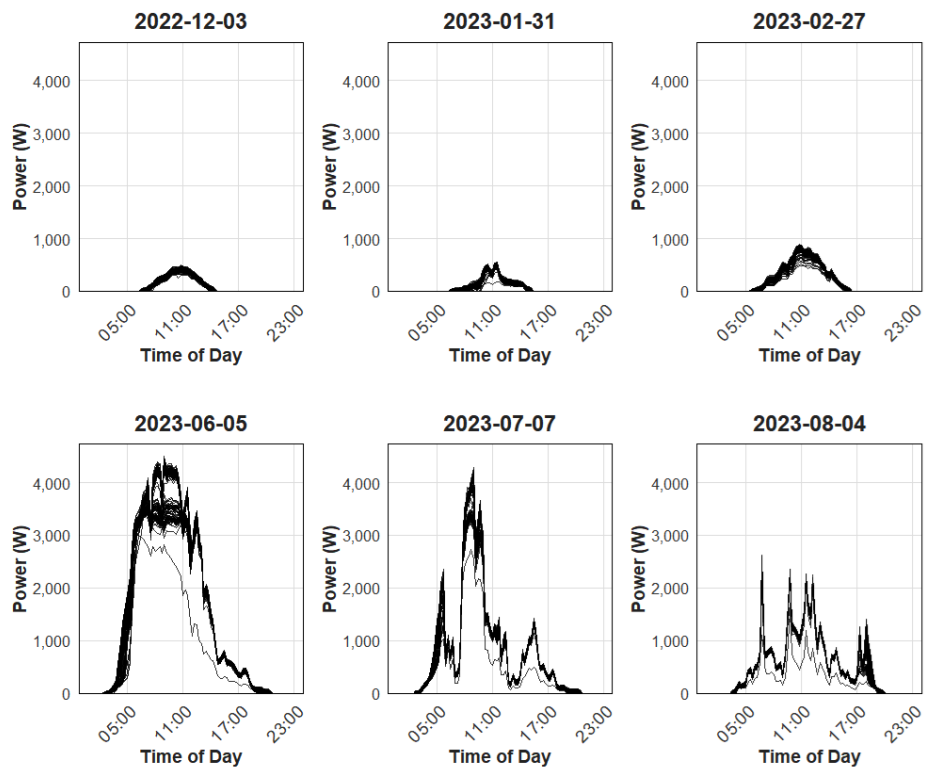


Fig. 1: Example of energy generation for each string in the solar power plant during different months.

Table 1 presents descriptive statistics for the total energy generated by solar panel strings. In fact, the 1st quartile Q_1 shows us that only 25 % of strings generated less than 43000 kWh. The similarity between the mean and median values suggests that there are no outliers based solely on total energy generated, as the mean is typically highly influenced by their presence.

Further, the total energy generated by the power plant during the same months was calculated, as shown in Figure 1. The doughnut plot presented in Figure 2 shows the distribution of the total generated energy across all four seasons. The largest share of

Table 1: Statistics of Total Power (kW)

Min.	Q_1	Median	Mean	Q_3	Max.
27826	43123	44148	44328	45876	48058

energy was generated in summer, accounting for 44.4% of the total annual production. Spring contributed 36.4%, autumn 14.9%, while winter represented the smallest share at 4.3%. These results show a clear seasonal pattern in the power plant's energy generation, with most electricity produced during periods of higher solar irradiance and longer daylight duration.

Seasonal share of total generated energy

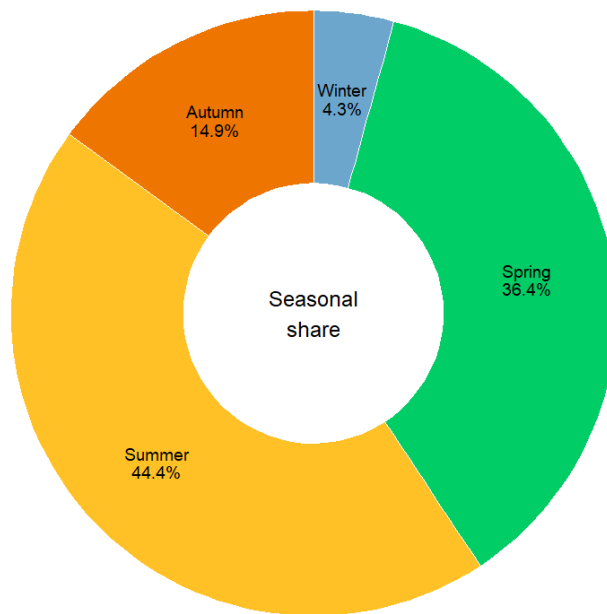


Fig. 2: Seasonal share of total generated energy of the entire power plant.

A brief data-quality analysis was performed before the anomaly-detection stage. The dataset contained 748 timestamps for which all measurements were missing across all 143 strings. Because neither electric current nor voltage was measured, 213,928 missing values were observed. The missing observations were grouped into three consecutive time intervals: 2022-05-24 22:30:00 to 2022-06-01 13:00:00, 2022-09-15 14:00:00 to 2022-09-15 15:00:00, and 2023-06-29 12:15:00 to 2023-06-29 15:00:00. To preserve continuity of the time series, the missing values were imputed by averaging the corresponding observations from the same time one day before and one day after the missing interval. This approach was chosen because the data are strongly periodic and neighboring days preserve the main daily production pattern.

An additional issue concerned negative current values. Across all strings, 54,690 cases of negative electric current were observed. Since negative current values do not represent meaningful production in this context and are likely due to inverter behavior during non-generating periods or measurement noise, these values were replaced with zero before power calculations. This preprocessing step ensured that the derived power values reflected only non-negative production levels and reduced the risk of introducing artificial anomalies in subsequent analyses.

To better assess whether simple aggregate statistics were sufficient to identify faulty strings, additional exploratory analysis was conducted. Pairwise Pearson correlations between strings were generally high, with an average correlation of 0.987, a median of 0.991, and a maximum of 0.999. At the same time, the minimum observed correlation was 0.780, indicating that a small number of strings deviated from the common behavior of the plant. This result suggests that most strings behave in a highly similar manner over time. In contrast, a small number exhibit persistent differences that may be linked to faults or long-term performance degradation.

A similar conclusion follows from the distribution of total generated power. Although the first quartile was 43,123 kW and the median was 44,148 kW, only three strings generated less than 40,000 kW over the full study period. This shows that total production alone is not sufficient for identifying all problematic systems, because many faults may affect only specific hours or seasons rather than the overall cumulative output. For this reason, the later methodology relies on a broader set of extracted features that capture temporal structure, seasonality, and changes in behavior across the whole time series.

3 Methodology

A summary of the workflow is presented in Figure 3. The work is mainly comprised of two parts. The first section addresses outlier detection, which aims to identify abnormally behaving solar panel strings. The second one focuses on constructing fault profiles for the most anomalous strings, providing a more detailed analysis of each system's performance and insights into previous findings.

In order to apply the chosen unsupervised anomaly detection methods to the solar panel string data, it is necessary to extract features that effectively capture the global structure and underlying patterns of the time series (see Table 2).

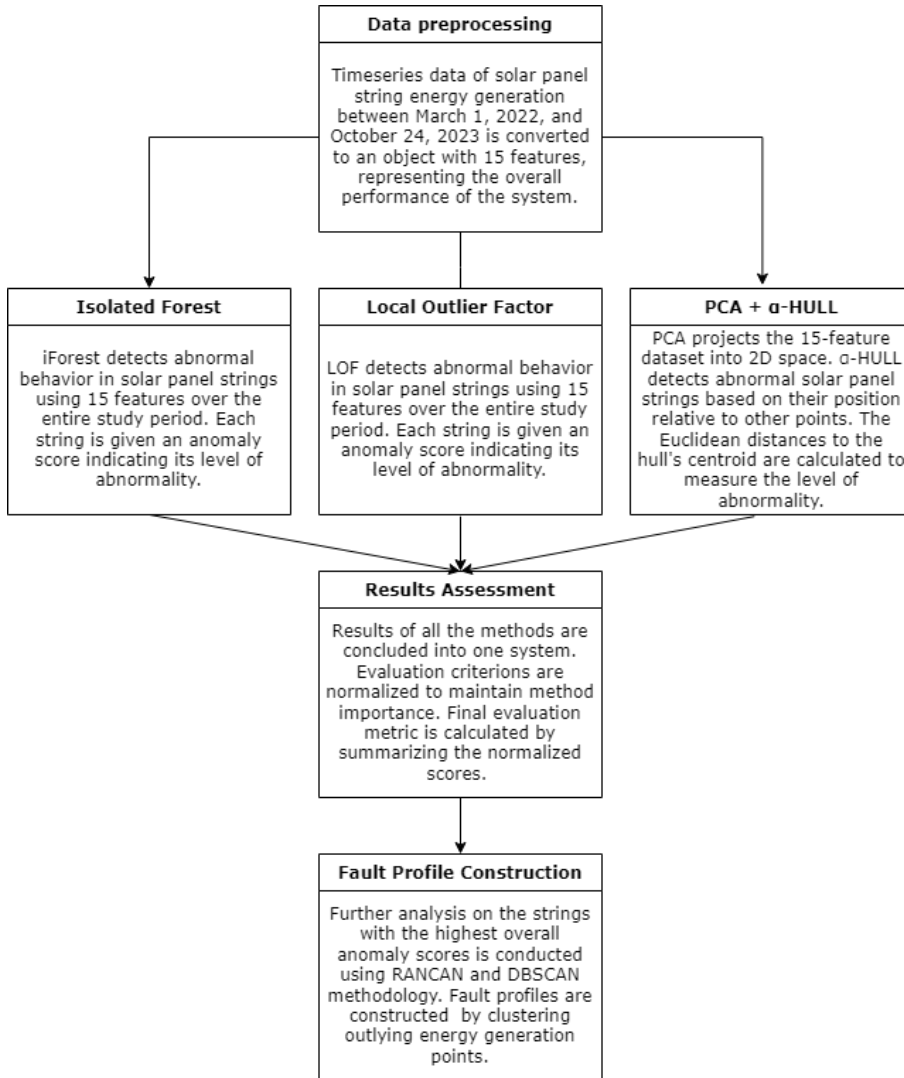


Fig. 3: Summary workflow chart.

To compute features like ACF1, trend strength, linearity, curvature, seasonal strength, entropy, peaks and troughs the R package *tsfeatures* [12] was used. These features are calculated using the STL (Seasonal and Trend decomposition using Loess) [13] method. The decomposition contains a trend, up to M seasonal components and a remainder component:

$$x_t = f_t + s_{1,t} + \dots s_{M,t} + e_t \quad (2)$$

where

Table 2: Features extracted from solar panel string time series for unsupervised anomaly detection.

Variable	Description
Mean	Mean value of the time series.
Variance	Variance of the time series.
ACF1	First-order autocorrelation.
Trend _{Strength}	Indicates how strong the trend is.
Linearity	Degree to which the data follows a straight line.
Curvature	Degree to which the data follows a curved pattern.
Seasonality _{Strength}	Indicates how strong the seasonal pattern is.
Entropy	Measures the "forecastability" of the time series.
Peak	Strength of peaks; computed from size and location of peaks in the seasonal component.
Trough	Strength of troughs; computed from size and location of troughs in the seasonal component.
Crossings	Number of times the time series crosses the mean line.

- f_t is the smoothed trend component,
- $s_{i,t}$ is the i -th seasonal component,
- e_t is a remainder component.

The trend and seasonality strengths are calculated by the following formulas:

$$Trend_{Strength} = 1 - \frac{Var(e_t)}{Var(f_t + e_t)} \quad (3)$$

$$Seasonality_{Strength} = 1 - \frac{Var(e_t)}{Var(s_{i,t} + e_t)} \quad (4)$$

Additional features included are hourly energy generation. For each object, 4 features are computed representing the total energy generated per grouped hours — from 08:00 to 11:00, 12:00-15:00, 16:00-19:00 and 20:00-23:00 between March 1, 2022, and October 24, 2023. Hours 24:00-07:00 are not included because none of the strings generated any energy during this period. In fact, this may help identify shadowing, as if some system is shaded during some specific hours, its power output will be significantly lower than others during these time-frames. Therefore, with the addition of hourly energy generation, each time series will contain a total of 15 features. Given the diverse scales of features, min-max normalization will be used.

3.1 Outlier detection

After data preprocessing, a dataset with 143 objects and 16 features has been obtained, where each row represents a solar panel string. The computed features provide insights

into the performance of each individual string throughout the entire examined time frame. Thus, now various strategies of high dimensional data anomaly detection can be applied to identify abnormally behaving systems.

3.1.1 PCA and α -HULL

Anomaly detection in high dimensional data can be effectively conducted by integrating PCA [8] with the α -hull method [11]. To begin, PCA is applied to the dataset to reduce its dimensionality, allowing the identification of the number of principal components that adequately capture the essential variance, tailored to specific use cases. Subsequently the α -hull method can be performed on the reduced dataset. The α -hull method is a generalization of convex hull method. Convex hull is the smallest convex set that contains a set of points. It is similar to stretching a rubber band around the outermost points, the area the rubber band encloses is the convex hull. The α -hull method extends this concept by introducing a parameter α , which represents the radius of a generalized disk. The α -hull method uses these disks to determine the boundary of the point set. The size of α crucially affects the resulting shape:

- When α is very small, the α -hull can potentially enclose smaller clusters of points or even individual points, showing a more detailed boundary.
- When α is very large, the α -hull approaches the convex hull, as the generalized disks become large enough to enclose all points, and the shape simplifies to that of a convex hull.

Although this method does not directly yield any anomaly score, a naive scoring system based on the Euclidean distance from the centroid C of the cluster enclosed by the hull was implemented. The centroid coordinates of cluster A were calculated using the points that are in that cluster:

$$C = (C_x, C_y) \quad (5)$$

where

$$C_x = \mathbb{E}\left(\sum_{o \in A} o_x\right) \quad (6)$$

and

$$C_y = \mathbb{E}\left(\sum_{o \in A} o_y\right) \quad (7)$$

3.1.2 Isolation forest

Isolation Forest [9] is a tree-based approach for outlier detection. It works by isolating anomalous instances from normal ones through the construction of binary tree data structures across the features of the dataset. The iForest algorithm operates through a series of steps:

1. **Random Feature Selection.** The algorithm begins by randomly selecting a feature from the dataset.
2. **Random Split Value Selection.** Once a feature is chosen, Isolation Forest randomly selects a split value between the minimum and maximum values of that feature.
3. **Binary Tree Construction.** Using the randomly selected feature and split value, iForest constructs a binary tree data structure.

These steps are iterated for the remaining features until binary trees are constructed for all features. Outliers are identified based on their average path length in each tree. Anomaly score equation:

$$s(z, n) = 2^{-\frac{\mathbb{E}(h(z))}{c(n)}}, \quad (8)$$

where:

- z is data point,
- n is total number of instances in the dataset,
- $h(z)$ is path length of a point z , measured by the number of edges z traverses from the root node until a leaf node is reached,
- $\mathbb{E}(h(z))$ is average path length of point z over all binary trees,
- $c(n) = 2H(n-1) - \frac{2(n-1)}{n}$ is path length normalization constant; $H(i)$ is the i -th harmonic number.

3.1.3 Local outlier factor

Unsupervised anomaly detection for high-dimensional data can also be done using the LOF method [10]. LOF is based on the local density deviation of a given data point with respect to its neighbors. The main advantage of this method compared to other outlier detection methods is that it can identify local outliers. The method starts by finding the k nearest neighbors $N_k(z)$ of data point z . If a tie between some points occurs, more than k points may be used. Then, for each point, the average local reachability density (*LRD*) is computed:

$$LRD_k(z) = \frac{\sum_{o \in N_k(z)} dist(z, o)}{|N_k(z)|}, \quad (9)$$

where

- $dist(z, o)$ – Distance measure between points z and o . In our case, the Euclidean metric is used:

$$d(z, o) = \sqrt{(z_1 - o_1)^2 + (z_2 - o_2)^2 + \dots + (z_n - o_n)^2}. \quad (10)$$

- $|N_k(z)|$ – Number of elements in set $N_k(z)$.

The final step is to calculate the Local Outlier Factor (*LOF*), which is the average local reachability density of the neighbors divided by the object's own local reachability density:

$$LOF_k(z) = \frac{\sum_{o \in N_k(z)} LRD_k(o)}{|N_k(z)| \cdot LRD_k(z)}. \quad (11)$$

3.2 Fault profile construction

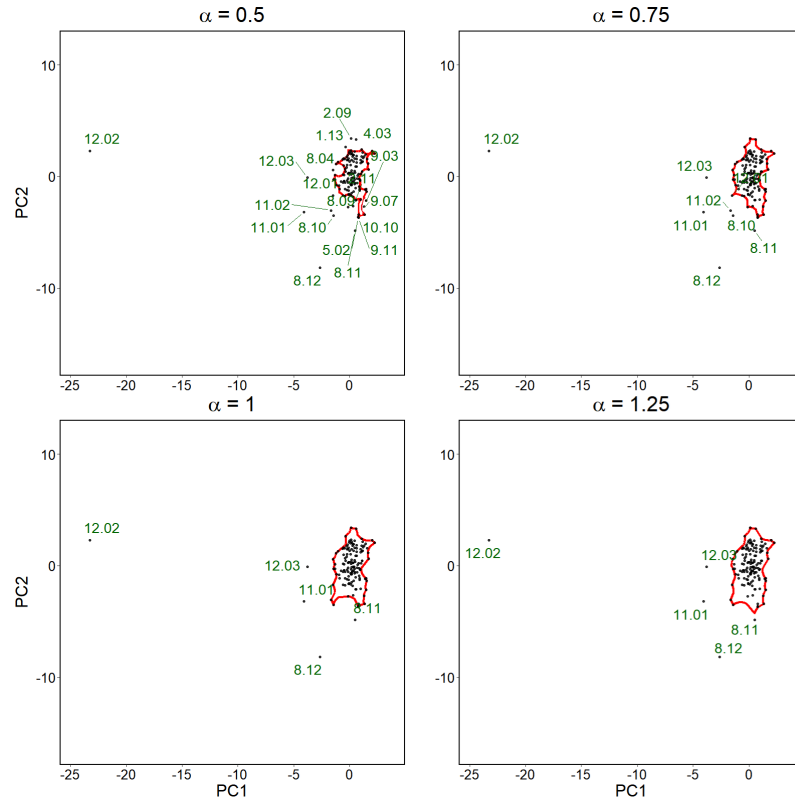
Before proceeding to construct the fault profile, the results provided by different models must first be evaluated. To achieve this, a system that calculates a combined anomaly score based on the outputs of various methods was introduced. Specifically, the anomaly scores from iForest, LOF, and the Euclidean distance from α -HULL were used. To ensure consistent influence among all methods, min-max normalization was applied to these scores. Then the normalized scores were aggregated to get a final evaluation metric. The five systems exhibiting the highest anomaly scores will be thoroughly investigated by using the RANSAC + DBSCAN methodology:

1. **Outlier Detection:** The first step identifies outliers in power output values. It is assumed that nearby solar systems with identical specifications exhibit a linear relationship under normal conditions. For each solar panel string, an optimal energy reference is built using its neighboring strings from the same inverter. The benchmark is the maximum energy output of these neighbors at each timestamp. Linear regression and RANSAC then estimate this linear relationship, flagging deviations as outliers.
2. **Pattern Visualization:** Outliers are plotted with hour on the x-axis and date on the y-axis, excluding nighttime. Clusters of outliers at specific times or dates reveal patterns of abnormal performance.
3. **Outlier Clustering:** DBSCAN groups dense clusters of outliers while filtering out noise. This produces a clear and consistent fault profile for each string.

4 Experimental results

The implementation results of α -HULL with varying α values are depicted in Figure 4. For each α value, an α -convex hull was constructed, with the generalized disk radius set equal to α . As shown in the graphs, selecting a lower α value results in more outlying points. When α was set to 1 and 1.25, two strings from the 8th inverter, one string from 11th inverter and two strings from 12th inverter were found to be outliers. When α value was reduced to 0.75 then 3 additional outliers were detected, one of each from the 8th, 11th and 12th inverters. When α value was further lowered by 0.25, then 11 new outliers were caught. Among those 11 strings 8 were from previously not detected inverters (1st, 2nd, 4th, 5th and 9th). The maximum number of anomalous strings detected is 19. Notably, five outliers - 8.11, 8.12, 11.01, 12.02, and 12.03 — are consistently present across all variations. Here, the notation $x.y$ indicates inverter x and string y .

For every outlier identified Euclidean distance (10) between that point and the computed centroid C was calculated. Results when using $\alpha = 0.5$ are shown in Table 3. In

Fig. 4: α -HULL results.

scenarios with different α values, the distances differ due to discrepancies in both the outliers and the computed centroids. Hence, primarily, focus was put on the outcome when $\alpha = 0.5$, as it offers a broader selection of points.

In the case of the iForest method, a lot more information about the solar panel string is used. The algorithm uses all 16 features to detect abnormally behaving strings. The decision was made to check the results of outliers using two different *contamination* parameter values: 0.05 and 0.10. This implies that encountering a total of either 5 % or 10 % outliers within the entirety of the dataset was expected.

For the LOF method, the *contamination* parameter must also be specified. Values of 0.05 and 0.10 were chosen, expecting 5 % or 10 % outliers respectively. The neighborhood size for density computation is fixed at 10, determined after experimenting with different values. Results are shown in Table 5.

For the comparison iForest with a *contamination* parameter set to 0.10, the LOF method with a *contamination* value of 0.10, and the PCA + α -HULL method with an α value set to 0.5 were used.

Table 6 presents the shared outliers identified among results obtained from different methods. Notably, four outlier strings (12.02, 8.11, 12.01, and 8.10) are consistently

Table 3: α -Hull outlier distances, when $\alpha = 0.5$.

String	Distance
12.02	23.63597
8.12	8.871502
11.01	5.533009
8.11	5.072597
8.10	4.097642
12.03	4.052169
9.11	3.939188
5.02	3.908802
10.10	3.778132
11.02	3.774506
9.07	3.662785
2.09	3.183685
4.03	3.13508
9.03	3.108478
8.09	2.979747
2.11	2.883931
12.01	2.606329
1.13	2.509189
8.04	1.778152

Table 4: iForest results.

Contamination = 0.05		Contamination = 0.1	
String	Anomaly Score	String	Anomaly Score
12.01	0.248189	12.01	0.267126
8.11	0.138375	8.11	0.157312
12.02	0.064189	12.02	0.083125
10.13	0.058485	10.13	0.077421
8.10	0.024483	8.10	0.043420
12.04	0.014717	12.04	0.033654
8.09	0.005382	8.09	0.024318
8.08	0.000806	8.08	0.019742
		6.09	0.011683
		1.02	0.005514
		12.08	0.003381
		7.10	0.002234
		1.08	0.002157
		11.12	0.000340
		8.07	0.000027

flagged across all three methods. Furthermore, there were seven mutual outliers between iForest and LOF, five between iForest and α -HULL, and six between LOF and α -HULL method pairs.

Different anomaly scores were aggregated across all the methods to provide an overall assessment of the system's performance relative to others. Min-max normalization

Table 5: LOF results.

Contamination = 0.05		Contamination = 0.10	
String	Anomaly Score	String	Anomaly Score
12.01	2.955928	12.01	2.955928
12.02	2.086247	12.02	2.086247
8.11	2.049132	8.11	2.049132
10.13	1.740679	10.13	1.740679
1.02	1.52342	1.02	1.52342
12.10	1.453929	12.10	1.453929
		8.10	1.431485
		11.01	1.393056
		6.01	1.390974
		7.03	1.342076
		6.08	1.322753
		1.08	1.316031
		11.02	1.307126

Table 6: Common outliers.

Methods	Common outliers
iForest, LOF, α -HULL	12.02 ; 8.11 ; 12.01 ; 8.10
iForest, LOF	12.02 ; 8.11 ; 12.01 ; 8.10 ; 1.02 ; 1.08 ; 10.13
iForest, α -HULL	12.02 ; 8.11 ; 12.01 ; 8.10 ; 8.09
LOF, α -HULL	12.02 ; 8.11 ; 12.01 ; 8.10 ; 11.01 ; 11.02

is necessary to maintain the same level of influence between all methods. Furthermore, the total scores by summing the normalized results from all three methods were calculated. The 5 strings with the highest combined anomaly scores are shown in Table 7. In instances where a string was not identified as an outlier by a particular method, it was denoted with ”-”.

Table 7: Combined anomaly scores.

String	iForest	LOF	α -HULL	Total
12.01	1	1	0.037889	2.037889
12.02	0.311113	0.472538	1	1.783651
8.11	0.588864	0.450027	0.150722	1.189613
10.13	0.289758	0.26295	-	0.552708
8.10	0.16246	0.075424	0.106117	0.344001

The fault profile construction process can be illustrated using a specific example, focusing on the 12.01 solar panel string (see Figure 5), which has the highest anomaly

score among the systems analysed. The reference data source is constructed using other systems of the 12th inverter. In this visualization, inliers are denoted in green, while outliers are marked in red. The RANSAC method detects points that deviate from the found linear relationship. In the context of energy generation, points below indicate cases in which the examined system generated less than the reference source. The energy generation points are then visualized through a scatter plot, with the hour on the x-axis and the date on the y-axis. To ensure clarity, all energy generation points at night are removed, since neither the reference source nor the string of interest generates energy during these hours. Certain deviating energy-generation points consistently occur along the X-axis, indicating potential performance issues for the string on that particular day. However, the presence of points in a dense region along the X-axis and across the Y-axis suggests a frequent malfunctioning pattern. Hence, the denser regions formed represent a profile of consistent malfunction throughout specific hours. Some outliers are far from any dense cluster; they would be considered noise in this case. Notably, there are no outliers during the winter months. This could be explained by the fact that solar irradiance is significantly lower during winter. Later, DBSCAN was applied to cluster the points. It can be seen that six distinct clusters have been constructed among the observed points. Other energy-generation points, marked in grey, are treated as noise and are not further analysed. Based on these findings, the string exhibits recurring issues during two distinct time periods: from 10:00 to 12:00 and from 15:00 to 20:00. The consistent occurrence of these issues during specific hours suggests that the string is affected by shadowing. During periods of high solar irradiance, some objects cast shadows, obstructing the panels from sunlight and thereby reducing energy output.

It is also important to note that no outliers were observed during the winter months for the 12.01 string. A likely explanation is that solar irradiance is substantially lower in winter, and the overall plant generation is reduced during this period. As a result, differences between strings become less visible and chronic faults are harder to detect. This observation is consistent with the earlier seasonal analysis and further supports the decision to study fault profiles primarily during higher-irradiance periods.

Applying the same methodology to other solar panel strings identified as anomalies produces the results illustrated in Figure 6. Similar fault patterns emerge across distinct systems, with notable discrepancies in energy generation between 7:00-10:00 and 16:00-20:00 in most systems. The clearest fault pattern is observed in the 12.02 solar panel string, where DBSCAN identified two dense clusters, indicating significantly reduced energy production during hours 14:00-20:00. Given the consistency of these outliers during high-irradiance timeframes and their occurrence only during specific hours, this suggests a shadowing problem. This interpretation is supported by feedback from plant operators, suggesting that the detected patterns correspond to operational issues.

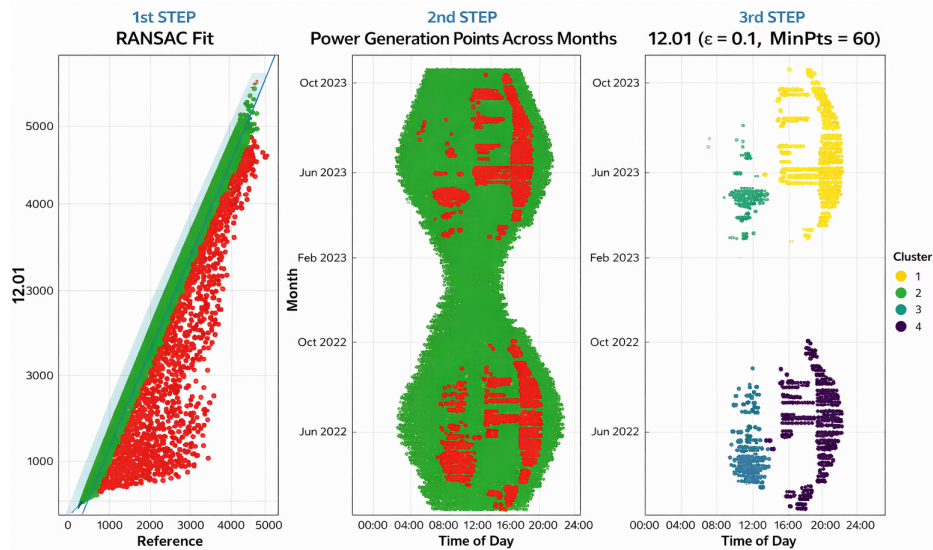


Fig. 5: Fault profile construction for 12.01 string

For comparison, the same profiling procedure was applied to a string that had not previously been identified as anomalous. String 4.03 was selected as a non-faulty example (see Figure 7). In this case, DBSCAN did not form any dense clusters from the detected outlier points. Most deviations were concentrated within a single short period and did not recur in a structured manner over time. This indicates the absence of a persistent malfunction pattern and suggests that the observed points are more likely to be noise than evidence of a chronic fault.

5 Conclusions

Through the application of various anomaly detection methods — PCA + α -HULL, Isolation Forest (iForest), and Local Outlier Factor (LOF) — multiple faulty solar panel strings in the solar power plant were successfully identified. There were four systems detected as anomalies in all three methods: 12.02, 8.11, 12.01, and 8.10.

The combination of these methods ensured a comprehensive assessment, highlighting strings with significant deviations from normal operation. The combined anomaly score, derived from the normalized results of the three methods, provided a metric for identifying the most anomalous strings. This score helped prioritize strings for further analysis.

Using RANSAC and DBSCAN methodology, detailed fault profiles were constructed for the identified faulty strings with the highest anomaly scores. These profiles revealed recurring patterns of reduced energy generation, occurring consistently during specific

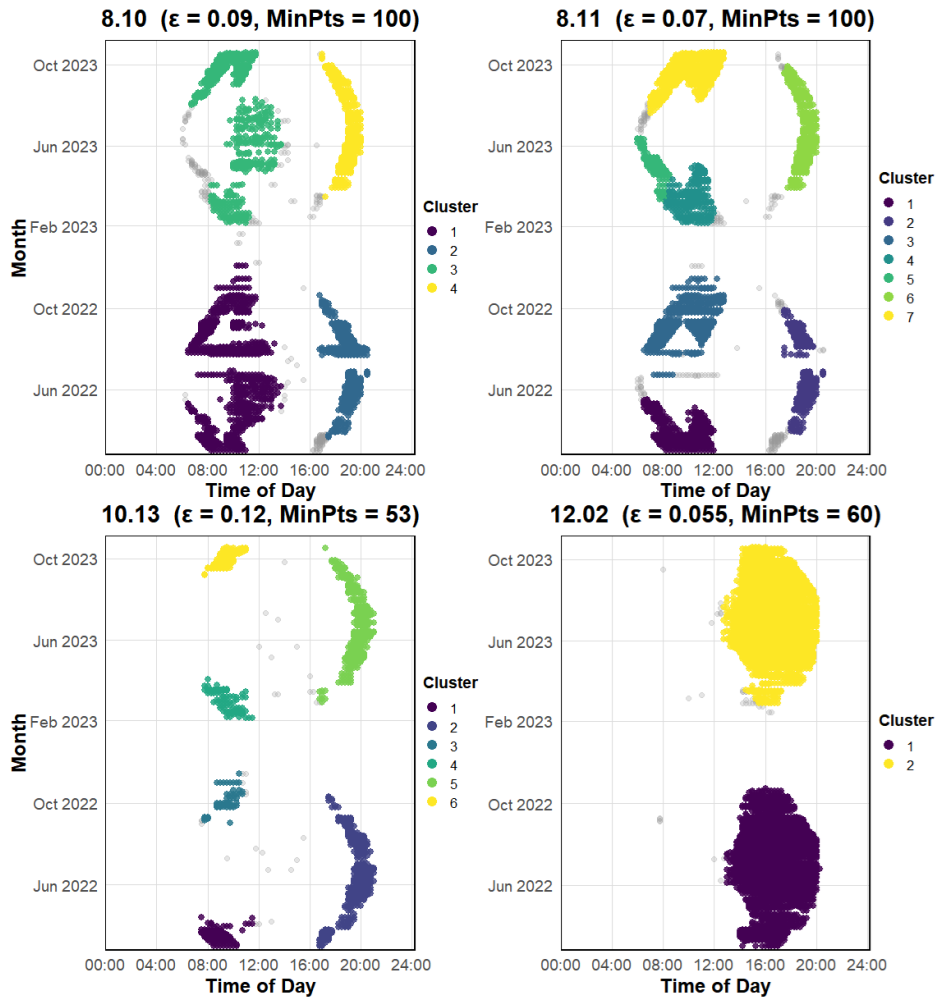


Fig. 6: Fault profiles of other strings detected as outliers

hours, indicating potential issues such as shadowing. The clustering of outliers provided a clear visualization of the malfunction periods, further validating the results from the anomaly detection methods.

This methodology also enables detailed performance analysis over any timeframe, providing clear insights into the behavior of each string. The visualizations make it easy to compare strings, and support data-driven decisions for maintenance or optimization, making it an effective tool for monitoring and managing solar power plants.

Despite the promising results, several limitations should be acknowledged.

The study relies on unsupervised methods without ground truth labels, making it difficult to evaluate detection accuracy directly. Therefore, anomalies are identified based

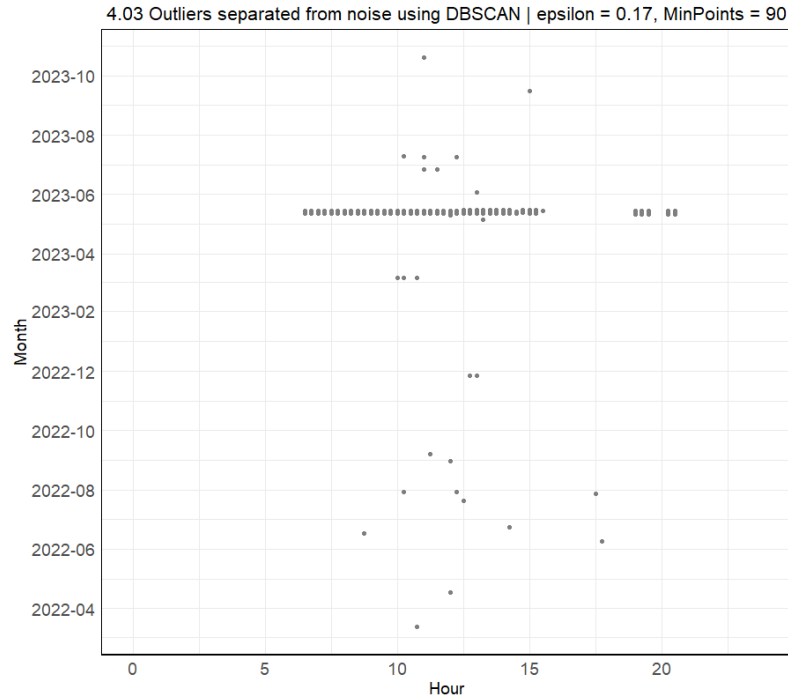


Fig. 7: 4.03 string DBSCAN results

on their consistent detection across multiple methods rather than verified fault labels. Despite this limitation, the findings are consistent with expert knowledge from plant operations, which confirmed shadowing in most of the detected strings.

The performance of the methods is also sensitive to parameter selection. Parameters such as the α value in α -HULL, contamination in Isolation Forest, and neighborhood size in LOF can influence the results. These parameters can be refined using expert knowledge or practical insights, thereby improving the reliability of anomaly detection.

In addition, the selected feature set, while informative, may not capture all aspects of abnormal behavior. External factors such as weather conditions or solar irradiance were not included and could improve detection performance.

Finally, the dataset is limited to a single solar power plant with identical component specifications, which may limit the generalizability of the results to other plants or systems with different configurations.

References

1. Perdue, M., & Gottschalg, R. (2015). Energy yields of small grid connected photovoltaic system: Effects of component reliability and maintenance. *IET Renewable Power Generation*, 9(5), 432–

437. <https://doi.org/10.1049/iet-rpg.2014.0048>
2. Gupta, Y., Yadav, N. P., Singh, A., Kumar, A., & Vishwakarma, S. (2022). Faults occur in solar PV power generation system. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 11(3), 1368–1375.
 3. Bernadette, D., Twizerimana, M., Bakundukize, A., Pierre, B. J., & Theoneste, N. (2021). Analysis of shading effects in solar PV system. *International Journal of Sustainable and Green Energy*, 10(2), 47–62. <https://doi.org/10.11648/j.ijrse.20211002.12>
 4. Tsafarakis, O., Sinapis, K., & Van Sark, W. G. (2018). PV system performance evaluation by clustering production data to normal and non-normal operation. *Energies*, 11(4), Article 977. <https://doi.org/10.3390/en11040977>
 5. Tsafarakis, O., & van Sark, W. G. (2023). A density-based time-series data analysis methodology for shadow detection in rooftop photovoltaic systems. *Progress in Photovoltaics: Research and Applications*, 31(5), 506–523. <https://doi.org/10.1002/pip.3661>
 6. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (Vol. 96, No. 34, pp. 226–231). AAAI Press.
 7. Hyndman, R. J., Wang, E., & Laptev, N. (2015, November). Large-scale unusual time series detection. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)* (pp. 1616–1619). IEEE. <https://doi.org/10.1109/ICDMW.2015.104>
 8. Jaadi, Z. (2021, May 20). *A step-by-step explanation of principal component analysis (PCA)*. Built In. <https://builtin.com/data-science/step-step-explanation-principal-component-analysis>
 9. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422). IEEE. <https://doi.org/10.1109/ICDM.2008.17>
 10. Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (pp. 93–104). Association for Computing Machinery. <https://doi.org/10.1145/342009.335388>
 11. Pateiro-López, B., & Rodríguez-Casal, A. (2010). Generalizing the convex hull of a sample: The R package alphahull. *Journal of Statistical Software*, 34(5), 1–28. <https://doi.org/10.18637/jss.v034.i05>
 12. Hyndman, R., Kang, Y., Montero-Manso, P., O’Hara-Wild, M., Talagala, T., Wang, E., & Yang, Y. (2019). *tsfeatures: Time series feature extraction* [R package]. CRAN. <https://CRAN.R-project.org/package=tsfeatures>
 13. Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1), 3–73.

About authors

Dominykas Vilčinskis is Data Science student at Vilnius University, currently pursuing a Master’s degree after completing his Bachelor’s studies in Data Science in 2025. During his studies, he has been actively involved in academic events and research activities focused on data analysis and machine learning. He has presented his work at several conferences, including a poster presentation titled “Fault Detection in Solar Power Plants Using Energy Production Data” at the 15th Conference on Data Analysis Methods for Software Systems in 2024. He also co-authored a publication on “Wage Prediction for Salaried Employees Using Machine Learning Methods,” presented at the national conference Lietuvos magistrantų informatikos ir IT tyrimai in 2025. His main research interests include applied machine learning and data-driven decision making.

Lukas Voveris is a Master's student in Data Science at Vilnius University, Faculty of Mathematics and Informatics. He earned his Bachelor's degree in Data Science at Vilnius University in 2025. His research focuses on solar PV monitoring and energy analytics. He co-authored and presented the poster "Fault Detection in Solar Power Plants Using Energy Production Data" at the 15th Conference on Data Analysis Methods for Software Systems in 2024. He also authored "Implementation of Machine Learning and Statistical Techniques in Solar Energy Generation Monitoring Systems". This paper was presented at the national conference Lietuvos magistrantų informatikos ir IT tyrimai in 2025 and at the AI2SEP project conference in Varaždin, Croatia. His research interests include applied machine learning, time series forecasting, solar irradiation modeling, and decision support for energy systems.

Jolita Bernatavičienė graduated from Vilnius Pedagogical University in 2004 and received a master's degree in informatics. In 2008, she received a doctoral degree in computer science (PhD) from the Institute of Mathematics and Informatics jointly with Vilnius Gediminas Technical University. She is a senior researcher at the Cognitive Computing Group of Vilnius University's Institute of Data Science and Digital Technologies. Her research interests include databases, data mining, neural networks, image analysis, visualisation, decision support systems and internet technologies, and high-performance computing. She supervises 3 PhD students and has written more than 60 articles, 18 of which are in CA WoS database.

Machine Learning for Air Quality and CO_2 Emissions: The Role of Data Understanding

Ivan Maslov,
Agnieszka Głowacka,
Bartosz Dziewit, and
Paulina Trybek

Abstract

The emergence of machine learning (ML) has enabled sophisticated approaches to environmental prediction, yet the datasets underlying these models remain underexamined. This study investigates the role of data quality and structure in ML-based environmental applications, focusing on greenhouse gas (GHG) emissions and particulate matter concentrations. For Poland and Slovakia, a LightGBM model was trained to predict CO_2 emissions across major economic sectors: Residential, Power, Transport, Industry, and Aviation. Predictive performance was highest in sectors with regular seasonal patterns, while low-variability sectors such as Domestic Aviation posed greater challenges. For particulate matter, meteorological and time-related features were used to forecast $PM_{2.5}$ and PM_{10} during the heating season. Models captured general temporal patterns and seasonal peaks, though extreme events were partially underestimated. The findings demonstrate that predictive accuracy is strongly influenced by dataset quality, resolution, and structure, alongside emission regularity and environmental conditions. This chapter underscores the importance of careful dataset design and preprocessing in ML applications for environmental monitoring, offering practical guidance for improving the reliability of emission and air quality forecasting.

Keywords: machine learning; CO_2 emissions; dataset transparency; $PM_{2.5}$ particles; air quality forecasting

Ivan Maslov and Agnieszka Głowacka
University of Silesia, Faculty of Science and Technology, 40-007 Katowice, Poland
e-mail: ivan.maslov@us.edu.pl, agnieszka.glowacka@us.edu.pl
Bartosz Dziewit and Paulina Trybek
University of Silesia, Institute of Physics, 40-007 Katowice, Poland
e-mail: bartosz.dziewit@us.edu.pl, paulina.trybek@us.edu.pl

1 Introduction

Air pollution constitutes one of the most significant environmental health risks worldwide. According to the World Health Organization, exposure to polluted air contributes to millions of premature deaths annually, primarily due to fine particulate matter ($PM_{2.5}$) [1]. Previous global assessments have further demonstrated that ambient $PM_{2.5}$ accounts for a substantial mortality burden and that significant reductions in premature deaths could be achieved through region-specific air quality improvements aligned with WHO guidelines [2].

At the same time, atmospheric carbon dioxide (CO_2), the dominant anthropogenic greenhouse gas, is the principal driver of long-term climate change. More broadly, greenhouse gas (GHG) emissions are widely recognized as the primary force behind ongoing global warming. While particulate matter ($PM_{2.5}$, PM_{10}), carbon monoxide (CO), and carbon dioxide (CO_2) differ in their physio-chemical properties and atmospheric lifetimes, they frequently originate from common combustion-related sources such as transport, industry, and residential heating. This overlap in emission sources highlights the close interconnection between air quality and climate policy: measures targeting fossil fuel combustion can simultaneously reduce harmful air pollutants and greenhouse gas emissions.

In 2020, the European Union adopted a plan to reduce net greenhouse gas emissions by at least 55% by 2030 compared to 1990 levels, alongside the longer-term objective of achieving climate neutrality by 2050 [3]. Such policy frameworks not only aim to mitigate climate change but also have the potential to deliver substantial public health co-benefits through improved air quality.

To address these challenges, data-driven approaches, including machine learning and artificial intelligence methods, are increasingly applied to model and predict atmospheric concentrations of pollutants such as $PM_{2.5}$ or CO_2 [4, 5, 6]. By leveraging large environmental datasets and complex nonlinear relationships among meteorological, emission, and spatial variables, these methods can improve the accuracy of air quality forecasting and support evidence-based environmental management and policy-making. However, the reliability of such predictive models strongly depends on the availability of high-quality and representative data that adequately capture the key factors influencing atmospheric concentrations of pollutants, including meteorological conditions, emission sources, and spatial-temporal variability.

Despite certain similarities in their emission sources, predicting atmospheric concentrations of $PM_{2.5}$ and CO_2 involves different methodological challenges. Particulate matter concentrations are typically characterized by strong short-term variability and pronounced spatial heterogeneity driven by local emission sources, meteorological conditions, and atmospheric processes such as dispersion and secondary aerosol formation. In contrast, atmospheric CO_2 concentrations tend to exhibit smoother spatial patterns and stronger long-term temporal trends associated with global-scale carbon cycles and cumulative greenhouse gas emissions. As a result, predictive models for $PM_{2.5}$ often require high-resolution spatiotemporal data capturing local environmental conditions, whereas CO_2 modeling more frequently emphasizes large-scale temporal dynamics and broader emission pattern.

The importance of CO_2 in climate policy can be illustrated by recent emission statistics. According to the most recent Polish National Greenhouse Inventory report (NIR) [7], prepared under the United Nations Framework Convention on Climate Change (UNFCCC), carbon dioxide accounts for 81.4% of total greenhouse gas emissions in Poland. A similar dominance of CO_2 is observed at the European level, where it represents roughly 80% of total greenhouse gas emissions [8]. Therefore, reducing CO_2 emissions plays a central role in achieving climate neutrality.

In order to keep such emissions under control, policy makers, researchers and analysts require reliable modeling approaches and prediction algorithms, which are among the key topics studied in modern computer science. In recent years, a plethora of articles has been published applying various machine learning (ML) models and techniques for the prediction of CO_2 emissions [9, 10, 11, 12, 13]. This research area is still under active development, and no single authoritative solution has yet been adopted at an intergovernmental level.

Related work explores a wide range of machine learning algorithms across different sectoral contexts. However, the data used to train these models is often accepted without sufficient scrutiny. In most cases, datasets originate from official publications or open-source repositories, are lightly processed to handle missing values, and are then passed directly to models without deeper analysis.

The characteristics and limitations of the source data—such as spatial and temporal resolution, aggregation methods, data composition, uncertainty evaluation, and even the fact that much of the data represents estimates rather than direct measurements—rarely receive adequate attention. As a result, datasets are frequently taken for granted, despite their significant influence on model outcomes.

Unlike CO_2 emissions, which are largely estimated based on activity data and emission factors, concentrations of particulate matter such as $PM_{2.5}$ and PM_{10} are typically determined through direct physical measurements. This means that these data reflect the actual state of the atmosphere at a specific place and time.

Pollutant concentration is expressed as the amount of particles per cubic meter of air. PM_{10} includes particles with a diameter of up to 10 micrometers, while $PM_{2.5}$ includes particles up to 2.5 micrometers in diameter. This distinction is important both from a health and a measurement perspective, as finer particles behave differently in the atmosphere. Various technologies are used to measure $PM_{2.5}$ and PM_{10} concentrations. The reference method remains gravimetry, which involves drawing air through a filter and determining particle concentrations based on the increase in the filter's mass. While highly accurate, this approach does not provide real-time results. In practice, automatic analyzers, such as the Beta Attenuation Monitor (BAM) and the Tapered Element Oscillating Microbalance (TEOM), are more commonly employed, enabling hourly measurements and continuous monitoring of particulate levels. Increasingly important are also low-cost sensors based on light scattering, which allow for denser monitoring networks but are characterized by greater uncertainty [14]. Technological advancements have increased the availability of real-time data and the density of observations. This is particularly important for epidemiological analyses and predictive modeling. At the same time, a larger number of devices implies greater variability in measurement quality. Under conditions of high humidity, especially during fog, microdroplets of water may be recorded as additional particulate matter, leading to overestimated results. Dif-

ferent types of instruments may also respond differently to the same atmospheric conditions, resulting in systematic differences in measurements. Additionally, sensors—especially lower-cost models—may lose accuracy over time if not regularly calibrated [15]. An important feature of PM data is their high temporal resolution. Dust concentration measurements are typically recorded at hourly intervals, and sometimes even more frequently, meaning that a single station generates thousands of observations per year. This enables the analysis of diurnal cycles, smog episodes, weekend effects, and seasonal patterns related to the heating period. However, a high number of observations does not automatically translate into greater representativeness. Measurements refer to a specific location and reflect local emission and dispersion conditions. PM data therefore have a point-based and spatially limited character, as they represent a record of the instantaneous atmospheric state at a given location rather than an averaged measure at the regional or national scale [16].

In this study, we examine the distinct characteristics and challenges of environmental datasets for both CO_2 and particulate matter ($PM_{2.5}$ and PM_{10}) in the context of machine learning applications. Unlike particulate matter, which is measured directly at specific locations and reflects short-term local atmospheric conditions, CO_2 emissions are largely estimated based on activity data and emission factors, making them subject to additional sources of uncertainty. These differences imply that predictive modeling for each pollutant type requires tailored approaches: high-resolution spatiotemporal data for particulate matter and broader temporal and sectoral information for CO_2 . Accordingly, this work tests predictive models for both CO_2 and PM concentrations, highlighting how data set quality, representativeness, and the nature of the underlying measurements influence the reliability and interpretability of machine learning forecasts.

2 Analysis of carbon emission data

There are typically two types of CO_2 emission data used in practice: direct measurements and emission estimates. The former is relatively rare and is usually available either from short-term, locally focused research projects or from controlled test datasets, such as those provided by the automotive industry through the Worldwide Harmonized Light Vehicles Test Procedure (WLTP) framework [17].

Emission estimates, on the other hand, are far more common. Large-scale CO_2 emission databases typically rely on estimations derived from statistical information about human activities, such as energy production, transportation, and industrial processes.

The main reference for CO_2 emission estimation is the 2006 IPCC Guidelines for National Greenhouse Gas Inventories [18]. The Intergovernmental Panel on Climate Change (IPCC) is a United Nations body responsible for assessing scientific knowledge related to climate change. It was established in 1988 by the United Nations Environment Programme (UNEP) and the World Meteorological Organization (WMO) to provide a consistent and transparent framework for the evaluation and reporting of greenhouse gas (GHG) emissions.

The 2006 IPCC Guidelines establish a concrete methodology for estimating greenhouse gas inventories, that is, GHG datasets developed with defined quality assurance

procedures and uncertainty evaluation. The IPCC framework provides a division of emissions by major human activity sectors, namely energy, industrial processes, agriculture, forestry and other land use, and waste, along with further subdivisions within each sector. In addition, the Guidelines offer detailed guidance on how emission values should be calculated for different activities and data availability levels.

The IPCC provides a basic approach for emission estimation (1), in which emissions are calculated as the product of activity data (AD) and an emission factor (EF). Activity data represent the magnitude of a human activity—such as the amount of fuel consumed in a given process—while the emission factor defines the amount of CO_2 emitted or removed per unit of activity.

$$Emissions = AD * EF \quad (1)$$

The form of activity data varies across economic sectors. For example, it may correspond to the quantity of coal burned in industrial facilities, or to a more elaborate representation of the aviation sector that accounts for the number of flights, travel distances, aircraft types, and other relevant characteristics. Nevertheless, the underlying estimation principle remains the same, which is why activity data derived using different approaches can be combined in total emission calculations.

Regarding emission factors, the IPCC provides a database of standardized values that can be used as defaults. However, it is considered good practice to apply country-specific or locally derived emission factors whenever possible, as they better reflect local conditions and technologies. For instance, if coal used within a given country has distinct physical or chemical characteristics, locally measured emission factors can provide more accurate estimates of the resulting CO_2 emissions.

The IPCC also introduces three tiers of analysis that differ in terms of data granularity and methodological complexity. Tier 1 relies on national-level activity statistics combined with default emission factors, while Tier 3 represents the most detailed approach, using highly disaggregated local activity data together with locally derived emission factors. The Guidelines generally recommend a bottom-up approach, which focuses on analyzing individual emission sources, rather than a top-down approach that estimates emissions based on aggregated indicators such as total fuel consumption and import-export information.

It is important to note that the 2006 IPCC Guidelines also state that direct emissions monitoring alone is generally not preferred due to the high cost and difficulties related to data interpretation. In particular, direct measurements are often limited to specific locations or facilities, making it challenging to obtain representative samples and to extrapolate results to larger spatial scales, such as an entire industrial sector or a whole country. As a result, fuel-based estimation methods are often preferred for national inventories, as they provide more consistent and scalable coverage [18].

2.1 CO₂ emissions and relevant datasets

As stated above, the IPCC Guidelines serve as a common reference point for most publicly available CO₂ datasets. The scientific community generally agrees that emissions can be estimated as the product of human activity data and an emission factor. However, the approaches to data collection and the choice of how activity data are modeled vary significantly between datasets.

Figure 1 illustrates, in broad strokes, the main steps involved in the creation of a CO₂ emissions estimation dataset. It also highlights the stages at which errors or discrepancies may be introduced.

To construct such a dataset, developers require both activity data and emission factor information. Activity data can be obtained from a variety of sources, including local and national statistical offices, as well as independent international agencies such as the International Energy Agency (IEA) and the Energy Institute. Similar diversity exists for emission factor sources. In addition, data set creators may choose different estimation models and apply various data refinement techniques, for example to address gaps in time series or the lack of sufficiently detailed activity data.

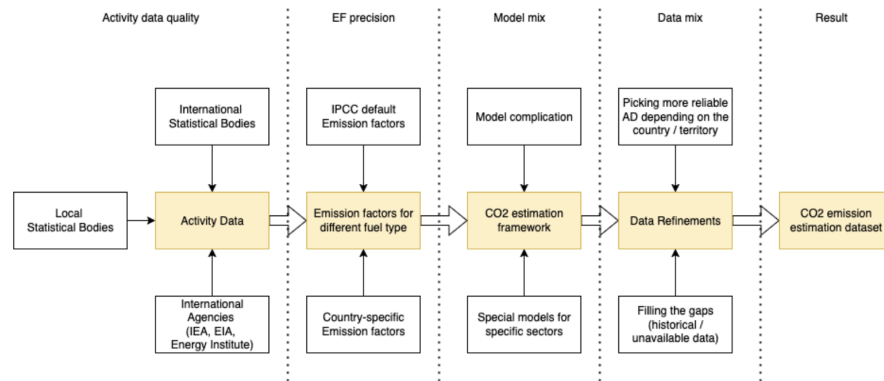


Fig. 1: Main components of a CO₂ estimation data set creation process.

As a result, emission estimates may differ from one data set to another. For example, Figure 2 compares CO₂ emission estimates for Poland and its neighboring country Slovakia, where reported emission levels are significantly lower.

The UNFCCC column represents estimates reported by national authorities and constitutes the official CO₂ emission inventory. These estimates are based on national statistical data covering energy use, industrial activity, and land use, and they typically rely on country-specific emission factors, for example those derived from information on local fuel characteristics such as coal quality.

The EDGAR dataset, developed by the European Union’s Joint Research Centre, is an independent emissions database that primarily uses activity data from the International Energy Agency and, in some cases, generalized emission factors. The third data set, produced by the Global Carbon Project, is particularly interesting because it relies on United Nations data for developed countries. In principle, this should result in estimates similar to those reported under the UNFCCC framework; however, noticeable differences remain.

Overall, no clear systematic bias can be observed across the datasets — for example, EDGAR estimates are not consistently higher or lower than others. All of these databases are widely regarded as credible and are commonly used in scientific analysis and policy development.

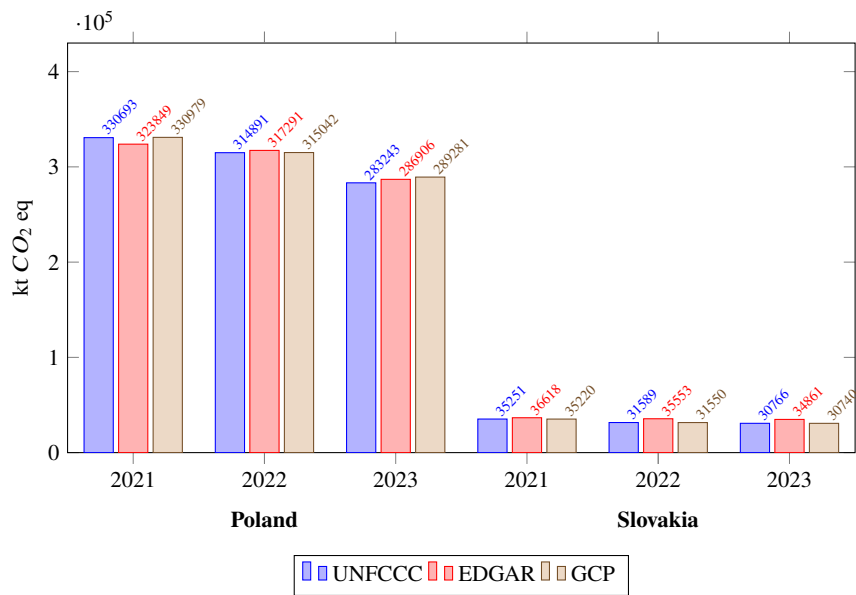


Fig. 2: Annual CO₂ emission estimation in kt CO₂ eq by country and source (2021 - 2023).

Nevertheless, these inconsistencies are important to consider when developing any kind of modeling or prediction project, especially when combining data from multiple sources in machine learning applications.

In the following section, we characterize several publicly available datasets that are widely used in AI and machine learning research.

2.2 Datasets review

The United Nations Framework Convention on Climate Change (UNFCCC) [19] is an international environmental treaty adopted in 1992 in response to growing concerns about climate change. Today, the Convention has been adopted by 197 Parties. Under the UNFCCC framework, countries are expected to report national greenhouse gas inventories in strict accordance with IPCC Guidelines.

The dataset contains annual data for greenhouse gases for each country, currently extending up to 2021. Although the database [20] provides aggregated data on all parties, important distinction exists between different groups of countries. Parties listed in Annex I, that are primarily developed countries and major greenhouse gas emitters, are required to submit detailed national inventory reports on an annual basis. In contrast, developing countries report their emissions through National Communications (NCs) or Biennial Update Reports (BURs), which are submitted less frequently and often with a delay of several years. For example, Chad reports emissions data only for the year 2010.

Using data from different Annex 1 countries could also present challenges, as countries may choose different ways of interpreting activity data and different precise models for emission estimation on the same activity sector.

Another important characteristic of data reported under the United Nations Framework Convention on Climate Change (UNFCCC) is that national inventories are largely based on country-specific data sources and reporting practices. These inventories rely heavily on local governmental activity data, such as national energy and industrial statistics, as well as locally determined fuel properties and emission factors. While this approach can better reflect national conditions, it also means that the quality and transparency of the underlying data may vary between countries. In some cases, this may raise questions about the consistency and reliability of reported statistics.

Global Carbon Budget data: The Global Carbon Project (GCP) is an international research initiative focused on studying the global carbon cycle. Each year, the GCP publishes the Global Carbon Budget report, which provides estimates of global carbon emissions and carbon sinks based on the latest available data and models [21].

In addition to the summary report, the GCP releases several openly available datasets that cover fossil fuel CO_2 emissions, land-use change emissions, and carbon uptake by land and ocean systems. These datasets are provided in both national and gridded formats, include uncertainty estimates, and are updated annually, making them widely used in climate research and data-driven modeling.

Previously, the data set was taken from US-based CDIAC, the Carbon Dioxide Information Analysis Center of the Oak Ridge National Laboratory. CDIAC project was closed in September 30, 2017. Nowadays GCP uses CDIAC-FF data as a base and extend it by 2–3 years using energy growth rates derived from data published by the Energy Institute.

As described in GCP documentation, CDIAC applied standardized emission factors to apparent energy consumption derived from United Nations energy statistics, including emissions from gas flaring. These estimates were then extrapolated for additional

years using growth rates previously obtained from BP’s Statistical Review of World Energy. Apparent consumption is calculated from data on energy production, imports, exports, and stock changes, and differs from observed consumption, which is based on direct industry reporting or sales data. Within the IPCC framework, this approach corresponds to the Reference Approach, used for cross-checking the national inventories, while the Sectoral Approach relies on alternative activity data sources.

In recent years, the GCP has increasingly shifted toward using country-specific and sector-specific energy data sources where available. So sometimes and for some countries estimates derived from CDIAC-FF may be replaced by data reported under the UNFCCC. In another example, emissions from natural gas consumption are now partly derived from data provided by the Joint Organisations Data Initiative (JODI). GCP researchers acknowledge that this results in the use of multiple data sources and extrapolation methods, but argue that these approaches aim to estimate the same underlying quantities and therefore do not significantly alter the overall emission estimates.

Data from the Global Carbon Budget are used by other projects, notably by the Our World in Data project, which in turn serves as a popular open-source solution of CO_2 and greenhouse gas emissions data for machine learning applications [22].

EDGAR, which stands for the Emissions Database for Global Atmospheric Research, is a project developed by the Joint Research Centre of the European Commission [23]. As stated on the project website, EDGAR provides global estimates of past and present anthropogenic emissions of greenhouse gases and air pollutants at the country level and on a spatial grid [24].

EDGAR applies a distinct emission estimation model 2 that combines country-specific emission factor (EF) and activity data (AD), largely derived from assessments by the International Energy Agency, with information on national technological mixes (TECH). The model incorporates technology-dependent emission (EOP) factors and accounts for emission reductions achieved through the use of abatement systems (RED). This approach allows EDGAR to reflect differences in technologies and mitigation measures across countries while maintaining a consistent global framework.

$$EM_c(y, x) = \sum_{i,j,k} [AD_{c,i}(y) \cdot TECH_{c,i,j}(y) \cdot EOP_{c,i,j,k}(y) \cdot EF_{c,i,j}(y, x) \cdot (1 - RED_{c,i,j,k}(y, x))] \quad (2)$$

It is also important to note that EDGAR extends its emissions time series for the most recent years using projected or preliminary energy statistics, including growth rates published by the Energy Institute. This means that the most recent years in the EDGAR dataset are based on extrapolated data rather than fully reported actual activity statistics. For machine learning applications, this introduces additional complexity, as models may be trained on data compiled using different sources and estimation methods across time, which should be taken into account when interpreting results.

2.3 Challenges for machine learning applications

The current state of open-source CO_2 dataset composition presents significant challenges for researchers aiming to apply machine learning algorithms to carbon dioxide emission analysis and emission prediction. In the following, we highlight several of these challenges.

Data are preprocessed. Every CO_2 emissions dataset consists of estimates derived from underlying statistical activity data. This activity data may originate from different institutions that apply distinct methods of data collection and aggregation. Furthermore, missing values in these datasets are often filled using extrapolation based on historical averages or replaced with values from alternative data sources.

This inherent preprocessing can complicate machine learning research, as the data used for training, validation, or reference may not be uniform or fully comparable across datasets.

Low temporal resolution. Most publicly available CO_2 emission datasets provide data at an annual resolution at best, which significantly limits their suitability for many machine learning applications. The lack of higher-frequency observations restricts the ability of models to capture short-term dynamics and temporal variability.

Some authors argue [13] that this limitation can also influence methodological choices, encouraging the use of scaled data and evaluation metrics such as RMSE or MAE computed on normalized values rather than raw observations. As a result, reported model performance may be overstated and less representative of real-world predictive accuracy.

Changing nature of activity data. In some cases, statistical agencies and dataset developers revise historical emission estimates when new or improved activity data become available. In addition, the underlying activity data itself may change due to social, economic, or political actions that are independent of scientific modeling practices. Such changes can introduce structural breaks in emission time series that are not related to actual changes in emission behavior.

A clear example can be found in the most recent Polish National Greenhouse Gas Inventory submitted to the UNFCCC [7]. It is stated that the main cause of significant increase of GHGs emissions in 2016 - 2017 was "substantial rise of fuels use in road transport driven by effective combat against grey-zone at fuel market started in 2016". This suggests that a portion of fuel consumption may not have been fully captured in official statistics prior to this intervention.

From a data analysis perspective, such changes complicate trend estimation and prediction. Emission values before and after 2016 may not be directly comparable, and statistical correction or segmentation of the time series may be required before applying machine learning models for forecasting or trend analysis.

Unaddressed uncertainties. Every CO_2 emission estimation report is subject to uncertainty evaluation. Uncertainty is defined in the IPCC Guidelines [18] as the range of possible true values together with their likelihood. The Intergovernmental Panel on Climate Change (IPCC) does not prescribe a single acceptable uncertainty threshold for emission estimates; instead, it requires Parties to quantify and transparently report uncertainties associated with their inventories.

Uncertainties arise from multiple sources, most notably from the evaluation of emission factors and the aggregation of activity data, and their magnitude varies significantly across sectors and activities. For example, uncertainties in CO_2 emissions from fossil fuel combustion in Annex I UNFCCC countries are typically estimated to be on the order of $\pm 5\%$ to $\pm 10\%$, a range that is also reported for corresponding estimates in the EDGAR dataset [25]. In contrast, activities that are less well characterized exhibit substantially higher uncertainty. Within EDGAR, uncertainty estimates reach approximately $\pm 50\%$ for biogas combustion, $\pm 35\%$ for cement-related emissions, and up to $\pm 100\%$ for domestic aviation emissions [26].

Uncertainties may amplify when combined with estimation models, resulting in possible variation in final estimated values. That is why models trained on CO_2 inventories may achieve good performance with respect to reported values while still failing to accurately represent real-world emissions, particularly if the underlying data contain substantial and heterogeneous uncertainties.

Without explicitly accounting for these uncertainties during model evaluation, machine learning models risk producing overconfident predictions and misleading conclusions. Consequently, careful consideration of data uncertainty should be an integral part of both model development and evaluation when applying machine learning to CO_2 emission analysis.

Lack of domain-specific knowledge. The development of CO_2 emission prediction models is an inherently cross-disciplinary task. Proper interpretation of results requires not only technical expertise in machine learning, but also domain-specific knowledge of climate science and emission accounting.

Figure 3 illustrates a simplified representation of the carbon cycle. Carbon dioxide is emitted both as a result of anthropogenic activities and through natural processes, such as forest fires. At the same time, significant amounts of CO_2 are absorbed by land and ocean systems through various sinks and removal processes. These removals are typically accounted for under Land Use, Land-Use Change and Forestry (LULUCF).

Emissions and removals related to LULUCF are particularly difficult to estimate, which is why this sector is sometimes excluded when carbon emissions are reported. However, the magnitude of these fluxes is far from negligible. According to the 2025 Polish National Inventory [7], net removals from land use in 2023 are estimated at approximately 32 656 kt CO_2 eq, a value comparable to the total annual emissions of Slovakia in the same year.

This example highlights the risk of misinterpretation when domain knowledge is lacking. Machine learning practitioners working with CO_2 datasets should therefore treat emission figures with caution and, where possible, collaborate with domain experts in climate science or ecology to ensure correct interpretation and use of the data.

2.4 Example of prediction of CO_2

To investigate the predictability of CO_2 emissions across different sectors of the economy, separate models were developed for each sector. This sector-based approach allows the models to capture sector-specific relationships and emission dynamics. The

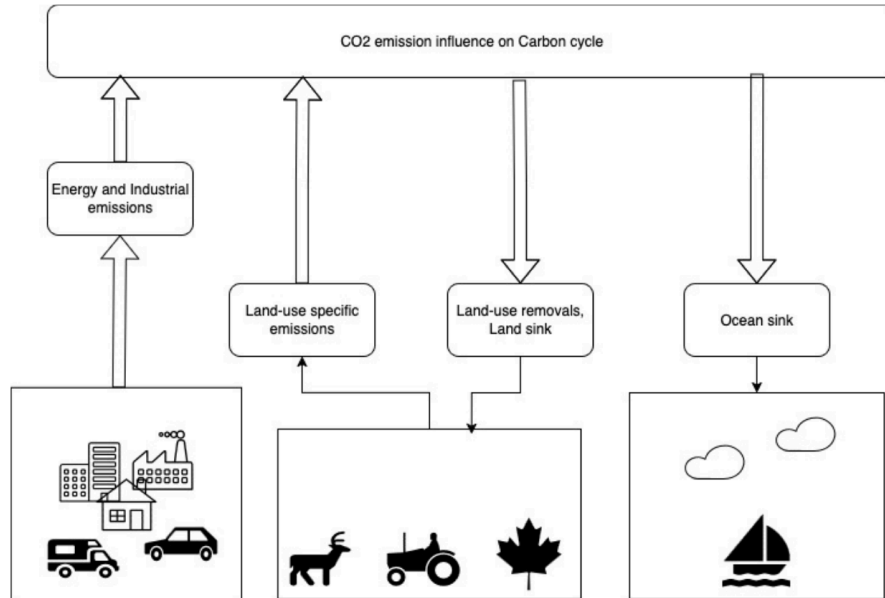


Fig. 3: Basic representation of the carbon cycle.

LightGBM algorithm, a gradient boosting method based on decision trees, was used for modeling. Light Gradient Boosting Machine (LightGBM) is an efficient and scalable gradient boosting framework that is widely used for classification and regression tasks [27]. The data used in this section come from [28] and are further characterized in article Piyu Ke et al. [29].

The data (covering from 2023-2025 for Poland and Slovakia) were split chronologically. The first 80% of observations were used to train the model and learn the relationships present in the data, while the remaining 20% were reserved for evaluating the model's ability to forecast new, unseen values.

The model used several types of features:

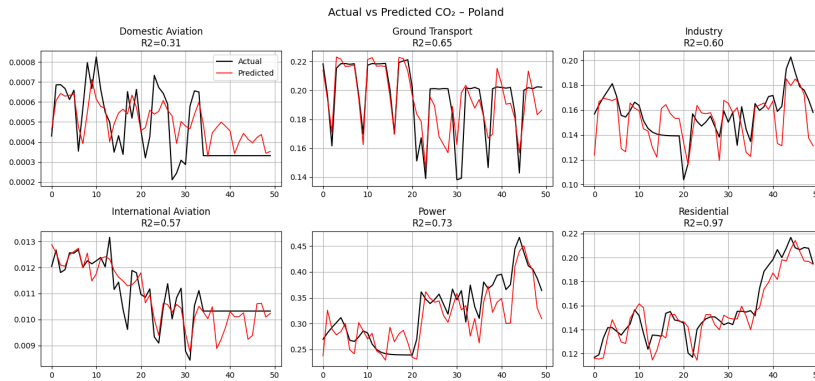
- lagged values from previous days (e.g., 1, 7, 14, and 30 days earlier),
- 7-day and 14-day moving averages,
- calendar variables such as day of the week and month,
- an overall time trend indicating whether emissions increase or decrease over a longer period.

Summarized prediction results for Slovakia and Poland are presented in Table 1. The corresponding visualizations of the predictions are shown in Figures 4 and 5.

In the case of Poland (see Figure 4) the predictability of CO₂ emissions varies across economic sectors. The best model performance was achieved in the Residential sector ($R^2 = 0.97$). This indicates that emissions in this area follow a very clear and repetitive

Table 1: Prediction performance for CO₂ emissions by sector in Slovakia and Poland.

Country	Sector	MAE	RMSE	R^2
Slovakia	Domestic Aviation	0.000007	0.000010	-0.020222
	International Aviation	0.000071	0.000088	0.834955
	Power	0.000743	0.001193	0.849839
	Residential	0.001150	0.001599	0.936084
	Ground Transport	0.001583	0.002228	0.690569
	Industry	0.005361	0.007676	0.407599
Poland	Domestic Aviation	0.000107	0.000136	0.312053
	International Aviation	0.000862	0.001029	0.569176
	Residential	0.004399	0.006512	0.969262
	Ground Transport	0.008854	0.013357	0.646320
	Industry	0.010669	0.013829	0.601022
	Power	0.020775	0.027573	0.733910

Fig. 4: Results of the LightGBM model used for the prediction of CO₂ in Poland.

pattern over time. The model accurately captures both short-term fluctuations and the overall upward trend observed at the end of the analyzed period.

Good results were also obtained in the Power ($R^2 = 0.73$), Ground Transport ($R^2 = 0.65$), and Industry ($R^2 = 0.60$) sectors. In the energy and road transport sectors, emissions change in a relatively regular manner, which allows the model to forecast them with reasonable accuracy. In the industrial sector, the fit is slightly weaker, which may be due to greater irregularity in emissions and the influence of external factors not directly included in the model (e.g., changes in production levels).

The weakest performance was observed in the aviation sectors. For Domestic Aviation ($R^2 = 0.31$) and International Aviation ($R^2 = 0.57$), the quality of predictions is lower. The data contain long periods during which emissions remain constant. This is not due to missing data, but rather to the specific characteristics of the sector. Low variability means there is less information available for the model, making it more difficult to learn patterns and accurately predict future values.

In the case of Slovakia (see Figure 5), the model also performed best in sectors where emissions follow a clear and repetitive pattern over time. The highest prediction quality was achieved in the Residential sector ($R^2 = 0.94$) and the Power sector ($R^2 = 0.85$). This means that emissions in these areas are relatively stable and show clear seasonality, which makes them easier to predict.

Good results were also obtained in International Aviation ($R^2 = 0.83$) and Ground Transport ($R^2 = 0.69$). In these sectors, emissions follow more regular patterns, which helps the model make better forecasts.

Weaker results were observed in the Industry sector ($R^2 = 0.41$). Emissions in industry are more irregular and may depend on external factors, such as changes in production, which were not directly included in the model.

The weakest predictive performance was observed in the Domestic Aviation sector, where the model's forecasts did not fully capture the observed temporal patterns in the data. Similar to Poland, the data show long periods when emissions stay at the same level. This means there is very little change over time. When the data hardly change, the model has little information to learn from, so predictions in this sector are not very accurate.

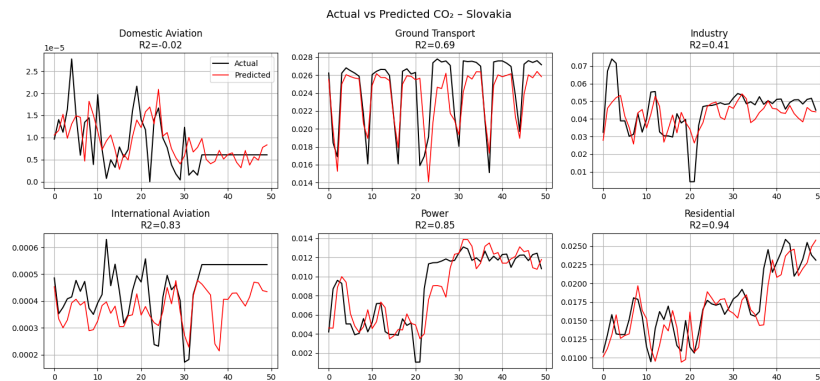


Fig. 5: Results of the LightGBM model used for the prediction of CO_2 in Slovakia.

In both countries, the highest predictability was observed in the Residential and Power sectors. This suggests that emissions in these areas follow the most regular and seasonal patterns, which makes modeling easier. In both countries, Domestic Aviation was the most problematic sector because low variability makes forecasting difficult. The main difference concerns International Aviation. In Poland, the model performance was moderate, while in Slovakia the results were much better. In the Industry sector, predictability is lower in both countries compared to sectors with strong seasonality, although Poland achieved slightly better results than Slovakia. Overall, the models work best in sectors with clear and repetitive emission patterns, and worse in sectors where emissions are low, stable, or irregular.

3 Analysis of $PM_{2.5}$ and PM_{10}

3.1 Factors influencing $PM_{2.5}$ and PM_{10} concentrations

$PM_{2.5}$ and PM_{10} concentrations result from the interaction of emission sources, chemical processes occurring in the atmosphere, and meteorological and spatial conditions. Emission sources can be divided into natural and anthropogenic. Natural sources include, among others, soil erosion, mineral dust transport, sea spray, wildfires, and volcanic eruptions. These phenomena are usually episodic in nature and strongly dependent on weather conditions. However, in many regions anthropogenic sources dominate, such as fuel combustion in the residential and municipal sector, transport, and power generation. Emissions from dispersed, low-height sources favor the accumulation of pollutants in the lower layers of the atmosphere, especially during periods of limited air exchange. Road transport generates both exhaust emissions and non-exhaust emissions (tire and road surface wear, as well as re-suspension of dust). A significant portion of $PM_{2.5}$ mass consists of secondary particles formed in the atmosphere from gaseous precursors such as SO_2 , NO_x , and ammonia. Through chemical reactions, sulfates and nitrates are formed, and their share depends on temperature and air humidity. These processes give particulate concentrations a distinct seasonal pattern [16].

Meteorological conditions play a crucial role in determining $PM_{2.5}$ and PM_{10} levels, as they influence the dispersion and removal of particles from the atmosphere. Even with similar emission levels, changes in weather can cause significant fluctuations in observed concentrations. Low wind speed favors the accumulation of pollutants, whereas stronger winds lead to their dilution and transport to other areas. Precipitation contributes to the removal of particles from the air, with larger PM_{10} particles typically being eliminated more effectively than finer $PM_{2.5}$ particles. Temperature and the vertical structure of the atmosphere are also important. Temperature inversions limit air mixing, promoting the accumulation of pollutants near the ground. Relative humidity can affect particle properties, increasing their mass and modifying the measurement signal. Spatial factors are also significant, such as building density, the presence of major transportation routes, and topography. PM_{10} responds more strongly to local mechanical sources, whereas $PM_{2.5}$ more often reflects the influence of regional transport [16]. As a result, $PM_{2.5}$ and PM_{10} concentrations are not a simple function of emissions but rather the outcome of a complex and nonlinear interaction among emission processes, atmospheric chemistry, and meteorological and spatial conditions. Their analysis requires consideration of temporal and spatial dependencies as well as multiple interacting environmental variables [16].

3.2 $PM_{2.5}$ and PM_{10} data characteristics

In analyses of $PM_{2.5}$ and PM_{10} concentrations, a wide range of factors could potentially be considered, such as sectoral emissions, road traffic intensity, or industrial activity. In practice, however, the availability and usability of such data are significantly limited.

Sectoral emission data are typically published as annual inventories and at an administrative scale (e.g., national or regional), which makes it impossible to directly link them with concentration measurements at an hourly resolution. Similarly, traffic intensity data are usually point-based, cover selected road sections, and do not always spatially coincide with the locations of air quality monitoring stations. As a result, the use of these variables would require numerous assumptions regarding spatial and temporal interpolation, which could increase analytical uncertainty. For this reason, the following analysis focuses on meteorological variables, which are available in a consistent format and at high temporal resolution, and can be directly integrated with PM measurement data.

3.3 OpenAQ platform

OpenAQ [30] is a global platform that collects air quality data from multiple countries and makes them available in a standardized format. It does not conduct its own measurements; instead, it integrates information from public monitoring systems, such as national environmental protection agencies and regional monitoring networks. In this way, the platform aggregates dispersed data from around the world and harmonizes them by standardizing measurement units (e.g., $\mu\text{g}/\text{m}^3$), parameter names, and time formats (most commonly UTC). OpenAQ does not modify the measurement values themselves; rather, it organizes their technical structure and makes them accessible to a broad range of users, enabling further use in analysis and research.

The data provided by OpenAQ primarily consist of real-time measurements of air pollutant concentrations, most commonly particulate matter ($\text{PM}_{2.5}$ and PM_{10}) and selected gaseous pollutants such as NO_2 , SO_2 , and O_3 . Values are typically expressed in $\mu\text{g}/\text{m}^3$ or ppb, depending on the pollutant type, and are assigned to specific monitoring stations. Each observation is linked to a geographic location, a timestamp, and the type of measured parameter. From an analytical perspective, these data are tied to a specific station and moment in time, allowing both temporal analyses (e.g., diurnal and seasonal patterns) and spatial comparisons across different locations. The platform is open and transparent, which facilitates integration with other datasets, such as meteorological data. At the same time, users should account for potential gaps in time series, differences in reporting frequency, and changes in station operation, all of which may affect data completeness and continuity [30].

3.4 Meteostat

Meteostat is an open-source meteorological data platform based on measurements from stations operated by national meteorological services. The service does not generate its own observations; instead, it aggregates and organizes data provided by official institutions such as NOAA (National Oceanic and Atmospheric Administration) or DWD (Deutscher Wetterdienst), presenting them in a standardized format. Meteostat provides

data through several technical interfaces. These include a JSON-based API, a Python library that enables data analysis (e.g., using the Pandas library), and the option to download bulk datasets for individual meteorological stations. The range of available variables includes key atmospheric parameters, such as air temperature, relative humidity, wind speed and direction, precipitation, and atmospheric pressure. Depending on the location, additional information may also be available, such as cloud cover or visibility. Data are published, among others, at hourly resolution, enabling their direct integration with $PM_{2.5}$ and PM_{10} concentration measurements [31].

3.5 Integration of environmental data

The integration of air quality data and meteorological data is based on spatial and temporal matching of observations. In practice, this means linking measurements using the geographic coordinates of stations and their timestamps. Each PM monitoring station is assigned data from the nearest meteorological station, assuming it is representative of the given area, and observations are synchronized at the same temporal resolution—most commonly hourly.

However, it should be emphasized that meteorological stations and PM monitoring stations are not always located in the exact same place. This constitutes a simplification that may introduce additional uncertainty, particularly in areas with complex topography. The dataset used is not merely a straightforward record of environmental conditions. It is also shaped by the measurement technologies applied, the adopted procedures, and the organization of the monitoring system. Different devices may measure particulate matter slightly differently, stations are situated in specific types of locations, and data are further processed and standardized before being made available. This means that the final structure of the dataset is influenced not only by atmospheric conditions themselves, but also by the way they are observed. Predictive models therefore learn not only the relationships between meteorological conditions and particulate concentrations, but also characteristics of the measurement system from which the data originate. If the dataset contains gaps, differences between stations, or technological changes over time, the model may partially “learn” these patterns as well. A high number of observations and a standardized data format do not guarantee full comparability of all measurements. Therefore, when interpreting model results, it is important to remember that the data represent an organized record of observations rather than a perfect and entirely neutral reflection of reality.

3.6 Example of PM prediction

The analysis combined hourly $PM_{2.5}$ and PM_{10} concentration data from the OpenAQ (3.3) platform with meteorological variables obtained from Meteostat (3.4), including air temperature, relative humidity, wind speed, precipitation, and atmospheric pressure. Both datasets covered the same location. The datasets were integrated by matching ob-

servations in space and time. Each air quality monitoring station was linked to the nearest meteorological station based on geographic coordinates. The data were then aligned to the same hourly resolution, creating a single dataset for modeling.

The goal of the experiment was to build two predictive models to forecast hourly $PM_{2.5}$ and PM_{10} concentrations during the heating season. In each case, the dependent variable was the concentration of $PM_{2.5}$ or PM_{10} (in $\mu\text{g}/\text{m}^3$), while the independent variables included selected meteorological factors and time-related variables.

The models used current meteorological values as well as their time lags and moving averages (e.g., 6-hour and 12-hour averages). Calendar variables, such as hour of the day and month, were also included to capture daily and seasonal patterns.

The dataset was split chronologically into training and testing parts to reflect real forecasting conditions. Model performance was evaluated using time-series cross-validation. This method provides a more reliable estimate of predictive performance than a single train–test split while preserving the time order of observations.

The reported root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2) values are averages from multiple sequential splits. The input data were not standardized or normalized because tree-based gradient boosting models, such as LightGBM, are not sensitive to the scale of input variables. Therefore, feature scaling was not required.

The prediction results are summarized in Table 2.

Table 2: Predictive performance of the LightGBM models for $PM_{2.5}$ and PM_{10} concentrations.

	RMSE [$\mu\text{g}/\text{m}^3$]	MAE [$\mu\text{g}/\text{m}^3$]	R^2
$PM_{2.5}$	4.08	2.83	0.93
PM_{10}	3.08	2.34	0.78

The model predicting $PM_{2.5}$ concentrations achieved a clearly higher R^2 value (0.93) than the model predicting PM_{10} concentrations (0.78). This suggests that $PM_{2.5}$ concentrations are more strongly influenced by the meteorological and time-related variables included in the model.

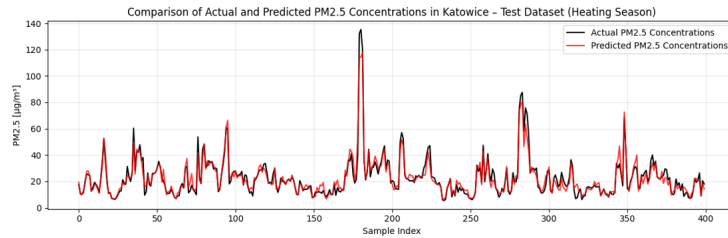


Fig. 6: Results of the LightGBM model used for the $PM_{2.5}$ prediction.

The visualization of the predicted data is presented in Figure 6. The model correctly captured the general pattern of the $PM_{2.5}$ time series, including short-term changes and periods of high concentrations typical of the heating season. However, it slightly underestimated extreme values, which may be due to the tendency of tree-based models to smooth out very high observations.

A similar experiment was carried out for the PM_{10} fraction during the heating season (see Figure 7).

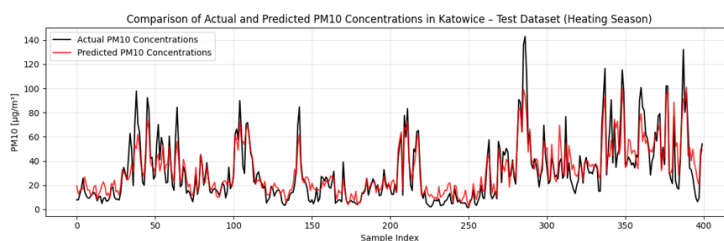


Fig. 7: Results of the LightGBM model used for the PM_{10} prediction.

The model correctly reproduced the overall pattern of the time series, including periods of higher concentrations. Compared to $PM_{2.5}$, the PM_{10} predictions showed greater short-term variability and responded more clearly to sudden increases. However, similar to the finer fraction, extreme values were still partly underestimated, especially during sharp spikes in concentrations.

4 Discussion

Data-driven analyses of greenhouse gas and particulate matter emissions have become increasingly feasible due to the availability of open-source datasets and high-resolution environmental measurements. These resources provide researchers with the opportunity to investigate emission dynamics, develop predictive models, and explore the interactions between anthropogenic activities, atmospheric processes, and meteorological conditions.

Despite their limitations, open-source CO_2 emission estimation datasets play an important role in enabling data-driven research. They provide computer scientists with accessible inputs for developing and testing machine learning models that incorporate diverse forms of activity data and use estimated emissions as target variables. As such, these datasets have contributed significantly to methodological exploration and comparative analysis in emission modeling.

At the same time, the limitations identified in this work strongly support the case for expanded and more consistent monitoring coverage. Advances in sensing technologies and large-scale data acquisition systems offer the potential to collect higher-resolution and more direct measurements of CO_2 emissions. Such data could improve our under-

standing of emission dynamics and support the development of more advanced predictive models that rely less on indirect statistical activity data and strong estimation assumptions. Improving measurement coverage and data quality is therefore a key step toward more reliable machine learning applications in carbon emission analysis.

The results highlight that the predictability of CO₂ emissions strongly depends on the economic sector. In both Slovakia and Poland, the Residential and Power sectors exhibited the highest predictability, reflecting clear, repetitive, and seasonal emission patterns. In contrast, Domestic Aviation consistently showed the weakest performance due to low variability and long periods of nearly constant emissions, which limit the information available for model learning. As described in [26], the aviation sector also tends to exhibit higher levels of error in the estimation process. Intermediate results were observed for Ground Transport and Industry, where emissions are more irregular or influenced by external factors such as production changes. A notable difference between countries was found in International Aviation: predictability was moderate in Poland but considerably higher in Slovakia, suggesting differences in sector-specific emission dynamics. Overall, these findings indicate that sector-specific modeling is essential for accurately forecasting CO₂ emissions, and that model performance is largely determined by the regularity and variability of emissions within each sector.

Similarly, the predictive modeling of $PM_{2.5}$ and PM_{10} concentrations demonstrates the crucial role of meteorological and temporal factors in shaping particulate levels. The models successfully captured general temporal patterns, including short-term fluctuations and seasonal peaks, although extreme values were somewhat underestimated. The results further suggest that finer particles ($PM_{2.5}$) are more strongly influenced by the predictors used, while both fractions benefit from the integration of high-resolution meteorological data. These findings underscore the importance of considering complex, nonlinear interactions among emissions, atmospheric chemistry, and weather conditions when modeling particulate matter concentrations.

Taken together, the analyses of both CO₂ and particulate matter highlight that predictive modeling is most effective when the underlying environmental patterns are regular, seasonal, and well-characterized. At the same time, low variability, episodic events, and irregular emission dynamics pose challenges that must be addressed through improved monitoring, data integration, and methodological refinement. These insights point to future directions for enhancing the reliability and accuracy of machine learning approaches in air quality including carbon emission research.

Acknowledgement

This work was supported by the AI2SEP project (No. 2023-1-PL01-KA220-HED-000166765). We gratefully acknowledge the project's support and funding, which enabled this research. The authors declare that they have no competing interests relevant to the content of this chapter.

References

1. World Health Organization. (2024). *Ambient (outdoor) air pollution* [Fact sheet]. [https://www.who.int/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)
2. Apte, J. S., Marshall, J. D., Cohen, A. J., & Brauer, M. (2015). Addressing global mortality from ambient PM_{2.5}. *Environmental Science & Technology*, 49(13), 8057–8066. <https://doi.org/10.1021/acs.est.5b01236>
3. European Commission. (2020). *Impact assessment – Stepping up Europe’s 2030 climate ambition: Investing in a climate-neutral future for the benefit of our people* (SWD(2020) 176 final). European Commission.
4. Harishkumar, K., Yogesh, K., & Gadicha, M. (2020). Forecasting air pollution particulate matter (PM_{2.5}) using machine learning regression models. *Procedia Computer Science*, 171, 2057–2066. <https://doi.org/10.1016/j.procs.2020.04.221>
5. Masood, A., & Ahmad, K. (2020). A model for particulate matter (PM_{2.5}) prediction for Delhi based on machine learning approaches. *Procedia Computer Science*, 167, 2101–2110. <https://doi.org/10.1016/j.procs.2020.03.258>
6. Kumari, S., & Singh, S. K. (2023). Machine learning-based time series models for effective CO₂ emission prediction in India. *Environmental Science and Pollution Research*, 30(55), 116601–116616. <https://doi.org/10.1007/s11356-023-30514-x>
7. National Centre for Emission Management (KOBiZE). (2025). *Poland. 2025 National Inventory Document (NID): National greenhouse gas inventory submission under the UNFCCC*. KOBiZE.
8. European Environment Agency. (2025). *Annual European Union greenhouse gas inventory 1990–2023 and inventory document 2025* (EEA/PUBL/2025/024). European Environment Agency.
9. Ma, N., Shum, W. Y., Han, T., & Lai, F. (2021). Can machine learning be applied to carbon emissions analysis: An application to the CO₂ emissions analysis using Gaussian process regression. *Frontiers in Energy Research*, 9, Article 756311. <https://doi.org/10.3389/fenrg.2021.756311>
10. Ghorbal, A. B., Grine, A., Elbatal, I., Al-Mofleh, H., & El-Saeed, A. R. (2025). Predicting carbon dioxide emissions using deep learning and Ninja metaheuristic optimization algorithm. *Scientific Reports*, 15(1), Article 4021. <https://doi.org/10.1038/s41598-025-83214-z>
11. Salem, K. M., Rey-Hernández, J. M., Rey-Martínez, F. J., & Elgharib, A. O. (2025). Assessing the accuracy of AI approaches for CO₂ emission predictions in buildings. *Journal of Cleaner Production*, 513, Article 145692. <https://doi.org/10.1016/j.jclepro.2025.145692>
12. Al Nuaimi, H. S., Acquaye, A., & Mayyas, A. (2025). Machine learning applications for carbon emission estimation. *Resources, Conservation and Recycling Advances*, 27, Article 200263. <https://doi.org/10.1016/j.rcradv.2025.200263>
13. Begum, A. M., & Mobin, M. A. (2025). A machine learning approach to carbon emissions prediction of the top eleven emitters by 2030 and their prospects for meeting Paris agreement targets. *Scientific Reports*, 15(1), Article 19469. <https://doi.org/10.1038/s41598-025-04236-5>
14. European Parliament & Council of the European Union. (2008). Directive 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on ambient air quality and cleaner air for Europe. *Official Journal of the European Union*, 152, 1–44.
15. Jayaratne, R., Liu, X., Thai, P., Dunbabin, M., & Morawska, L. (2018). The influence of humidity on the performance of a low-cost air particle mass sensor and the effect of atmospheric fog. *Atmospheric Measurement Techniques*, 11(8), 4883–4890. <https://doi.org/10.5194/amt-11-4883-2018>
16. World Health Organization. (2021). *WHO global air quality guidelines: Particulate matter (PM_{2.5} and PM₁₀), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*. World Health Organization.
17. United Nations Economic Commission for Europe. (2021). UN Regulation No 154 – Uniform provisions concerning the approval of light duty passenger and commercial vehicles with regards to criteria emissions, emissions of carbon dioxide and fuel consumption and/or the measurement of electric energy consumption and electric range (WLTP) [2021/2039]. *Official Journal of the European Union*, 423, 1–210. <http://data.europa.eu/eli/reg/2021/2039/oj>
18. Eggleston, H. S., Buendia, L., Miwa, K., Ngara, T., & Tanabe, K. (Eds.). (2006). *2006 IPCC guidelines for national greenhouse gas inventories*. Institute for Global Environmental Strategies.

19. United Nations Framework Convention on Climate Change. (1992). *United Nations Framework Convention on Climate Change*. UNFCCC Secretariat.
20. UNFCCC Secretariat. (2025). *GHG data from UNFCCC*. <https://unfccc.int/topics/mitigation/resources/registry-and-data/ghg-data-from-unfccc>
21. Friedlingstein, P., O’Sullivan, M., Jones, M. W., Andrew, R. M., Hauck, J., Olsen, A., Peters, G. P., Peters, W., Pongratz, J., Schwingshackl, C., Sitch, S., Canadell, J. G., Ciais, P., Jackson, R. B., Alin, S. R., Anthoni, P., Barbero, L., Bates, N. R., Becker, M., ... Zhu, D. (2025). Global Carbon Budget 2024. *Earth System Science Data*, 17(3), 965–1039. <https://doi.org/10.5194/essd-17-965-2025>
22. Ritchie, H., Rosado, P., & Roser, M. (2023). *CO2 and greenhouse gas emissions*. Our World in Data. <https://ourworldindata.org/co2-and-greenhouse-gas-emissions>
23. Directorate-General for Joint Research Centre. (2025). *EDGAR - Emissions Database for Global Atmospheric Research*. European Commission. <https://edgar.jrc.ec.europa.eu/methodology>
24. Crippa, M., Guizzardi, D., Pagani, F., Banja, M., Muntean, M., Schaaf, E., Monforti-Ferrario, F., Becker, W. E., & Vignati, E. (2024). *GHG emissions of all world countries* (JRC138862). Publications Office of the European Union. <https://doi.org/10.2760/4002897>
25. Banja, M., Crippa, M., Guizzardi, D., Muntean, M., Pagani, F., & Pisoni, E. (2025). A comparative analysis of EDGAR and UNFCCC GHG emissions inventories: Insights on trends, methodology and data discrepancies. *Earth System Science Data*, 17(11), 6461–6486. <https://doi.org/10.5194/essd-17-6461-2025>
26. Solazzo, E., Crippa, M., Guizzardi, D., Muntean, M., Choulga, M., & Janssens-Maenhout, G. (2021). Uncertainties in the Emissions Database for Global Atmospheric Research (EDGAR) emission inventory of greenhouse gases. *Atmospheric Chemistry and Physics*, 21(7), 5655–5683. <https://doi.org/10.5194/acp-21-5655-2021>
27. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 3146–3154.
28. Ke, P., Deng, Z., Zhu, B., Zheng, B., Wang, Y., Boucher, O., Arous, S. B., Zhou, C., Dou, X., Sun, T., Li, Z., Yan, F., Cui, D., Hu, Y., Huo, D., Pierre, J., Engelen, R., Davis, S. J., Ciais, P., & Liu, Z. (2022). Carbon Monitor Europe, near-real-time daily CO2 emissions for 27 EU countries and the United Kingdom. *Scientific Data*, 9(1), Article 687. <https://doi.org/10.1038/s41597-022-01761-2>
29. Ke, P., Deng, Z., Zhu, B., Zheng, B., Wang, Y., Boucher, O., Arous, S. B., Zhou, C., Andrew, R. M., Dou, X., Sun, T., Song, X., Li, Z., Yan, F., Cui, D., Hu, Y., Huo, D., Chang, J.-P., Engelen, R., ... Liu, Z. (2023). Carbon Monitor Europe near-real-time daily CO2 emissions for 27 EU countries and the United Kingdom. *Scientific Data*, 10(1), Article 374. <https://doi.org/10.1038/s41597-023-02284-y>
30. OpenAQ. (2025). *OpenAQ documentation*. <https://docs.openaq.org>
31. Meteostat. (2025). *Meteostat developer documentation*. <https://dev.meteostat.net/overview>
32. Gilfillan, D., & Marland, G. (2021). CDIAC-FF: Global and national CO2 emissions from fossil fuel combustion and cement manufacture: 1751–2017. *Earth System Science Data*, 13(4), 1667–1880. <https://doi.org/10.5194/essd-13-1667-2021>
33. W3C. (2008). *Web content accessibility guidelines (WCAG) 2.0*. <https://www.w3.org/TR/WCAG20/>
34. Alam, G. M. I., Arfin Tanim, S., Sarker, S. K., Hasan, M. M., & Islam, M. S. (2025). Deep learning model based prediction of vehicle CO2 emissions with eXplainable AI integration for sustainable environment. *Scientific Reports*, 15, Article 3655. <https://doi.org/10.1038/s41598-025-87233-y>
35. Andrew, R. M., & Peters, G. P. (2024). *The Global Carbon Project’s Fossil CO2 Emissions Dataset* (v18) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.14106218>
36. Anita, W. M., Ueda, K., Uttajug, A., Seposo, X. T., & Takano, H. (2023). Association between long-term ambient PM2.5 exposure and under-5 mortality: A scoping review. *International Journal of Environmental Research and Public Health*, 20(4), Article 3270. <https://doi.org/10.3390/ijerph20043270>
37. Ayaz, I. (2024). Forecasting CO2 emissions with machine learning methods: Türkiye example and future trends. *Naturengs*, 5(2), 82–87.

About authors

Ivan Maslov is a student of Applied Computer Science at the University of Silesia in Katowice. He focuses on software development and modern IT technologies. He has experience working with digital tools, Python, as well as creating and editing multimedia content. His interests include software systems, media, and emerging technologies, combining a technical background with experience in data-driven environments.

Agnieszka Głowacka is a first-year Master's student of Micro- and Nanotechnology at the University of Silesia in Katowice. Her interests focus on the application of modern technologies in the analysis and processing of scientific data. She is particularly interested in interdisciplinary approaches combining elements of physics, chemistry, and informatics, as well as practical applications of advanced technologies in science and industry.

Bartosz Dzięwit is an assistant professor at the Faculty of Science and Technology of the University of Silesia in Katowice, affiliated with the Institute of Physics, and currently serves as the Director of the Applied Computer Science program. His research focuses on particle physics (especially neutrino physics), data analysis, and computer science, and he is actively involved in teaching and supervising students in areas such as computer systems, networks, and cybersecurity.

Paulina Trybek is an assistant professor at the University of Silesia in Katowice, affiliated with the Institute of Physics, where she specializes in the analysis of biomedical time series. She is actively involved in numerous student projects, supporting the development of data analysis competencies. She is also the coordinator of the project "Developing Talents in Artificial Intelligence to Solve Disruptive Environmental Problems".



University of Maribor Press

Nanoplastics and Biostructures: Exploring the Capabilities of MD Computer Simulations

Jakub Kraus,
Piotr Mika,
Mateusz Wiśniewski,
Paulina Trybek, and
Zbigniew Dendzik

Abstract

Micro- and nanoplastics (MNPs) represent a class of emerging contaminants whose diminutive dimensions, extensive surface area, and chemical resilience facilitate intimate interactions with biomacromolecules. Owing to their ability to traverse biological barriers, MNPs accumulate in tissues and directly engage proteins, nucleic acids, and lipid assemblies, thereby perturbing structure and function. Nanoplastics, in particular, adsorb onto protein surfaces, disrupt secondary and tertiary conformations, and partition within lipid bilayers. Molecular dynamics (MD) simulations provide atomistic insight into these processes—adsorption, corona formation, membrane insertion, and conformational modulation—complementing *in vitro*, *in vivo*, and epidemiological investigations. Current knowledge on MNP-induced alterations of protein architecture and membrane integrity is synthesized, integrating toxicological data, biophysical measurements, and MD results. A streamlined MD workflow is presented for nanoplastic–protein and nanoplastic–membrane systems, outlining key structural, thermodynamic, and dynamical observables (e.g., protein RMSD, membrane order parameters, surface coverage, interaction energies) that correlate with experimental endpoints such as enzymatic inhibition, membrane leakage, oxidative stress, and inflammation. Finally, conceptual and methodological challenges in linking atomistic mechanisms to adverse outcome pathways and risk assessment are discussed.

Keywords: nanoplastics; molecular dynamics; computer simulations; protein corona; lipid bilayer; biomolecular interactions; toxicology

Jakub Kraus and Piotr Mika
University of Silesia, Faculty of Science and Technology, 40-007 Katowice, Poland
e-mail: jakub.kraus@us.edu.pl, piotr.mika@us.edu.pl,
Mateusz Wiśniewski, Paulina Trybek and Zbigniew Dendzik
University of Silesia, Institute of Physics, 40-007 Katowice, Poland
e-mail: mateusz.wisniewski@us.edu.pl, paulina.trybek@us.edu.pl, zbigniew.dendzik@us.edu.pl

1 Introduction

Plastic production has grown to hundreds of millions of tons per year, with a substantial fraction entering aquatic and terrestrial ecosystems as waste [2]. Weathering, mechanical abrasion, and (bio)chemical degradation fragment bulk polymers into microplastics (< 5 mm) and eventually nanoplastics (1-100 nm) [2]. Their small size facilitates long-range transport, cellular uptake, and translocation across physiological barriers including intestinal epithelium, gills, and the blood-brain barrier [2, 1, 10].

At nanoscale dimensions, plastic particles exhibit behavior reminiscent of engineered nanomaterials, including high specific surface area, strong adsorption of biomolecules, and size- and surface-dependent toxicity [2, 1]. Experimental studies have shown that nanoplastics can be internalized by diverse cell types, accumulate in tissues, perturb metabolic and signaling pathways, and induce oxidative stress, inflammation, and neurotoxicity [2, 1]. At the same time, *in vivo* concentrations and exposure regimes often differ from those used in laboratory tests, leading to uncertainty in environmental and human health risk assessment [2, 1].

A critical open question is how MNPs perturb the structure and function of specific biomolecules and assemblies, such as enzymes, structural proteins, lipid bilayers, nucleic acids, and protein complexes. Nanoplastics can form protein coronas that modulate cellular uptake [13, 14, 15, 16, 17], bind directly to protein surfaces and alter secondary structure [3], or dissolve into lipid bilayers and modify membrane organization and dynamics [4, 9, 11]. Addressing these questions requires integration of toxicology, biophysics, and molecular simulation.

Molecular dynamics simulations are uniquely suited to resolve the nanoscale and molecular mechanisms of MNP-biomolecule interactions, as demonstrated in recent MD studies of nanoplastic-membrane systems and nanoplastic-protein model systems [3, 4]. Their theoretical foundations and algorithmic details are described extensively elsewhere [5, 6, 7, 12]. In parallel, adverse outcome pathway (AOP) frameworks, originally developed for traditional toxicants, are being adapted to micro- and nanoplastics to provide mechanistic links from molecular initiating events (MIEs), such as reactive oxygen species (ROS) generation or membrane perturbation, to organism- and population-level adverse outcomes [1].

The aims of this article are therefore four-fold:

1. To summarize the current experimental and computational evidence for the effects of micro- and nanoplastics on protein structure and other biological molecules [2, 3, 4, 1].
2. To introduce MD methodology in this context, emphasizing system construction, force-field selection, and sampling strategies for nanoplastic-biomolecule systems [5, 6, 7, 12, 4].
3. To propose and illustrate a concrete set of test MD simulations for nanoplastic-protein and nanoplastic-membrane interactions, including practical observables and their interpretation.
4. To review which physical quantities can be extracted from MD trajectories and how they relate to experimentally relevant toxicity endpoints and AOP concepts [1].

2 Micro and nano plastics in Biological Systems

2.1 Sources, sizes, and environmental occurrence

Macro-scale plastic waste in oceans, freshwaters, soils, and sediments gradually fragments into micro- and nanoplastics via combined action of photodegradation, thermal oxidation, mechanical abrasion, and biodegradation [2]. Microplastics originate both from primary sources (e.g. cosmetic microbeads, industrial pellets) and secondary fragmentation, while nanoplastics arise predominantly as secondary particles, though some commercial nanoplastic applications exist [2]. The dominant commodity polymers include polystyrene (PS), polyethylene (PE), polypropylene (PP), polyvinyl chloride (PVC), and polyethylene terephthalate (PET) [2].

Nanoplastics are difficult to detect and quantify in environmental matrices due to their small size, low mass, and interference from natural colloids [2]. Consequently, estimates of environmental concentrations remain uncertain, complicating the extrapolation of laboratory toxicity data to real-world scenarios [2]. Standardization of sampling, isolation, and analytical protocols for nanoscale plastic particles remains an active area of research [2].

2.2 Uptake, translocation, and bioaccumulation

In aquatic organisms, PS and PMMA nanoplastics have been shown to be ingested by invertebrates (e.g. daphnids, rotifers, copepods, mussels) and vertebrates (e.g. fish), distribute to tissues including gills, liver, and gonads, and even cross the blood-brain barrier [2, 1]. In some cases, nanoplastics are transferred along food chains (algae → zooplankton → fish), affecting feeding behavior and lipid metabolism in higher trophic levels [2].

In mammalian models and *in vitro* intestinal models, PS and PET nanoplastics can cross epithelial barriers with low but non-negligible bioavailability, enter the circulation, and reach secondary organs [2, 1]. Surface chemistry (e.g. amine vs. carboxyl functionalization) and size strongly influence uptake, intracellular trafficking, and toxicity [2, 1]. These observations confirm that nanoplastics can reach cellular and subcellular compartments where direct molecular interactions with proteins, membranes, and nucleic acids become possible [10].

2.3 General toxicological endpoints

Across species, frequently reported endpoints of MNP exposure include:

- Oxidative stress (increased ROS, altered antioxidant enzyme activities; impaired redox homeostasis) [1].

- Inflammation (cytokine induction, immune cell activation, tissue infiltration and damage) [1].
- Neurotoxicity (acetylcholinesterase inhibition, altered neurotransmitter levels, behavioral changes such as reduced locomotion and altered feeding) [2, 1].
- Growth inhibition, developmental delays, reproductive toxicity, and altered energy metabolism in multiple taxa [2, 1].

AOP-based analyses highlight ROS generation as a central MIE for both micro- and nanoplastics, with oxidative stress and inflammatory signaling as key events leading to growth inhibition, behavioral changes, and other adverse outcomes [1]. For nanoplastics in particular, the high surface-to-volume ratio and capacity to cross cellular membranes amplify the importance of these molecular events [1].

3 Direct effects on proteins and other biomolecules

3.1 Adsorption and protein corona formation

When nanoplastics enter biological fluids, they rapidly adsorb proteins and other biomolecules, forming a “protein corona” that defines the biological identity of the particle and strongly influences cellular uptake and downstream responses [2, 1]. Coronas on PS nanoplastics can include plasma proteins, lipoproteins, complement factors, and pattern-recognition molecules, altering immune recognition, transport properties, and biodistribution [2, 1, 10].

From a structural perspective, adsorption can:

- Partially unfold proteins or shift conformational equilibria between folded, misfolded, and intermediate states [9].
- Mask or expose functional motifs (e.g. receptor-binding domains, catalytic residues).
- Promote protein-protein crosslinking or aggregation at the plastic surface, with potential amyloid-like behavior in some cases.

These effects depend strongly on surface chemistry, curvature, and heterogeneity of the nanoplastic, variables that can be systematically explored with all atom and/or coarse-grained MD simulations [5, 7].

3.2 Nanoplastics and secondary structure in model peptides

Atomistic MD simulations have begun to clarify how nanoplastics perturb secondary structure in model peptides. Hollóczy and co-workers investigated the interaction of plastic nanoparticles (including PS-like and nylon-6,6 particles) with helical peptides and tryptophan zipper motifs using explicit-solvent MD and structural analysis [3]. Ramachandran plots and secondary-structure assignments revealed that adsorption to

polymer nanoparticle surfaces can alter α -helix and β -hairpin stability, shifting dihedral angle distributions relative to neat aqueous solution [3]. Hydrophobic side chains showed strong affinity for hydrophobic plastic surfaces, and adsorption changed the local backbone conformational landscape, with potential implications for misfolding or aggregation [3, 9].

In classical backbone dihedral-angle space, conformational preferences can be described by a probability density $p(\phi, \psi)$, from which a conformational free-energy surface

$$F(\phi, \psi) = -k_B T \ln p(\phi, \psi)$$

can be derived [5, 6, 9]. Here k_B is the Boltzmann constant, T the absolute temperature, and $p(\phi, \psi)$ the normalized probability of observing a given pair of dihedral angles. Nanoplastic-induced shifts in $p(\phi, \psi)$ therefore correspond to changes in the free-energy landscape of peptide secondary structure [9, 18].

These simulations demonstrate that even simple hydrophobic nanoplastics can influence the free-energy landscape of peptide secondary structure, particularly for motifs with large hydrophobic exposure or marginal stability [3, 9, 8].

3.3 Nanoplastics and membrane proteins

Many membrane proteins rely on specific lipid environments and membrane physical properties (thickness, curvature, lateral pressure profile) for proper folding, dynamics, and function [11, 10]. Any nanoplastic-induced perturbation of the bilayer may therefore secondarily alter membrane protein structure and activity. MD simulations of polyethylene nanoplastics in phosphatidylcholine bilayers show that nanoparticles dissolve into the hydrophobic core, disintegrating into single polymer chains that significantly increase bilayer thickness, alter lipid tail conformations, and enhance lateral lipid diffusion [4].

Bilayer thickness d_{bilayer} is commonly defined as the average distance between phosphorus atoms in opposing lipid leaflets [11]:

$$d_{\text{bilayer}} = \left\langle z_{\text{P}}^{(\text{upper})} - z_{\text{P}}^{(\text{lower})} \right\rangle,$$

where $z_{\text{P}}^{(\text{upper})}$ and $z_{\text{P}}^{(\text{lower})}$ are the instantaneous z -coordinates of headgroup phosphorus atoms in the upper and lower leaflet, and $\langle \rangle$ denotes a time and molecule average. Holl czki and Gehrke observed an increase of ~ 1.1 nm ($\sim 27\%$) in d_{bilayer} upon incorporation of PE nanoplastics into a POPC bilayer [4].

Lipid tail ordering is often quantified via the deuterium order parameter

$$S_{\text{CD}} = \frac{1}{2} \langle 3 \cos^2 \theta - 1 \rangle,$$

where θ is the angle between the C-D (or C-H) bond vector and the bilayer normal. Increased S_{CD} corresponds to more ordered, extended hydrocarbon chains. Nanoplastic-

induced changes in S_{CD} provide a direct measure of how polymer incorporation modifies chain conformations [4].

These physicochemical changes can in principle:

- Shift helix tilt and rotation for transmembrane segments.
- Alter oligomerization equilibria of membrane protein complexes.
- Perturb gating, transport, or signaling processes that depend sensitively on membrane mechanics and thickness [11, 10].

While detailed MD studies of explicit membrane proteins in the presence of nanoplastics remain scarce, the membrane-centric results strongly suggest that MNPs can indirectly affect membrane protein structure and function via bilayer remodeling [4, 11, 1].

3.4 Other biomolecules: nucleic acids and carbohydrates

Direct MD work on nanoplastic-nucleic acid interactions is limited, but by analogy to carbon nanomaterials and other hydrophobic nanoparticles, one can anticipate base stacking interactions with hydrophobic surfaces and potential destabilization of local DNA/RNA structure, especially in unpaired or loop regions [7, 1]. Carbohydrate-rich structures such as mucins or glycosaminoglycans can adsorb onto charged nanoplastics, influencing corona composition, colloidal stability, and diffusion in mucus or extracellular matrix [1, 10]. Explicit simulation of these systems would require specialized force fields and longer timescales, but the conceptual framework is similar: MNPs reshape local conformational ensembles via non-specific hydrophobic and electrostatic interactions [5, 12].

4 Molecular dynamics methodology for nanoplastic-biomolecule systems

4.1 General MD framework

Classical MD integrates Newton's equations of motion for all atoms in a system, given a potential energy function (force field) that encodes bonded and non-bonded interactions [5, 6, 7]. For a system of N atoms with coordinates \mathbf{r}_i and masses m_i , the equations of motion are

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i(\mathbf{r}_1, \dots, \mathbf{r}_N) = -\nabla_{\mathbf{r}_i} U(\mathbf{r}_1, \dots, \mathbf{r}_N),$$

where U is the potential energy and \mathbf{F}_i the force acting on atom i [5, 6]. In practice, these equations are integrated numerically with algorithms such as the velocity Verlet scheme with a timestep Δt (typically 1-2 fs for all-atom biomolecular simulations) [5, 6].

For nanoplastic-biomolecule systems, typical choices include all-atom or united-atom force fields for proteins and lipids (e.g. AMBER, CHARMM, OPLS, GROMOS)

combined with polymer models for commodity plastics parameterized to reproduce structural and thermodynamic properties [7, 3, 4]. Explicit solvent (e.g. SPC/E or TIP3P water with ions) is usually employed to capture hydration and screening effects [5, 7, 4].

Periodic boundary conditions and particle-mesh Ewald (PME) electrostatics are standard for treating long-range interactions [5, 6]. Temperature and pressure control (e.g. Nosé-Hoover thermostat, Langevin thermostat, Parrinello-Rahman or Langevin piston barostat) maintain near-physiological conditions [5, 7].

The direct output of an MD simulation consists of the time-dependent positions and velocities of every atom in the system, collectively referred to as the trajectory. This trajectory is typically stored at regular intervals (e.g. every picosecond) and serves as the primary raw data from which all physical observables—such as RMSD, secondary structure content, diffusion coefficients, and interaction energies—are computed during post-processing. The trajectory thus provides a complete microscopic record of the system's evolution, enabling detailed analysis of structural transformations, binding events, and dynamic fluctuations that occur during the simulation.

Current all-atom MD simulations of nanoplastic–biomolecule systems are typically limited to timescales of tens to hundreds of nanoseconds and system sizes of hundreds of thousands to a few million atoms. These constraints arise from the computational cost of evaluating forces at each timestep and integrating equations of motion for all particles. Nevertheless, even within these limits, MD simulations can capture critical molecular events—such as initial protein adsorption, partial helix unfolding, nanoplastic insertion into lipid bilayers, and the onset of membrane remodeling—that provide valuable mechanistic insight into how nanoplastics perturb the structure and function of biological molecules.

4.2 Representing nanoplastics

Nanoplastics can be represented on a spectrum from coarse-grained beads to detailed atomistic models of polymer chains [7, 3, 4]. For PE and PS, united-atom descriptions have been successfully used to simulate plastic nanoparticles and chains in aqueous and lipid environments [3, 4]. Two common representations are:

- **Globular nanoparticles:** approximately spherical aggregates of multiple entangled polymer chains at a given diameter (e.g. 5 nm), representing a particulate nanoplastic [3].
- **Disentangled chains:** multiple linear polymer chains dispersed or partially aggregated within a medium, representing nanoplastics that have penetrated and reconfigured within a biomolecular assembly [4].

The choice depends on the biological question (e.g. initial adsorption vs. long-term incorporation and dissolution).

4.3 System construction: proteins and membranes

For protein-nanoplastic simulations, a typical protocol is [5, 7]:

1. Choose a protein of interest (e.g. small α -helical peptide, enzyme with known plastic binding, membrane protein).
2. Obtain or build an equilibrated structure in water (and membrane if applicable).
3. Place a nanoplastic particle or polymer chains at a defined distance and orientation relative to the protein, with sufficient solvent and ions to reproduce the desired ionic strength.
4. Perform energy minimization, gradual heating, and equilibration with positional restraints before production runs.

For membrane systems, Hollóczy and Gehrke constructed large POPC bilayers (~ 2300 lipids) with a PE nanoparticle embedded as a transmembrane object and conducted extensive equilibration and 200 ns production runs, complemented by systems with disentangled chains between lipid leaflets and appropriate control systems [4]. Similar strategies can be applied to mixed-lipid bilayers, cholesterol-rich membranes, or organelle-like compositions [11].

4.4 Sampling considerations and enhanced methods

Timescales for protein adsorption, partial unfolding, and membrane remodeling can exceed hundreds of nanoseconds [3, 4]. Where necessary, enhanced sampling methods (e.g. metadynamics, umbrella sampling, replica-exchange MD) can accelerate exploration of rare events such as insertion, desorption, or large conformational transitions [12]. Reaction coordinates might include:

- Distance between protein center of mass and nanoplastic.
- Polymer insertion depth into a membrane (measured along the bilayer normal).
- Backbone RMSD and fraction of native contacts Q for specific secondary-structure elements.

The potential of mean force (PMF) $W(z)$ for nanoplastic insertion along coordinate z (e.g. depth in a bilayer) can be obtained from umbrella sampling as [12]

$$W(z) = -k_B T \ln P(z) + C,$$

where $P(z)$ is the unbiased probability distribution and C an arbitrary constant. The PMF quantifies the free-energy cost or gain associated with moving the particle along z .

4.5 MD simulation workflow

For clarity, Figure 1 shows a simplified flow diagram of a typical MD simulation workflow as used for nanoplastic-biomolecule systems. This algorithmic view emphasizes that nanoplastic-specific models enter primarily in the system-definition stage (structures, force field) and in the analysis stage (choice of observables directly linked to toxicological endpoints [5, 7, 1]).

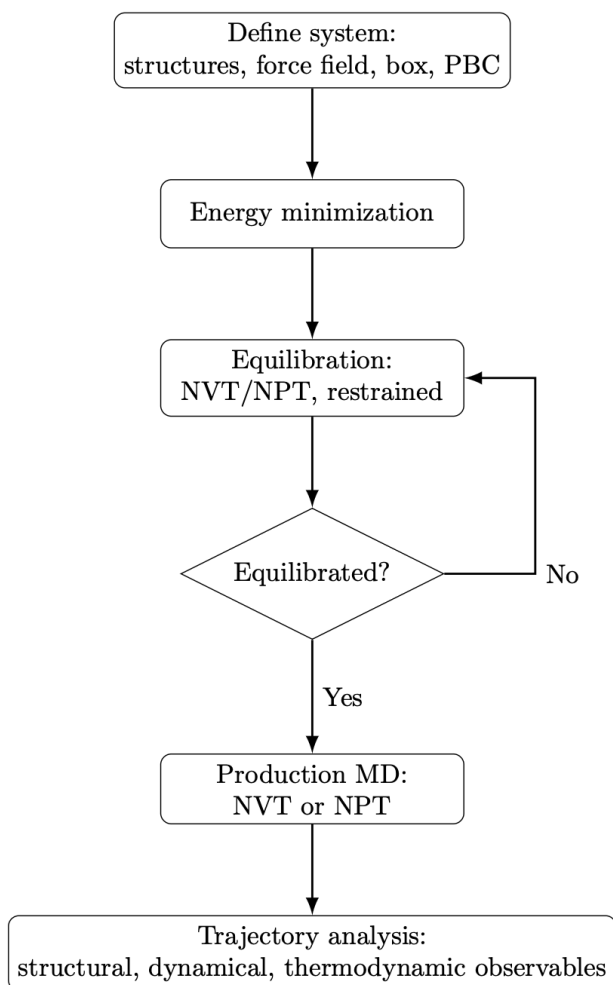


Fig. 1: Simplified flow diagram of a molecular dynamics simulation workflow for nanoplastic-biomolecule systems [5, 6].

5 Example simulations

5.1 Nanoplastic-model protein adsorption

5.1.1 System design

A minimal, yet informative setup includes:

- **Protein:** a small α -helical peptide (e.g. 10-20 residues) or a β -hairpin motif such as a tryptophan zipper, previously used to study nanoplastic-protein interactions [3, 9].
- **Nanoplastic:** a ~ 5 nm PS or nylon-6,6 nanoparticle modeled as entangled chains; optionally variants with different surface chemistries (neutral, carboxylated, aminated).
- **Solvent:** explicit water with physiological ionic strength (e.g. 0.15 mole per liter NaCl).

Multiple initial configurations (e.g. different orientations and distances) should be simulated to assess variability and convergence [5, 6].

5.1.2 Observables and mathematical definitions

From these trajectories, one can compute the following characteristics.

(1) Secondary structure content

The time-dependent fraction $f_\alpha(t)$ of residues in α -helical conformation can be monitored using DSSP or similar algorithms [9]. An overall average fraction

$$\bar{f}_\alpha = \frac{1}{T} \int_0^T f_\alpha(t) dt$$

allows comparison between the free peptide and the peptide adsorbed on the nanoplastic surface. Analogous expressions can be defined for β -sheet or coil content [9].

(2) Backbone RMSD and RMSF

The root-mean-square deviation (RMSD) of backbone atoms from a reference structure (e.g. initial or average structure) is defined as [5, 7]

$$\text{RMSD}(t) = \sqrt{\frac{1}{N_{\text{bb}}} \sum_{i=1}^{N_{\text{bb}}} \|\mathbf{r}_i(t) - \mathbf{r}_i^{\text{ref}}\|^2},$$

where N_{bb} is the number of backbone atoms, $\mathbf{r}_i(t)$ their positions at time t , and $\mathbf{r}_i^{\text{ref}}$ the reference positions. Residue-wise flexibility can be captured by the root-mean-square fluctuation (RMSF)

$$\text{RMSF}_i = \sqrt{\langle \|\mathbf{r}_i(t) - \langle \mathbf{r}_i \rangle\|^2 \rangle_t},$$

where $\langle \mathbf{r}_i \rangle$ is the time-averaged position of atom i [6].

(3) Protein-plastic contact number

A simple measure of contact between protein and nanoplastic is

$$N_c(t) = \sum_{i \in \text{prot}} \sum_{j \in \text{plast}} H(r_c - \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\|),$$

where H is the Heaviside step function, r_c a cutoff (e.g. 0.4 nm), and the sums run over protein and plastic atoms, respectively [5]. Averaging $N_c(t)$ over time and decomposing by residue type provides insight into binding interfaces.

(4) Interaction energies

The non-bonded interaction energy $E_{\text{int}}(t)$ between protein and nanoplastic can be decomposed as [7]

$$E_{\text{int}}(t) = E_{\text{Coul}}(t) + E_{\text{LJ}}(t),$$

where E_{Coul} and E_{LJ} are Coulombic and Lennard-Jones contributions. These can be further decomposed by residue or side-chain class to identify hydrophobic or electrostatic “hotspots” of binding.

5.2 Nanoplastic-lipid bilayer simulations

5.2.1 System design

Following Hollóczy and Gehrke, one can construct [4, 11]:

- A POPC or mixed-lipid bilayer in saline solution.
- A globular PE nanoparticle (5 nm diameter) embedded as a transmembrane object.
- A companion system where the same total number of monomers are arranged as disentangled chains between the leaflets.
- A control bilayer without plastic.

These systems enable direct comparison of structural and dynamical metrics across conditions.

5.2.2 Observables and mathematical definitions

Key membrane observables include the following characteristics.

(1) Bilayer thickness and area per lipid

As defined above, the bilayer thickness is

$$d_{\text{bilayer}} = \left\langle z_{\text{P}}^{(\text{upper})} - z_{\text{P}}^{(\text{lower})} \right\rangle,$$

and the area per lipid A_{lipid} in a rectangular box of dimensions $L_x \times L_y \times L_z$ is [11]

$$A_{\text{lipid}} = \frac{L_x L_y}{N_{\text{lipid,leaflet}}},$$

where $N_{\text{lipid,leaflet}}$ is the number of lipids per leaflet. Nanoplastic incorporation tends to increase d_{bilayer} and decrease A_{lipid} compared to a neat bilayer [4].

(2) Tail order parameters

For each C-H bond along the lipid tails, the deuterium order parameter

$$S_{\text{CD}} = \frac{1}{2} \langle 3 \cos^2 \theta - 1 \rangle$$

quantifies chain ordering, where θ is the angle between the bond vector and bilayer normal [11]. A shift toward more positive S_{CD} indicates greater alignment and chain extension. Nanoplastic-induced changes in S_{CD} directly reflect modifications in membrane mechanical properties [4].

(3) Lateral diffusion coefficient

The lateral diffusion coefficient D of lipid molecules in the bilayer plane can be extracted from the two-dimensional mean-square displacement (MSD) [11, 6]

$$\text{MSD}(t) = \left\langle \left\| \mathbf{r}_{\parallel}(t_0 + t) - \mathbf{r}_{\parallel}(t_0) \right\|^2 \right\rangle_{t_0},$$

where \mathbf{r}_{\parallel} is the 2D position (projection onto xy plane). For sufficiently long times in the diffusive regime,

$$\text{MSD}(t) \approx 4Dt,$$

so that D can be estimated from the slope of $\text{MSD}(t)$ vs. t . Hollóczy and Gehrke observed an increase in D in the presence of PE nanoplastics [4].

(4) Surface coverage and interfacial areas

Using Voronoi tessellation-based domain analysis, one can compute the total nanoplastic surface area A_{tot} and its partition into interfaces with lipids $A_{\text{plast-lip}}$, water $A_{\text{plast-wat}}$, and other plastic atoms [3, 4]. A simple measure of surface coverage by lipids is

$$\theta_{\text{lip}} = \frac{A_{\text{plast-lip}}}{A_{\text{tot}}}.$$

An increase in θ_{lip} over time reflects progressive embedding of nanoplastic material into the bilayer [4].

6 Physical quantities from MD and their interpretation

Table 1 summarizes representative quantities obtainable from MD trajectories in this context, along with their physical meaning and relevance.

In the context of proteins and peptides, the key structural observables extracted from MD trajectories - such as the time-dependent secondary structure content, Ramachandran angle distributions, root-mean-square deviation (RMSD), and root-mean-square fluctuation (RMSF) - are directly linked to how specific segments of the polypeptide chain become ordered, loosened, or partially unfolded in the presence of nanoplastics [9, 10]. First, the evolution of the secondary structure content (e.g. the fraction of residues in α -helical or β -sheet conformation) quantifies the stability of individual motifs under different environmental conditions [9]. If a peptide or protein maintains a high and nearly constant average helical fraction \bar{f}_{α} in aqueous solution, while \bar{f}_{α} decreases significantly in the presence of nanoplastics, this indicates that the particles promote local helix unfolding. In structural terms, adsorption of nanoplastics to hydrophobic faces of a helix may stabilize alternative side-chain orientations and backbone hydrogen-bond patterns, thereby shifting the equilibrium toward more disordered or coil-like states [3, 9]. Conversely, an increase in the fraction of coil structure in specific sequence segments signals that these regions lose their native secondary structure and become more flexible and extended [9, 10].

Second, Ramachandran plots and the associated conformational free energy surface $F(\phi, \psi)$ provide a microscopic view of backbone conformational changes [9, 18]. Shifts of the dominant populations in (ϕ, ψ) space away from regions typical for α -helical conformations (around $\phi \approx -60^\circ$, $\psi \approx -45^\circ$) toward regions associated with β -structures or disordered states indicate that the peptide backbone rotates around the C_{α} -N and C_{α} -C bonds and adopts alternative conformations [9]. In the presence of nanoplastics, such shifts can arise because interactions with the plastic surface stabilize particular side-chain orientations or backbone hydrogen-bonding patterns, thereby altering the location of the free-energy minima in (ϕ, ψ) space and moving the system toward misfolded or partially unfolded conformations [3, 4].

Global measures such as RMSD with respect to a reference structure report how far the entire protein deviates from its initial or experimentally determined conforma-

Table 1: Representative MD observables for nanoplastic-biomolecule systems.

Quantity	Definition / MD evaluation	What it reveals	Relevance in nanoplastic context
Secondary structure content	Fraction of residues in α , β , 3_{10} , π , coil via DSSP or similar [9]	Stability of helices, sheets, turns	Adsorption-induced helix or sheet destabilization linked to loss of protein function or misfolding [8, 3].
Ramachandran distributions	2D histograms of ϕ , ψ dihedrals and free energy $F(\phi, \psi)$ [9]	Allowed vs. disfavored backbone conformations	Shifts in backbone ensembles near plastic surface vs. bulk indicate structural perturbation [3].
RMSD / RMSF	Deviation from reference structure and residue-wise fluctuations [5, 6]	Global and local flexibility	Increased fluctuations near binding sites may correlate with activity loss or gain.
Protein-plastic contact number	Number of atoms or residues within a cutoff distance from plastic [5]	Binding interface size and composition	Identifies hydrophobic patches driving adsorption and potential hotspots for structural change.
Bilayer thickness	Average leaflet separation (P-P distance) [11]	Membrane stretching or thinning	Nanoplastic-induced thickening observed for PE in POPC, with implications for membrane protein function [4].
Area per lipid	Box area divided by number of lipids per leaflet [11]	Lateral packing density	Decrease upon plastic incorporation indicates denser packing and possible permeability changes [4].
Tail order parameters	$S_{CD} = \frac{1}{2} \langle 3 \cos^2 \theta - 1 \rangle$ per tail segment [11]	Chain ordering and fluidity	Changes reflect altered mechanical properties and diffusion barriers [4].
Lateral diffusion coefficient	Slope of MSD vs. time in 2D, $MSD(t) \approx 4Dt$ [6, 11]	Membrane fluidity	Increased diffusion in presence of plastic suggests modified viscosity and domain dynamics [4].
Interaction energies	Non-bonded between plastic and biomolecules [7]	energies and binding forces	Decomposition by residue or lipid species indicates specificity of nanoplastic-biomolecule contacts.
Surface coverage and interfacial areas	Voronoi tessellation on molecular surfaces [3, 4]	How much plastic surface contacts lipids vs. water	Demonstrates dissolution of nanoplastics into membranes and entanglement with lipids [3, 4].

tion [5, 6, 7]. A gradual increase of RMSD that reaches a plateau at a moderate value typically reflects relaxation toward a thermally equilibrated conformation in the given environment, whereas a pronounced growth of RMSD and large fluctuations suggest transitions to alternative, structurally distinct states [5]. In the context of nanoplastic exposure, a substantial and persistent increase of RMSD for a functional helix or domain relative to the control simulation in water would be indicative of destabilization of that structural element, potentially compromising its biological role [9, 8]. Complementary, residue-resolved RMSF quantifies local flexibility along the sequence, allowing

one to identify regions that become either more mobile or more rigid [6]. An increase in RMSF for residues in contact with the nanoplastic surface points to enhanced local mobility, suggesting frequent transitions between folded and partially unfolded substates, whereas a decrease in RMSF may signal that the protein segment becomes “locked” against the plastic surface in a single, adsorbed conformation [3, 4]. Together, these observables translate changes in the microscopic conformational ensemble into quantitative descriptors of structural destabilization, local unfolding, or rigidification triggered by nanoplastics.

For lipid bilayers, the set of observables described above - including the bilayer thickness d_{bilayer} , the area per lipid A_{lipid} , the deuterium order parameter S_{CD} , and the lateral diffusion coefficient D - captures how nanoplastics alter lipid packing, chain extension, membrane fluidity, and mechanical properties [11]. In particular, thickening of the bilayer accompanied by a decrease in A_{lipid} corresponds to a state in which the hydrocarbon chains are more extended and lipids are packed more densely in the plane of the membrane [11]. This situation has been observed in simulations where polyethylene-like nanoplastics dissolve into the hydrophobic core and disintegrate into individual chains, effectively acting as “spacers” that push lipid tails into more elongated conformations and increase the distance between opposing leaflets [4]. From the point of view of membrane structure, such changes can modulate the hydrophobic mismatch between lipids and embedded membrane proteins, shift helix tilt angles, and modify the energetic cost of pore or defect formation [11, 10].

The deuterium order parameter S_{CD} provides a more local, segment-resolved measure of chain extension and orientational order [11]. More positive values of S_{CD} indicate that the corresponding C-H (or C-D) bonds are preferentially aligned with the bilayer normal, which is characteristic of straighter, more ordered hydrocarbon chains; values closer to zero indicate increased conformational disorder and a higher population of gauche states along the chain. An overall increase in S_{CD} in the presence of nanoplastics therefore reflects a shift toward more ordered and extended lipid tails, consistent with membrane thickening and a more “gel-like” state of the bilayer core [4, 11]. Conversely, a decrease in S_{CD} would imply increased chain bending and a more fluid, disordered membrane. By quantifying these changes, MD simulations provide a direct microscopic link between nanoplastic-induced alterations at the level of individual lipid chains and macroscopic membrane properties such as rigidity, permeability, and domain formation [11, 1].

The lateral diffusion coefficient D characterizes the translational mobility of lipids in the membrane plane and is a sensitive indicator of membrane fluidity [6, 11]. If incorporation of nanoplastic chains into the bilayer core increases D relative to a control system, this suggests that local packing interactions between neighboring lipids are weakened or reorganized, facilitating lateral rearrangements and potentially leading to the formation of heterogeneous microdomains with distinct dynamical properties [4, 11]. Conversely, a decrease in D indicates that the membrane becomes more viscous and laterally constrained, which could hinder the diffusion and oligomerization of membrane proteins, and thereby affect signaling processes that rely on lateral organization [11, 10]. In combination with d_{bilayer} , A_{lipid} , and S_{CD} , the diffusion coefficient thus completes a coherent picture of how nanoplastics reshape the membrane landscape at both structural and dynamical levels.

Finally, interaction energies between nanoplastics and biomolecules, together with the number and nature of contacts at the interface, illuminate how specific molecular interactions drive the structural changes discussed above [7, 5]. A strongly stabilizing interaction energy between nanoplastics and proteins, combined with a high number of hydrophobic contacts, indicates the formation of a persistent adsorption interface, often involving exposure of otherwise buried hydrophobic side chains and “flattening” of helices or loops against the plastic surface [3, 4]. In membrane systems, favorable interaction energies between nanoplastic chains and lipid tails, accompanied by extensive polymer-lipid contacts, signal dissolution of the nanoplastics into the hydrophobic core and entanglement with lipid acyl chains [4]. These energetic and contact-based descriptors therefore bridge geometric observables (secondary structure content, bilayer thickness, order parameters) with a mechanistic understanding of the driving forces underlying nanoplastic-induced structural perturbations [7, 5].

Taken together, the physical quantities computed from MD trajectories - geometric, energetic, and dynamical - provide a consistent and quantitative language for describing how nanoplastics alter the conformational ensembles of proteins and peptides, reshape lipid bilayers, and modify the interactions that govern biomolecular structure and function [9, 11, 4, 1]. They thus form a crucial link between atomistic mechanisms and experimentally observed toxicological endpoints, such as loss of enzymatic activity, membrane leakage, oxidative stress, and inflammation [2, 1].

7 Nanoplastics, oxidative stress, and adverse outcome pathways

Hu and Palić systematically reviewed micro- and nanoplastic-induced oxidative and inflammatory events and mapped reported endpoints onto existing AOPs [1]. Across diverse organisms and models, ROS generation emerged as a common MIE, followed by oxidative stress, activation of MAPK and Nrf2 pathways, lipid peroxidation, DNA damage, mitochondrial dysfunction, lysosomal disruption, apoptosis, and inflammation [1].

From an MD perspective, several of these events relate directly or indirectly to structural and dynamical changes in proteins and membranes:

- **Membrane perturbation:** altered bilayer properties can affect mitochondrial inner membrane function, promoting ROS production and initiating ROS-induced ROS-release cascades [1, 11].
- **Lysosomal integrity:** nanoplastic accumulation and membrane disruption can compromise lysosomal membranes, releasing hydrolytic enzymes and further exacerbating oxidative damage [1].
- **Enzyme inactivation:** structural changes in antioxidant enzymes or signaling proteins may impair detoxification and repair pathways, amplifying oxidative stress [9, 10].

AOP-based schematics for micro- and nanoplastics indicate that from an ecological perspective, ROS-driven cascades ultimately lead to growth inhibition, behavioral changes, and reproductive impairment, while from a human-health perspective, apop-

tosis and chronic inflammation emerge as probable outcomes, although detailed AOPs remain incomplete [1].

8 Perspectives, challenges, and future directions

Despite rapid progress, significant gaps remain in our understanding of how micro- and nanoplastics perturb biomolecular structure and function:

- Experimental toxicology often uses high doses and simplified exposure scenarios that may not reflect environmental conditions [2, 1].
- The diversity of plastic types, sizes, shapes, and surface chemistries complicates generalization from model particles (often pristine PS) to environmental nanoplastic mixtures [2].
- Environmentally aged nanoplastics, with modified surfaces and adsorbed contaminants, likely interact differently with biomolecules than pristine laboratory particles [2, 1].
- Comprehensive MD studies of realistic systems (e.g. mixed coronas on heterogeneous nanoplastics interacting with multi-domain proteins or complex membranes) are computationally demanding [3, 4, 5].

Nonetheless, MD can play a central role in bridging molecular-level mechanisms with higher-level toxicological outcomes by:

- Systematically exploring structure-activity relationships for different polymer chemistries, sizes, and functionalizations using standardized sets of observables such as those listed in Table 1 [5, 6].
- Quantifying how specific nanoplastics perturb known protein motifs (e.g. α -helices, β -hairpins, catalytic loops) or membrane properties and mapping these perturbations to MIEs and KEs in AOP frameworks [1, 9, 11].
- Providing mechanistic hypotheses to guide targeted experiments and refine AOPs, including predictions of size- and surface-charge dependence of ROS generation, membrane damage, and enzyme inactivation [1, 12].

Future work should prioritize integration of MD with high-resolution biophysical experiments (e.g. NMR, cryo-EM, advanced fluorescence), realistic exposure scenarios, and multi-scale modeling that connects nanoscale events to cellular and organismal responses [9, 10, 4, 1]. As analytical methods for detecting and characterizing environmental nanoplastics continue to improve, it will become increasingly feasible to simulate realistic nanoplastic-biomolecule systems with parameters directly informed by field data [2].

Acknowledgement

This work was supported by the AI2SEP project (No. 2023-1-PL01-KA220-HED-000166765). We gratefully acknowledge the project's support and funding, which enabled this research. The authors declare that they have no competing interests relevant to the content of this chapter.

References

1. Hu, M., & Palić, D. (2020). Micro- and nano-plastics activation of oxidative and inflammatory adverse outcome pathways. *Redox Biology*, 37, Article 101620. <https://doi.org/10.1016/j.redox.2020.101620>
2. Shen, M., Zhang, Y., Zhu, Y., Song, B., Zeng, G., Hu, D., Wen, X., & Ren, X. (2019). Recent advances in toxicological research of nanoplastics in the environment: A review. *Environmental Pollution*, 252, 511–521. <https://doi.org/10.1016/j.envpol.2019.05.115>
3. Hollóczki, O., & Gehrke, S. (2019). Nanoplastics can change the secondary structure of proteins. *Scientific Reports*, 9(1), Article 16013. <https://doi.org/10.1038/s41598-019-52495-w>
4. Hollóczki, O., & Gehrke, S. (2020). Can nanoplastics alter cell membranes? *ChemPhysChem*, 21(1), 9–12. <https://doi.org/10.1002/cphc.201900881>
5. Frenkel, D., & Smit, B. (2002). *Understanding molecular simulation: From algorithms to applications* (2nd ed.). Academic Press.
6. Allen, M. P., & Tildesley, D. J. (2017). *Computer simulation of liquids* (2nd ed.). Oxford University Press. <https://doi.org/10.1093/oso/9780198803195.001.0001>
7. Leach, A. R. (2001). *Molecular modelling: Principles and applications* (2nd ed.). Pearson.
8. Weaver, T. M. (2000). The π -helix translates structure into function. *Protein Science*, 9(1), 201–206. <https://doi.org/10.1110/ps.9.1.201>
9. Branden, C., & Tooze, J. (1999). *Introduction to protein structure* (2nd ed.). Garland Science.
10. Lodish, H., Berk, A., Kaiser, C. A., Krieger, M., Bretscher, A., Ploegh, H., Amon, A., & Martin, K. C. (2016). *Molecular cell biology* (8th ed.). W. H. Freeman.
11. Mouritsen, O. G. (2005). *Life - As a matter of fat: The emerging science of lipidomics*. Springer. <https://doi.org/10.1007/b138666>
12. Chipot, C., & Pohorille, A. (Eds.). (2007). *Free energy calculations: Theory and applications in chemistry and biology*. Springer. <https://doi.org/10.1007/978-3-540-38448-9>
13. Du, T., Yu, X., Shao, S., Li, T., Xu, S., & Wu, L. (2023). Aging of nanoplastics significantly affects protein corona composition thus enhancing macrophage uptake. *Environmental Science & Technology*, 57(8), 3206–3217. <https://doi.org/10.1021/acs.est.2c07447>
14. Cai, R., Baimanov, D., Yuan, H., Xie, H., Yu, S., Zhang, Z., Yang, J., Zhao, F., You, Y., Guan, Y., Zheng, P., Xu, M., Qi, M., Zhang, Z., Zhong, S., Li, Y. F., & Wang, L. (2024). Protein corona-directed cellular recognition and uptake of polyethylene nanoplastics by macrophages. *Environmental Science & Technology*, 58(32), 14158–14168. <https://doi.org/10.1021/acs.est.4c02251>
15. Xiao, S., Wang, J., Digiacomo, L., Amici, A., De Lorenzi, V., Pugliese, L. A., Cardarelli, F., Cerrato, A., Laganà, A., Cui, L., Papi, M., Caracciolo, G., Marchini, C., & Pozzi, D. (2024). Protein corona alleviates adverse biological effects of nanoplastics in breast cancer cells. *Journal of Nanobiotechnology*, 22(1), Article 373. <https://doi.org/10.1186/s12951-024-02641-4>
16. Tang, Y., Wang, R., Xu, Z., Wang, J., Zhou, D., Zhang, S., Wei, D., Ren, J., & Liu, S. (2024). Contribution of cancer-specific protein coronas to the pro-tumor effects of nanoplastics through enhanced cellular interactions. *ACS Applied Materials & Interfaces*, 16(49), 65540–65552. <https://doi.org/10.1021/acsami.4c14824>
17. Simpson, K., Martin, L., O'Leary, S. L., Watt, J., Moon, T., Luo, T., & Xu, W. (2025). Environmental protein corona on nanoplastics altered the responses of skin keratinocytes and fibroblast cells to the particles. *Journal of Hazardous Materials*, 494, Article 138722. <https://doi.org/10.1016/j.jhazmat.2025.138722>

18. Pauling, L., & Corey, B. B. (1951). Atomic coordinates and structure factors for two helical configurations of polypeptide chains. *Proceedings of the National Academy of Sciences of the United States of America*, 37(5), 235–240. <https://doi.org/10.1073/pnas.37.5.235>

About authors

Jakub Kraus is a Master's student in Micro- and Nanotechnology with a strong engineering background in Applied Computer Science. His primary research focuses on computational biophysics, specifically the modeling of ion channel gating dynamics. By leveraging his expertise in software engineering and systems programming, he develops and implements machine learning algorithms to analyze complex nanoscale transport phenomena.

Piotr Mika is a Micro- and Nanotechnology Master's student, building on an engineering education in Applied Computer Science. His research interests lie in computational modeling and simulation of complex physical phenomena at the nanoscale. With experience in high-performance computing and modern machine learning methods, he applies advanced numerical and data-driven approaches to problems in biophysics.

Mateusz Wiśniewski is a PhD student at University of Silesia in Katowice, affiliated with the Institute of Physics at the Faculty of Science and Technology. His research interests include stochastic processes and Brownian motion. In particular, he focuses on the theoretical modeling of non-Markovian dynamics of microscopic systems.

Paulina Trybek is an assistant professor at the University of Silesia in Katowice, affiliated with the Institute of Physics, where she specializes in the analysis of biomedical time series. She is actively involved in numerous student projects, supporting the development of data analysis competencies. She is also the coordinator of the project "Developing Talents in Artificial Intelligence to Solve Disruptive Environmental Problems."

Zbigniew Dendzik is an associate professor at the University of Silesia in Katowice, affiliated with the Institute of Physics. His research interests include properties of two-dimensional phases of mesogens, prenematic fluctuations in nanoparticles-doped liquid crystals and mechanical properties of biomembranes, studied with molecular dynamics computer simulation methods.



University of Maribor Press

Modeling Heat Transfer in an Urban Settlement with 3D Cellular Automata and Artificial Intelligence

Robert Leskovar

Abstract

This chapter presents a three-dimensional (3D) cellular automata (CA) model simulating urban heat transfer in an artificial $50 \times 50 \times 20$ m city block at moderate latitude ($\varphi = 46.24^\circ$ N). Each cell is assigned one of five material classes — outdoor air, asphalt, park/grass, structural concrete, and interior air — and advances temperature states through eight physics layers: conductive diffusion, solar heating, nocturnal radiative cooling, free-atmosphere relaxation, asphalt plume convection, surface Newton cooling, interior ventilation, and boundary conditions. A distinguishing aspect of this work is its development workflow: rather than conventional coding, human-specified intent and domain constraints in natural language guided Claude AI (Anthropic) [27] in generating, debugging, and iteratively refining the Python implementation. The human supplied physical intuition, validated outputs, and corrected physically implausible results through a joint debugging process. Preliminary simulations over a 30-day May period demonstrate realistic diurnal temperature cycles, street-surface temperatures 4–6 °C above park surfaces at solar noon, and an asphalt plume decaying exponentially up to 10 m above ground. These results confirm the CA paradigm’s suitability for urban heat studies [5, 2] and illustrate the potential of large language model (LLM)-assisted scientific prototyping.

Keywords: cellular automata; urban heat transfer; human-machine interaction; AI-assisted coding; urban microclimate

1 Introduction

Urban environments modify the local thermal climate through a combination of reduced vegetation cover, high-absorptivity paved surfaces, anthropogenic heat emissions, and altered sky-view factors [14, 15]. The resulting urban heat island (UHI) effect elevates near-surface temperatures by 2–10 °C relative to rural surroundings, with the strongest signal occurring at night when building materials release the solar energy accumulated during the day [16]. Quantifying and predicting these effects at the scale of individual city blocks is essential for evidence-based urban design, and calls for simulation tools that balance physical fidelity with computational tractability.

Cellular automata (CA) offer a compelling middle ground between computationally expensive solvers such as computational fluid dynamics (CFD) solvers and simplified empirical models [3, 5]. The applications of CAs span from pure physical systems to engineering and also to social sciences (i.e. [28]). Originally formalised by von Neumann and popularised as universal models of complexity by Wolfram [1], CA propagate discrete local states according to neighbourhood rules at each timestep. For heat conduction, Chopard and Droz [2] demonstrated that a lattice-gas CA reproduces the thermo-hydrodynamic equations accurately, and Lowekamp [4] extended this to three-dimensional parallel architectures. Subsequent work has applied CA to urban growth simulation [24], fire spread [6], and tree-network heat conduction [25].

A separate, rapidly evolving development in scientific computing is the use of large language models (LLMs) as interactive coding assistants. Rather than writing numerical simulation code from scratch, researchers can describe their intent in natural language and have an LLM generate candidate implementations, which the researcher then evaluates, critiques, and refines in dialogue. This human–machine co-authoring paradigm has been explored in urban planning and scenario analysis contexts [26], but its application to low-level physics-based simulation coding remains largely undocumented in the peer-reviewed literature.

This chapter makes two contributions. First, it describes the design and implementation of a 3D CA heat-transfer model for a synthetic city block featuring two intersecting roads, three buildings of differing heights, and surrounding park areas, parameterised for the Kranj, Slovenia May climate. Second, it documents the iterative dialogue between the human researcher and the Claude AI assistant ([27]) that produced the simulation code, highlighting how physical errors were detected, diagnosed, and corrected through this interaction. Preliminary visualisation results are presented to demonstrate the plausibility of the model outputs.

2 Methodology

2.1 Problem statement and goals

Rapid urbanization replaces natural permeable surfaces with heat-absorbing materials such as asphalt and concrete, fundamentally altering the local surface energy balance

and giving rise to the urban heat island (UHI) effect—a well-documented phenomenon in which cities record persistently higher air and surface temperatures than surrounding rural areas [33, 35]. Despite decades of observation, the three-dimensional thermal structure of urban blocks at the microscale (tens of metres) remains difficult to characterise, because diurnal cycles, material-class heterogeneity, vertical boundary-layer development, and cloud-modulated radiative forcing interact nonlinearly [31, 34]. Coupled mesoscale–microscale approaches, such as the Weather Research and Forecasting (WRF) model paired with urban canopy models (UCMs), have improved our understanding of canopy-layer heat retention and nocturnal warm anomalies [34, 32], yet most studies treat the urban volume as a two-dimensional or single-layer system and do not resolve the vertical decay of the thermal plume above high-absorptivity surfaces [33]. Furthermore, cloud-cover variability is frequently ignored or parametrised only at coarse temporal resolution, even though it is the dominant modulator of incoming shortwave radiation and, consequently, of surface temperature extremes [32].

The primary goal in this research is to simulate the three-dimensional temperature cube in a $50 \times 50 \times 20$ m urban block over a 30-day period in May, capturing:

- diurnal temperature cycles for each material class;
- the urban heat island signature (elevated asphalt temperatures);
- the vertical decay of the asphalt boundary-layer thermal plume;
- realistic cloud-modulated solar and radiative forcing.

A 3D cellular automaton with von Neumann neighborhood will be modeled (see Figure 1).

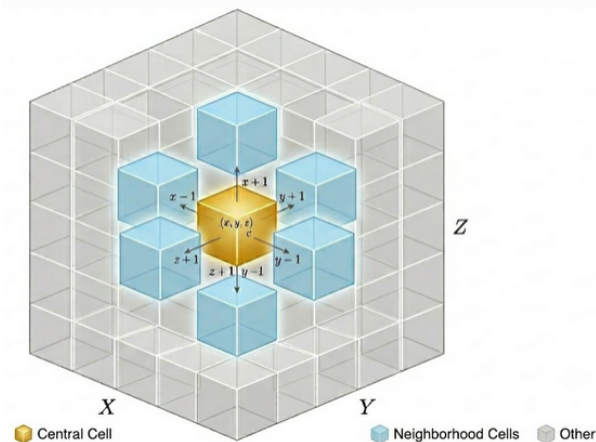


Fig. 1: A 3D cellular automaton with von Neumann neighborhood.

A secondary goal is to explore and document the human–AI interaction pattern that led to a functioning, physically validated simulation through iterative natural-language prompting and code refinement.

2.2 Methods

2.2.1 Simulation approach

The model uses an explicit three-dimensional CA on a $50 \times 50 \times 20$ rectilinear lattice ($\Delta x = 1$ m, $\Delta t = 600$ s). Each cell is assigned a material index; the temperature field $T_{i,j,k}$ is advanced by eight sequentially applied physics layers (see Section 3.2). The implementation is in Python 3 using NumPy for vectorised array operations and Matplotlib for visualisation.

2.2.2 Human–AI co-authoring workflow

The coding process followed an iterative dialogue structure between the human researcher and Claude AI:

1. The human described the desired model elements in natural language (domain size, materials, physics layers, climate inputs).
2. Claude AI generated a candidate Python implementation.
3. The human ran the code, inspected outputs, and identified physical or numerical inconsistencies.
4. The human reported findings back to Claude AI in plain language, which then proposed corrections with explanations.
5. Steps 3–4 were repeated until outputs matched expected physical behaviour.

This workflow produced twelve targeted fixes across two revision cycles, documented in the final code comments as FIX 1 through FIX 12.

2.3 Assumptions and limitations

The model adopts several simplifying assumptions that bound its applicability (see Table 1).

A particular challenge would pose automaton cells located below ground level, as well as the heterogeneous materials present on building surfaces. While these elements would substantially increase both the size and complexity of the model, they would simultaneously enhance its fidelity to the real-world system.

Table 1: Assumptions and limitations.

Assumption	Limitation
One-dimensional diffusivity lookup	Ignores material anisotropy (<5% error for urban materials) [9].
Uniform $\alpha_s = 0.7$	Asphalt ≈ 0.95 , concrete ≈ 0.65 [22]; blended value underestimates street peak.
No moisture / latent heat	Evapotranspiration from park is neglected [14].
No anthropogenic heat	Traffic, HVAC exhaust absent [5].
Linearised longwave radiation	Valid within $\pm 15^\circ\text{C}$ of $T_m = 290\text{ K}$ [9].
Simplified vertical solar shading	Full 3D ray-tracing not implemented; shadow test uses column- maximum height only [18].
ACH = 1.0 uniform	Ignores window-to-wall ratio, HVAC, and occupancy [15].
Free-atmosphere relaxation $\tau = 67\text{ min}$	\approx Mesoscale coupling would be more rigorous; the value prevents multi-day drift within the domain.

3 Research

3.1 City geometry

The $50 \times 50 \times 20$ domain represents a synthetic urban block. Two arterial streets intersect to form a cross-shaped road network: a north–south street, and an east–west street, both at ground level. Three buildings are distributed across the domain. Block A is a six-storey residential building located in the southwest quadrant. Block B is a twelve-storey tower in the northeast quadrant. Block C is an eight-storey mid-rise situated in the southeast quadrant. All three structures rise from ground level. All remaining ground-level cells not occupied by streets or buildings are designated as park surface. Precise definition:

- a N–S arterial street (cells $x = 22\text{--}27$, full y extent, $z = 0$);
- an E–W arterial street (cells $y = 23\text{--}26$, full x extent, $z = 0$);
- **Block A** — 6-storey residential building, SW quadrant ($x=5\text{--}15$, $y=5\text{--}15$, $z=0\text{--}6$);
- **Block B** — 12-storey tower, NE quadrant ($x=30\text{--}45$, $y=10\text{--}25$, $z=0\text{--}12$);
- **Block C** — 8-storey mid-rise, SE quadrant ($x=10\text{--}20$, $y=30\text{--}40$, $z=0\text{--}8$);
- surrounding park (MAT_PARK) at $z = 0$ for all remaining ground cells.

Buildings have a one-cell-thick concrete shell (MAT_BUILDING) enclosing an interior air core (MAT_INTERIOR).

Figure 2 shows the isometric layout of the modelled block.

Listing 1: City geometry generator

```
#
# 3. CITY GEOMETRY
#
```

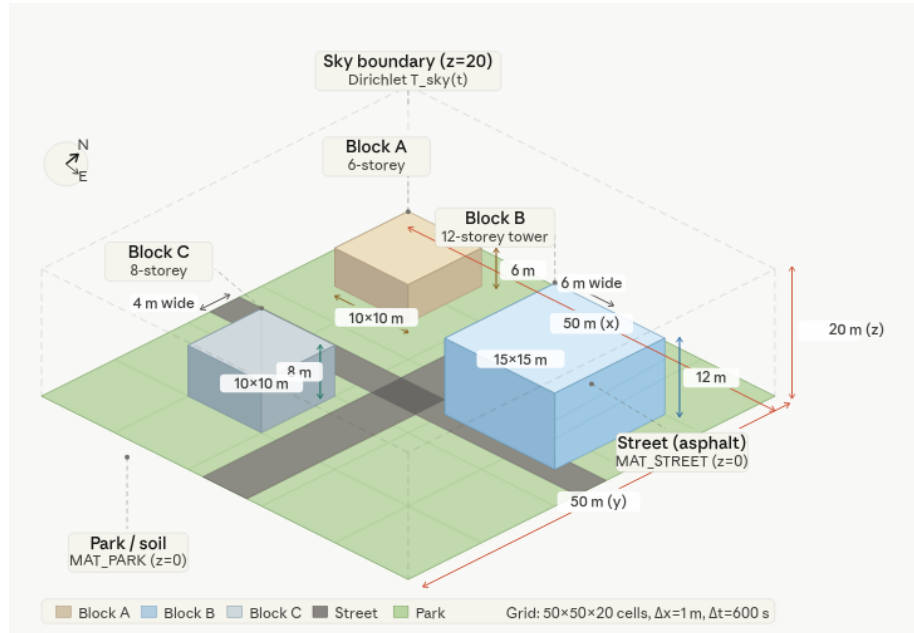


Fig. 2: Isometric view of the $50 \times 50 \times 20$ m CA domain. Three buildings (Blocks A, B, C), N–S and E–W streets, park ground cover, and domain bounding box ($z = 20$ m).

```
def generate_city_map(size):
    """
    z = 0      : ground layer (park, street, or building
                          footprint slab)
    z = 1+    : above-ground (building shell/interior,
                          open air columns)
    Three buildings with hollow interiors (MAT_INTERIOR)
    surrounded by MAT_BUILDING shell, set in a park
    grid crossed by two streets.
    """
    w, l, h = size
    mat = np.full(size, MAT_AIR, dtype=np.int8)

    # Ground: park everywhere, then overwrite with streets
    mat[:, :, 0] = MAT_PARK
    mat[w//2-3 : w//2+3, :, 0] = MAT_STREET # -NS
    mat[:, l//2-2 : l//2+2, 0] = MAT_STREET # -EW

    def add_building(mat, x1, x2, y1, y2, nz):
        """Shell of MAT_BUILDING with MAT_INTERIOR core."""
        mat[x1:x2, y1:y2, 0:nz] = MAT_BUILDING
```

```

    if nz > 2 and (x2-x1) > 2 and (y2-y1) > 2:
        mat[x1+1:x2-1, y1+1:y2-1, 1:nz-1] = MAT_INTERIOR

    add_building(mat, 5, 15, 5, 15, 6) # Block A — 6 storeys
    add_building(mat, 30, 45, 10, 25, 12) # Block B — 12 storeys
    add_building(mat, 10, 20, 30, 40, 8) # Block C — 8 storeys

    return mat

```

3.2 Physics of the CA update

Each 10-minute timestep applies eight physics layers in order. The governing equations and their CA discretisations are described in the following subsections.

3.2.1 Layer 1: Conductive diffusion

The heat equation [10];

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T, \quad (1)$$

discretised with a six-neighbour (von Neumann) pattern in 3D [3]:

$$T_{i,j,k}^{n+1} = T_{i,j,k}^n + \frac{\alpha \Delta t}{\Delta x^2} \left(T_{i\pm 1}^n + T_{j\pm 1}^n + T_{k\pm 1}^n - 6 T_{i,j,k}^n \right). \quad (2)$$

Stability requires the Fourier number $Fo = \alpha \Delta t / \Delta x^2 \leq 1/6$ [11, 23]. All five materials satisfy this constraint at $\Delta t = 600$ s, $\Delta x = 1$ m (the most diffusive material, outdoor air, yields $Fo = 0.012$).

Listing 2: Diffusion kernel (Layer 1)

```

T_pad      = np.pad(T, 1, mode='edge')
laplacian  = (T_pad[2:, 1:-1, 1:-1] +
             T_pad[: -2, 1:-1, 1:-1] +
             T_pad[1:-1, 2:, 1:-1] +
             T_pad[1:-1, : -2, 1:-1] +
             T_pad[1:-1, 1:-1, 2:] +
             T_pad[1:-1, 1:-1, : -2] - 6.0 * T)
alpha_map  = np.vectorize(DIFFUSIVITY.get)
(material_map).astype(np.float64)
new_T     += (alpha_map * DT / DX**2) * laplacian

```

3.2.2 Layer 2: Solar heating

The temperature increment on exposed surface cells is

$$\Delta T_{\text{solar}} = G_{\text{rate}} \cos \theta_z k_t \times (1 + r_{\text{season}} d), \quad (3)$$

where $G_{\text{rate}} = 0.017 \text{ }^\circ\text{C step}^{-1}$ is the corrected solar gain rate (derivation below), $\cos \theta_z$ is the cosine of the solar zenith angle [12], k_t is the sky clearness index, $r_{\text{season}} = 0.015 \text{ day}^{-1}$ is a May seasonal ramp (FIX 11), and d is the day counter (0-based).

The rate G_{rate} is derived from the energy balance on a surface cell assuming the solar flux is absorbed within a 0.1 m skin depth:

$$G_{\text{rate}} = \frac{I_0 \alpha_s f_{\text{slab}} \Delta t}{\rho C_p \Delta x} = \frac{800 \times 0.7 \times 0.1 \times 600}{2024000 \times 1.0} \approx 0.017 \text{ }^\circ\text{C step}^{-1}. \quad (4)$$

An early version of the code (generated in the first dialogue turn) had $G_{\text{rate}} = 0.17$, omitting the slab fraction $f_{\text{slab}} = 0.1$, causing a ten-fold overestimate of solar heating. This was corrected in the first major revision after the human observed unrealistically high surface temperatures.

Listing 3: Solar heating on exposed cells (Layer 2, FIX 1 & 11)

```
SOLAR_GAIN_RATE = 0.017
# deg C / step (FIX 1: was 0.17)
SOLAR_SEASON_RAMP_PER_DAY = 0.015
# fractional daily increase (FIX 11)

if solar > 0:
    season_ramp = 1.0 + SOLAR_SEASON_RAMP_PER_DAY * day_of_sim
    new_T[exposed_mask] += SOLAR_GAIN_RATE * solar
                        * cloud * season_ramp
```

3.2.3 Layer 3: Nocturnal radiative cooling

During night, top-surface cells lose energy via longwave emission, modulated by cloud cover [9]:

$$\Delta T_{\text{rad}} = -R_{\text{rate}} k_t, \quad (5)$$

with $R_{\text{rate}} = 0.010 \text{ }^\circ\text{C step}^{-1}$ (FIX 2, re-balanced against the corrected solar gain).

3.2.4 Layer 4: Free-atmosphere relaxation

Outdoor air cells are continuously relaxed toward the prescribed sky temperature $T_{\text{sky}}(t)$ at each step (FIX 4):

$$T_{\text{air}}^{n+1} = T_{\text{air}}^n + \lambda_{\text{atm}} (T_{\text{sky}} - T_{\text{air}}^n), \quad \lambda_{\text{atm}} = 0.15, \quad (6)$$

giving an e-folding relaxation time of $\Delta t/\lambda_{\text{atm}} = 67$ min. This prevents multi-day accumulation of heat in the outdoor air and keeps the air field nearly independent of adjacent surface temperatures, consistent with the large heat capacity of the free atmosphere.

Listing 4: Free-atmosphere relaxation (Layer 4, FIX 4)

```
ATM_RELAX = 0.15
# e-fold time ~67 min (FIX 4, was absent)

air_cells = iface["is_air"]
new_T[air_cells] += ATM_RELAX *
                    (sky_T - new_T[air_cells])
```

3.2.5 Layer 5: Asphalt boundary-layer plume

Asphalt absorbs significantly more solar energy than park soil, creating a warm plume above the street. Air cells in open columns above asphalt receive a fraction of the asphalt–sky temperature excess, decaying exponentially with height (FIX 9):

$$\Delta T_z = S_{\text{asp}} \cdot e^{-z/H_{\text{scale}}} \cdot (T_{\text{asp}} - T_{\text{sky}}), \quad z = 1, \dots, 10, \quad (7)$$

where $S_{\text{asp}} = 0.25$ is the fractional coupling per step and $H_{\text{scale}} = 4.0/(1 + v \text{ [m/s]} \times 0.3) \approx 3.1$ m at the Kranj May mean wind speed $v = 1.67$ m/s.

Listing 5: Asphalt plume heating above street columns (Layer 5, FIX 9)

```
ASPHALT_PLUME_TOP = 10 # max influence height [m]
ASPHALT_H_SCALE   = 4.0 / (1.0 + WIND_SPEED_MS * 0.3)
                  # ~3.1 m
ASPHALT_STRENGTH  = 0.25 # fractional coupling per step

excess_xy = new_T[:, :, 0] - sky_T
for z in range(1,
               min(ASPHALT_PLUME_TOP + 1,
                   GRID_SIZE[2])):
    mask_z = plume_mask[z]
    delta  = ASPHALT_STRENGTH *
            decay_weight[z] * excess_xy
    new_T[:, :, z][mask_z] += delta[mask_z]
```

3.2.6 Layer 6: Surface–air convection

Newton’s law of cooling is applied symmetrically between solid surfaces and adjacent air cells (FIX 3). The convective coefficient follows the Jurges correlation at 6 km/h wind speed, $h_c \approx 5.6 + 4 \times 1.67 \approx 12.3$ W/(m²K), reduced by a canyon sheltering factor of 0.8 to give $H_{\text{conv}} = 10$ W/(m²K) (FIX 7):

$$\text{CONV_SOLID}(m) = \min\left(1, \frac{H_{\text{conv}} \Delta t}{\rho C_p(m) \Delta x}\right). \quad (8)$$

3.2.7 Layer 7: Interior ventilation

Building interior air exchanges with outdoor air at 1 air change per hour (ACH = 1.0):

$$T_{\text{int}}^{n+1} = (1 - f_{\text{vent}}) T_{\text{int}}^n + f_{\text{vent}} \bar{T}_{\text{outdoor, adj}}, \quad (9)$$

where $f_{\text{vent}} = \text{ACH} \times \Delta t / 3600 \approx 0.167$.

3.2.8 Layer 8: Boundary conditions

The top layer $z = 19$ is held at the prescribed sky temperature (Dirichlet BC):

$$T_{i,j,19}^n = T_{\text{sky}}(t). \quad (10)$$

Park ground cells ($z = 0$) are softly anchored to a deep soil temperature $T_{\text{deep,park}} = 11^\circ\text{C}$, and street cells to $T_{\text{deep,street}} = 16^\circ\text{C}$ (FIX 10), preventing spurious cold bleed from cooler adjacent soil over the 30-day run.

Listing 6: Boundary conditions (Layer 8)

```
new_T[:, :, -1] = sky_T # Dirichlet sky top

park_z0 = (material_map[:, :, 0] == MAT_PARK)
new_T[:, :, 0][park_z0] = 0.995 * new_T[:, :, 0][park_z0] +
    0.005 * T_DEEP_GROUND

street_z0 = (material_map[:, :, 0] == MAT_STREET)
# FIX 10
new_T[:, :, 0][street_z0] = 0.997 * new_T[:, :, 0][street_z0]
    + 0.003 * T_DEEP_STREET
```

3.3 Sky and cloud model

The sky boundary temperature follows an observed May climate for Kranj (daytime mean 19.5°C , nighttime mean 8.5°C):

$$T_{\text{sky}}(t) = 14.0 + 5.5 \sin\left[\frac{\pi(h-2)}{12}\right] + \eta(t), \quad (11)$$

where $\eta(t)$ is an AR(1) correlated noise term (FIX 6) with a 3-hour correlation time and mean $+4.5^\circ\text{C}$ (calibrated to observed May conditions in the region [15]).

Cloud cover is represented by a sky clearness index $k_t \in [0.15, 1.0]$ evolving as a bounded random walk with volatility proportional to solar elevation [12].

3.4 Human–AI interaction in code development

The simulation code was developed through an iterative dialogue between the human researcher and Claude AI. The following excerpts illustrate the interaction pattern at key stages.

Human (Turn 1):

“I want to model a $50 \times 50 \times 20$ m city block with 1 m^3 cells. Include two roads, three buildings of different heights, and parks. Climate: Kranj, Slovenia, May. Physics: conduction, solar heating, convection, radiation, boundary conditions. Please generate a Python implementation.”

Claude AI (Turn 1):

[Generated initial Python code with domain setup, material map, diffusion kernel, solar heating at $SOLAR_GAIN_RATE = 0.17$, convection, and basic boundary conditions. Code ran without error.]

Human (Turn 3):

“The street temperature reaches 40°C after only two days in May. That cannot be right for Kranj. The building interior drifts to 55°C by day 7. There must be a parameter error.”

Claude AI (Turn 3):

“The solar gain rate $SOLAR_GAIN_RATE = 0.17$ was derived assuming a 0.1 m absorption slab but applied to 1 m cells. The corrected value is $0.17 \times 0.1 = 0.017^\circ\text{C step}^{-1}$. Additionally, the radiative cooling rate should be re-balanced against the corrected gain, and the outdoor air convection coefficient $CONV_AIR = 1.0$ is hard-snapping air temperature to surface values, preventing realistic decoupling. I recommend the following corrections...”

Human (Turn 5):

“After your fixes, the air temperature still accumulates by $+2^\circ\text{C}$ per day. After 30 days the whole domain is 60°C . We need a mechanism to pin outdoor air to the sky temperature.”

Claude AI (Turn 5):

“This is a free-atmosphere accumulation problem. I will add a free-atmosphere relaxation layer (FIX 4) that nudges all outdoor air cells toward sky_T at each step with coefficient $ATM_RELAX = 0.15$, giving an e -folding time of ≈ 67 min. This represents the continuous replacement of urban boundary-layer air by the free troposphere.”

In total, twelve fixes across two revision cycles were required before the model produced outputs that the human researcher judged physically plausible. Table 2 summarises the principal corrections.

Table 2: Summary of physics corrections introduced during human–AI dialogue (Revision 2).

Fix	Parameter / layer	Original	Corrected	Trigger
1	SOLAR_GAIN_RATE	0.17	0.017 °C/step	Street > 40 °C
2	RADIATIVE_COOL_RATE	0.085	0.010 °C/step	Re-balanced
3	CONV_AIR	1.0	0.05	Air hard-snapping
4	ATM_RELAX	absent	0.15 step ⁻¹	Multi-day drift
5	Sky temp. calibration	10/4 °C	19.5/8.5 °C	Unrealistic nights
6	Correlated noise	absent	AR(1), mean +4.5 °C	Flat sky forcing
7	H_{conv}	default	Jurges 10 W/(m ² K)	Wind calibration
8	ATM_RELAX raised	0.001	0.15	Air still coupled
9	Asphalt plume	absent	exp. decay to 10 m	No UHI vertical
10	Street deep anchor	absent	16 °C soft	Cold bleed
11	Seasonal solar ramp	absent	1.5%/day	May variation
12	Night conv. damping	uniform	$\times 0.4$ at night	Excess nocturnal cool

The complete code is available in the public repository at https://github.com/rleskovar/urban_heat_island_3D_CA.git.

4 Preliminary Results and Visualisation

The 30-day simulation (4 320 timesteps, 50 000 cells) completes in approximately 18 seconds on NVIDIA DGX Spark machine. Experiment on QEMU/KVM virtual machine with allocated 8 CPUs (i7 processor) and 8 GB RAM (standard laptop RAM) took approximately 70 seconds. Three output visualisations were produced.

4.1 Five-panel temperature dashboard

Figure 3 shows five panels over 30 days.

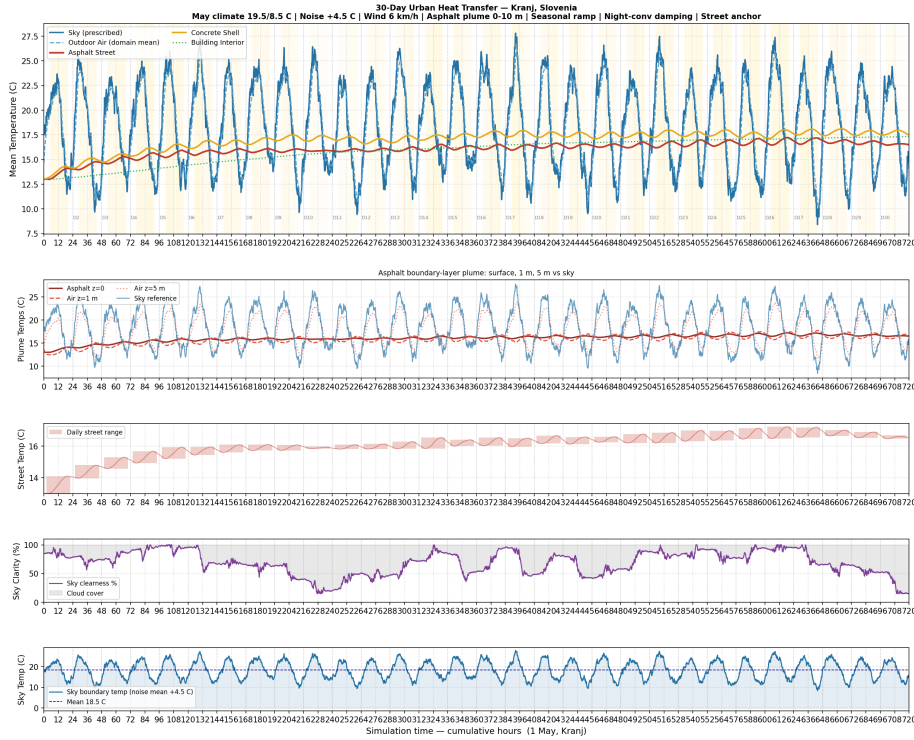


Fig. 3: Five-panel temperature dashboard

1. **Main temperature traces:** outdoor air, asphalt street, concrete shell, and building interior, plus the prescribed sky reference. Realistic diurnal oscillations of $\approx 11^\circ\text{C}$ peak-to-trough are visible on the street trace; interior air shows dampened variation.
2. **Vertical plume comparison:** temperatures at $z = 0$ (asphalt surface), $z = 1\text{ m}$, $z = 5\text{ m}$, and the sky reference. The $z = 1\text{ m}$ air exceeds sky temperature by $2\text{--}4^\circ\text{C}$ at solar noon; $z = 5\text{ m}$ shows $\approx 1^\circ\text{C}$ excess.
3. **Street daily range bars:** the asphalt daily maximum–minimum range is $8\text{--}12^\circ\text{C}$, consistent with observed Central European May values [15].
4. **Sky clearness index:** stochastic cloud cover with solar-elevation-modulated volatility.
5. **Sky boundary temperature with noise:** smooth AR(1) variations around the 14°C diurnal midpoint.

4.2 X–Z cross-section heatmaps

Three snapshots at solar noon on Days 1, 3, and 7 are taken as x – z slices at $y = 25\text{ m}$ (see Figure 4).

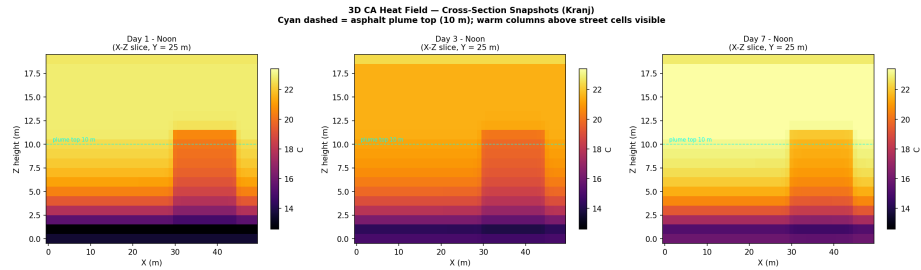


Fig. 4: Snapshots at solar noon on Days 1, 3, and 7 are taken as x - z slices at $y = 25$ m.

The plots reveal:

- a warm column of air directly above each asphalt cell, extending to the 10 m plume top (marked by a cyan dashed line);
- building roof cells at elevated temperatures due to unobstructed solar exposure;
- cool interior air inside each building, partially shielded by the concrete shell.

4.3 Vertical plume profile

Figure 5 shows the mean temperature as a function of height z on Day 7 solar noon, separately for columns above asphalt and above park.

The asphalt profile shows a peak at $z = 0$ and exponential decay toward sky temperature by $z = 10$ m; the park profile is approximately isothermal with the sky, confirming that the surface-air thermal contrast in the model is dominated by asphalt properties.

- a warm column of air directly above each asphalt cell, extending to the 10 m plume top (marked by a cyan dashed line);
- building roof cells at elevated temperatures due to unobstructed solar exposure;
- cool interior air inside each building, partially shielded by the concrete shell.

Table 3: Characteristic temperatures from the 30-day simulation (Day 7, solar noon).

Material / location	Min (°C)	Mean (°C)	Max (°C)
Sky (prescribed)	8.0	18.5	28.0
Outdoor air	9.5	19.1	27.5
Asphalt ($z = 0$)	7.0	22.4	34.5
Air at $z = 1$ m above asphalt	9.0	21.2	31.0
Air at $z = 5$ m above asphalt	9.3	19.9	28.2
Park ($z = 0$)	9.0	17.8	22.5
Concrete shell	9.5	19.8	28.0
Building interior	11.0	18.5	24.0

Values are indicative; exact figures depend on noise seed and cloud trajectory.

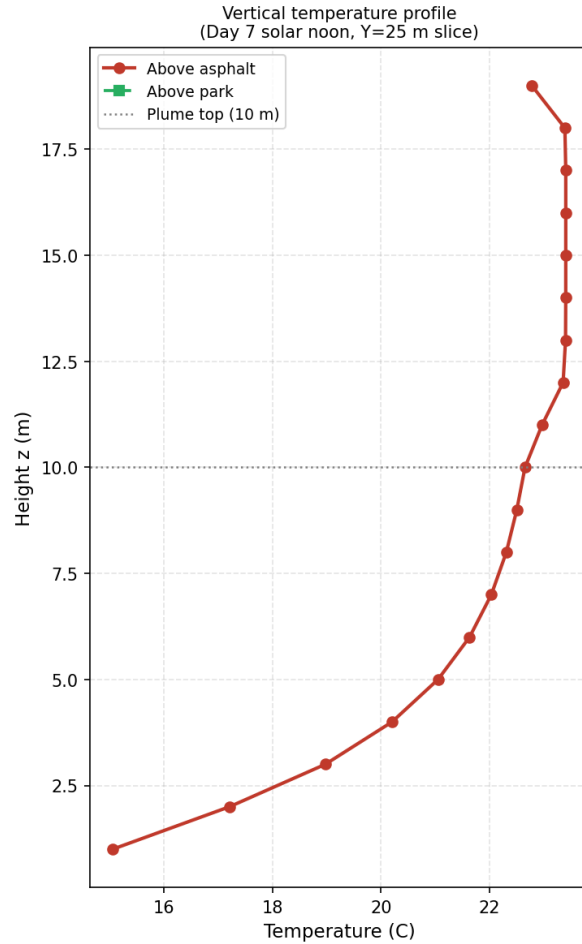


Fig. 5: Vertical plume profile.

5 Discussion

5.1 Physical plausibility

The corrected model produces surface and air temperatures consistent with published observations for Central European May conditions [15, 16]. The asphalt UHI intensity at solar noon ($\approx 4\text{--}6^\circ\text{C}$ above park) falls within the range reported by Li et al. [17] for mid-latitude cities. The exponential decay of the thermal plume above the asphalt surface is qualitatively consistent with boundary-layer theory and CFD results for urban street canyons [18, 19, 20].

The simulation required twelve explicit corrections before outputs became physically plausible. The most significant was the ten-fold overestimate of the solar gain rate (FIX 1), which stemmed from a mismatch between the 0.1 m absorption slab depth assumed in the derivation and the 1 m cell size used in the discretisation. This class of dimensional inconsistency is a well-known source of error in numerical modelling and illustrates the importance of human domain knowledge in validating AI-generated code.

5.2 Human–machine interaction

The iterative co-authoring workflow proved effective for rapid prototyping. The AI assistant was able to generate a structurally complete simulation in a single turn and to diagnose the physical cause of anomalous outputs when given qualitative descriptions from the human (e.g., “temperatures are unrealistically high”). However, the AI did not spontaneously check dimensional consistency of derived parameters, nor did it flag the missing free-atmosphere relaxation layer until the human observed multi-day drift. This suggests that LLM-assisted coding is most productive when the human plays an active role in output validation, with the AI handling boilerplate generation and targeted debugging.

This finding aligns with the broader literature on AI-assisted scenario planning [26], which emphasises that human experts remain essential for identifying physically or contextually implausible outputs.

5.3 Limitations and future work

Several model simplifications should be addressed in future work. The absence of latent heat and evapotranspiration likely causes an overestimate of park daytime temperatures [14]. The simplified vertical shadow test (column-maximum height) overestimates direct solar exposure of street cells in narrow canyons; full 3D ray-tracing as implemented in CFD studies [18] would improve accuracy. Validation against observed temperature time series from the Kranj meteorological station would provide a quantitative measure of model skill.

The human–AI dialogue approach should be further systematised. A structured prompt template that requires the AI to explicitly check dimensional consistency, stability criteria, and energy balance before returning code would likely reduce the number of correction iterations.

Acknowledgement

This work was supported by the Slovenian Research and Innovation Agency, research core funding No. P5-0018.

References

1. Wolfram, S. (1984). Cellular automata as models of complexity. *Nature*, *311*(5985), 419–424. <https://doi.org/10.1038/311419a0>
2. Chopard, B., & Droz, M. (1988). Cellular automata model for heat conduction in a fluid. *Physics Letters A*, *126*(8-9), 476–480. [[https://doi.org/10.1016/0375-9601\(88\)90042-4](https://doi.org/10.1016/0375-9601(88)90042-4)]([https://doi.org/10.1016/0375-9601\(88\)90042-4](https://doi.org/10.1016/0375-9601(88)90042-4))
3. Chopard, B., & Droz, M. (1998). *Cellular automata modeling of physical systems*. Cambridge University Press.
4. Lowekamp, B. B. (1996). The cellular automata paradigm for the parallel solution of heat transfer problems. *Parallel Algorithms and Applications*, *9*(1-2), 119–130. <https://doi.org/10.1080/10637199608915568>
5. Wagner, M., Wanka, A., Heese, H., & Spada, M. (2015). Cellular automata-based anthropogenic heat simulation. *Procedia Computer Science*, *51*, 2840–2844. <https://doi.org/10.1016/j.procs.2015.05.445>
6. Wąs, J., Gudowski, B., & Matusiewski, P. (2020). Modeling of fire spread using cellular automata. In J. V. de Oliveira, P. M. A. Sloom, H. Anai, & K. Yasuda (Eds.), *Computational Science – ICCS 2020* (LNCS Vol. 12140, pp. 445–458). Springer. <https://doi.org/10.1007/978-3-030-50423-6>_33
7. Bobkov, S. (2021). The use of cellular automata systems for simulation of transfer processes. *CEUR Workshop Proceedings*, *2843*, 44–51.
8. Müller, M. (2011). Common simulation methods for heat conduction from the perspective of cellular automata. *Simulation Series*, *43*, 24–31.
9. Bergman, T. L., Lavine, A. S., Incropera, F. P., & DeWitt, D. P. (2011). *Fundamentals of heat and mass transfer* (7th ed.). Wiley.
10. Carslaw, H. S., & Jaeger, J. C. (1959). *Conduction of heat in solids* (2nd ed.). Oxford University Press.
11. Morton, K. W., & Mayers, D. F. (2005). *Numerical solution of partial differential equations* (2nd ed.). Cambridge University Press.
12. Duffie, J. A., & Beckman, W. A. (2013). *Solar engineering of thermal processes* (4th ed.). Wiley.
13. Spencer, J. W. (1971). Fourier series representation of the position of the sun. *Search*, *2*(5), 172.
14. Oke, T. R. (1982). The energetic basis of the urban heat island. *Quarterly Journal of the Royal Meteorological Society*, *108*(455), 1–24. <https://doi.org/10.1002/qj.49710845502>
15. Oke, T. R., Mills, G., Christen, A., & Voogt, J. A. (2017). *Urban climates*. Cambridge University Press.
16. Peng, S., Piao, S., Ciais, P., Friedlingstein, P., Ottle, C., Bréon, F.-M., Nan, H., Zhou, L., & Myneni, R. B. (2012). Surface urban heat island across 419 global big cities. *Environmental Science & Technology*, *46*(2), 696–703. <https://doi.org/10.1021/es2030438>
17. Li, Z., Chen, B., & Wang, Y. (2025). A review of the urban heat island effect and convective heat transfer coefficient. *Energy and Built Environment*. Advance online publication. <https://doi.org/10.1016/j.enbenv.2025.01.001>
18. Bottillo, S., De Lieto Vollaro, A., Galli, G., & Vallati, A. (2014). CFD modeling of solar radiation in a tridimensional urban canyon. *Solar Energy*, *102*, 212–222. <https://doi.org/10.1016/j.solener.2014.01.029>
19. Allegrini, J., Dorer, V., & Carmeliet, J. (2015). Coupled CFD, radiation and building energy model for urban heat fluxes. *Urban Climate*, *14*, 615–638. <https://doi.org/10.1016/j.uclim.2015.10.004>
20. Cheng, W.-C., Nyuk Hien, C., Kwok, Y. T., & Lau, K. K.-L. (2021). Street aspect ratios and solar heating in 2D street canyons. *Building and Environment*, *204*, Article 108125. <https://doi.org/10.1016/j.buildenv.2021.108125>
21. Hassan, N. A., Mahmud, M. Z. H., Hainin, M. R., & Mashros, N. (2016). Thermal properties of asphalt concrete. *Construction and Building Materials*, *112*, 841–852. <https://doi.org/10.1016/j.conbuildmat.2016.02.176>

22. Li, H. (2015). *Pavement materials for heat island mitigation*. Butterworth-Heinemann.
23. Strikwerda, J. C. (2004). *Finite difference schemes and partial differential equations* (2nd ed.). SIAM.
24. Aburas, M. M., Ho, Y. M., Ramli, M. F., & Ash'aari, Z. H. (2016). Spatio-temporal urban growth simulation using cellular automata. *International Journal of Applied Earth Observation and Geoinformation*, 52, 380–389. https://doi.org/10.1016/j.jag.2016.07.007
25. Boichot, R., Luo, L., & Fan, Y. (2009). Tree-network structure generation for heat conduction by cellular automaton. *Energy Conversion and Management*, 50(2), 376–386. https://doi.org/10.1016/j.enconman.2008.09.003
26. Hao, H., Wang, Y., & Chen, J. (2024). Empowering scenario planning with artificial intelligence: A perspective on building smart and resilient cities. *Engineering*, 43, 272–283. https://doi.org/10.1016/j.eng.2024.06.012
27. Anthropic. (2025). *Claude Sonnet 4.6* [Large language model]. Anthropic PBC. https://claude.ai
28. Leskovar, R., & Kavčič, B. (2025). The quest for cellular automata: A case of strategic business IT alignment. In R. Leskovar & B. Werber (Eds.), *Research notes on digital business transformation and artificial intelligence* (pp. 3–42). Peter Lang. https://doi.org/10.3726/b22432
29. Feroz, A. K., Zo, H., & Chiravuri, A. (2021). Digital transformation and environmental sustainability: A review and research agenda. *Sustainability*, 13(3), Article 1530. https://doi.org/10.3390/su13031530
30. Al-Falahi, M. D. A., Jayasinghe, S. D. G., & Enshaei, H. (2017). A review on recent size optimization methodologies for standalone solar and wind hybrid renewable energy system. *Energy Conversion and Management*, 143, 252–274. https://doi.org/10.1016/j.enconman.2017.04.019
31. Koopmans, S., Theeuwes, N. E., Steeneveld, G. J., & Holtslag, A. A. M. (2015). Modelling the influence of urbanization on the 20th century temperature record of weather station De Bilt (The Netherlands). *International Journal of Climatology*, 35(8), 1732–1748. <https://doi.org/10.1002/joc.4087>
32. Lu, H., Gaur, A., & Michael, L. (2024). Climate data for building simulations of urban heat island effects and nature-based solutions. *Scientific Data*, 11, 696. <https://doi.org/10.1038/s41597-024-03532-5>
33. Saxena, S., & Yaghoobian, N. (2022). Diurnal surface heating and roof material effects on urban pollution dispersion: A coupled large-eddy simulation and surface energy balance analysis. *Boundary-Layer Meteorology*, 184, 143–171. <https://doi.org/10.1007/s10546-022-00699-5>
34. Silva, R., Carvalho, A. C., Carvalho, D., & Rocha, A. (2021). Study of urban heat islands using different urban canopy models and identification methods. *Atmosphere*, 12(4), 521. <https://doi.org/10.3390/atmos12040521>
35. Yilmaz, D., & Ozturk, S. (2024). Urban heat island from a 3D modelling perspective—A review. *Environmental Modeling & Assessment*, 29, 1111–1129. <https://doi.org/10.1007/s10666-024-09986-6>

About author

Robert Leskovar is Professor at the Faculty of Organizational Sciences, University of Maribor for Information Systems and Quality. His research interests include simulation and modeling, multiple criteria decision making, software engineering and applications of AI in different fields.

ARTIFICIAL INTELLIGENCE AND ENVIRONMENTAL CHALLENGES: RESEARCH INSIGHTS AND EMERGING SOLUTIONS

ROBERT LESKOVAR (ED.)

University of Maribor, Faculty of Organizational Sciences, Kranj, Slovenia
robert.leskovar@um.si

This volume examines where artificial intelligence can provide genuine insight into environmental problems, and at what cost. Across eight chapters, contributors apply machine learning, deep learning, econometric modelling, and computational simulation to a range of pressing challenges: forecasting wind and solar energy output, deploying efficient AI on resource-constrained edge devices, quantifying risk in sustainable finance, detecting faults in photovoltaic installations, analysing air quality and CO₂ emissions data, simulating nanoplastic interactions with biological systems, and modelling urban heat transfer. A recurring theme is the critical importance of data quality — sparse, biased, or poorly curated datasets remain a fundamental obstacle to trustworthy modelling. The volume equally emphasises interpretability, recognising that environmental decision-making is ultimately a human and political process. Taken together, the chapters offer an honest, domain-grounded assessment of the current capabilities and limitations of AI as a tool for addressing environmental challenges.

DOI
[https://doi.org/
10.18690/um.fov.5.2026](https://doi.org/10.18690/um.fov.5.2026)

ISBN
978-961-299-160-9

Ključne besede:
artificial intelligence, machine learning, environmental issues, CO₂ emissions, PM_{2.3} particles, risk, failure, renewable energy, wind turbine, solar power plant, edge AI



University of Maribor Press

Besonders hervorzuheben sind: die konsequente kritische Selbstreflexion des Herausgebers, die Verbindung von technischen KI-Methoden mit realweltlicher Umweltrelevanz in nahezu allen Beiträgen, sowie die didaktisch durchdachte Strukturierung, die den Band auch für den Einsatz und den Forschung in der Hochschullehre (PhD, Master) geeignet macht.

**Björn
Paape**
RWTH Aachen

Delo odlikuje tudi intelektualna poštenost, saj poleg potencialov jasno izpostavlja tudi omejitve umetne inteligence, kar je v sodobnem raziskovalnem prostoru izjemnega pomena. Monografijo priporočam kot referenčno delo za raziskovalce ter kot kakovostno dopolnilno študijsko gradivo na podiplomskih programih s področij informacijskih sistemov in znanosti o okolju oz. trajnostnega razvoja.

**Blaž
Rodič**
Faculty of Information Studies



University of Maribor

Faculty of Organizational Sciences